



UNIVERSITAT DE
BARCELONA

Treball de Fi de Grau
GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

**Millora de l'aplicació 'Every Walk you Take' pel
foment de l'activitat física i la ciència ciutadana.**

Marc Serrajordi Domènech

Director: Laura Igual Muñoz i
Maria Grau Magaña

Realitzat a: Departament de
Matemàtiques i
Informàtica

Barcelona, 13 de juny de 2023

Resum

Avui en dia, la població de Catalunya gaudeix d'una de les esperances de vida més altes del món. Amb l'augment de l'edat mitjana, és crucial garantir una qualitat de vida òptima. Tot i els avenços en tecnologia i medicina, aquesta franja de la població és més susceptible a malalties i infeccions. A més, els nivells elevats de contaminació actuals tenen un impacte directe en aquestes malalties, per tant, és fonamental abordar aquesta qüestió per promoure un envelliment saludable. La durada de la nostra vida i la proporció d'aquesta que transcorre en un bon estat de salut tenen repercussions importants tant per les persones com per les societats.

Aquest projecte és la continuació del Treball de Final de Grau "*Definició de rutes saludables a partir de dades de geolocalització i contaminació*" realitzat per Albert Pérez i consisteix en la millora de l'aplicació "Every Walk you Take" per tal de fomentar l'activitat física i l'envelliment saludable de les persones.

Concretament, es vol assolir aquests objectius mitjançant la implementació de noves funcionalitats, com ara tenir en compte la sensació tèrmica a l'hora de recomanar si una ruta és saludable o no, la implementació de dos mòduls de ciència ciutadana per a obtenir informació sobre les rutes i altres modificacions.

Resumen

Hoy en día, la población de Cataluña disfruta de una de las esperanzas de vida más altas del mundo. Con el aumento de la edad mediana, es crucial garantizar una calidad de vida óptima. A pesar de los avances en tecnología y medicina, esta franja de la población es más susceptible a enfermedades e infecciones. Además, los niveles elevados de contaminación actuales tienen un impacto directo en estas enfermedades, por lo tanto, es fundamental abordar esta cuestión para promover un envejecimiento saludable. La duración de nuestra vida y la proporción de ésta que transcurre en un buen estado de salud tienen repercusiones importantes tanto para las personas como para las sociedades.

Este proyecto es la continuación del Trabajo de Final de Grado “Definición de rutas saludables a partir de datos de geolocalización y contaminación” realizado por Albert Pérez y se centra en la mejora de la aplicación “Every Walk you Take” para fomentar la actividad física y el envejecimiento saludable de las personas.

Concretamente, en este proyecto se realiza la implementación de nuevas funcionalidades, como por ejemplo, tener en cuenta la sensación térmica a la hora de recomendar si una ruta es saludable o no, la implementación de dos módulos de ciencia ciudadana para obtener información sobre las rutas, entre otras modificaciones.

Abstract

Nowadays, the population of Catalonia enjoys one of the highest life expectancies in the world. With the increase in median age, it is crucial to ensure an optimal quality of life. Despite advancements in technology and medicine, this population group is more susceptible to diseases and infections. Additionally, the current high levels of pollution have a direct impact on these illnesses; therefore, it is essential to address this issue to promote healthy aging. The length of our lives and the proportion of it spent in good health have significant implications for both individuals and societies.

This project is a continuation of the Bachelor's Thesis “Definition of Healthy Routes based on Geolocation Data and Pollution” carried out by Albert Pérez. It aims to improve the “Every Walk you Take” application to support physical activity and healthy aging for individuals.

Specifically, these objectives will be achieved through the implementation of new features, such as considering the thermal sensation when recommending whether a route is healthy or not, the incorporation of two citizen science modules to gather information about the routes, and other modifications.

ÍNDEX

Resum.....	2
Resumen.....	3
Abstract.....	4
1. Introducció.....	6
1.1. Context i motivació.....	6
1.2. Objectius.....	7
2. Anàlisi previ.....	9
2.1. Aplicacions ja existents.....	9
2.2. Treball fet prèviament.....	9
2.3. Ciència ciutadana.....	12
2.3.1. Ciència ciutadana en aquest treball.....	13
3. Anàlisi.....	14
3.1. Casos d'ús.....	14
3.2. User Stories.....	15
4. Planificació.....	21
4.1. Metodologia àgil Scrum.....	21
4.2. Kanban.....	25
4.3. Diagrama de Gantt.....	26
5. Cost del projecte.....	27
6. Disseny i implementació.....	30
6.1. Arquitectura frontend.....	30
6.1.1. Programació amb React-Native.....	34
6.1.1.1. React Native Async Storage.....	34
6.1.2. Diagrama de components.....	36
6.2. Arquitectura backend.....	39
6.2.1. Estructura del codi.....	39
6.3. Base de dades.....	44
6.3.1. Protecció de dades.....	45
6.4. Endpoints.....	46
7. Resultats.....	49
7.1. Navegació.....	49
7.2. Dades recollides i anàlisi de les dades.....	56
7.3. Every Walk You Take.....	58
8. Conclusions.....	59
9. Treballs futurs.....	60
10. Bibliografia.....	61
11. Annex.....	62

1. Introducció

1.1. Context i motivació

Aquest projecte està emmarcat dins d'un estudi de recerca anomenat "Every Walk You Take", liderat per la Dra. Maria Grau de la Facultat de Medicina de la UB, el qual ha rebut finançament de la Unió Europea (més informació a l'apartat [7.3 Every Walk You Take](#)) i que té com a objectiu empoderar a les persones grans per aconseguir un procés d'envelliment saludable.

Dins d'aquest estudi de recerca, al gener 2023 es va presentar el Treball Final de Grau "Definició de rutes saludables a partir de dades de geolocalització i contaminació" [1] realitzat per Albert Pérez Costa. En aquest TFG es va desenvolupar una primera versió de l'aplicació (més informació a l'apartat [2.2. Treball fet prèviament](#)) que ampliarem en aquest treball per tal que es pugui fer servir en la prova de concepte del projecte Europeu i que sigui vàlid per ciència ciutadana.

A continuació introduïm algunes idees sobre el projecte de recerca de context [1]:

Quant de temps vivim i quina proporció d'aquesta vida es passa en bona salut tenen implicacions importants per a les persones i les societats.

Tradicionalment, la fragilitat de les societats envellides s'ha vist exposada per la càrrega social i econòmica de les malalties no transmissibles; tanmateix, hem après de la crisi de la COVID-19 sobre els efectes potencials —i ràpids— de les malalties infeccioses.

Tot i que les innovacions mèdiques modernes han augmentat la longevitat durant el segle passat, el canvi cap a una població més gran amb un major risc de malalties transmissibles i no transmissibles té el potencial de generar una crisi sanitària en els propers anys. Ara és el moment d'avançar i centrar-nos en un envelliment saludable i reeixit mitjançant una vida activa. De fet, la inactivitat física és la quarta causa de mort a tot el món, amb un cost aproximat de 67.500 milions de dòlars anuals als sistemes sanitaris a escala mundial.

Tanmateix, el model clàssic dels determinants de la salut mostra que els estils de vida individuals estan incrustats en les normes i xarxes socials, i en les condicions de vida i de treball, que al seu torn estan relacionades amb l'entorn socioeconòmic i cultural més ampli. Per tant, el repte de l'envelliment actiu i saludable requereix reconèixer que aquesta qüestió depèn no només de l'individu sinó també de les xarxes socials i comunitàries i de les condicions generals socioeconòmiques, culturals i ambientals.

Per promoure una vida activa, és imprescindible entendre les vies a través de les quals els entorns físics i socials influeixen en la vida activa i, al seu torn, com la vida activa influeix en els resultats de salut. El grup creixent de persones de 55 anys i més pot aconseguir un envelliment saludable mitjançant la vida activa en el panorama actual d'una ciutat intel·ligent? El desenvolupament de la vida activa ha demostrat efectes beneficiosos per prevenir tant les malalties transmissibles com les no transmissibles. La nostra hipòtesi és que si oferim recomanacions personalitzades sobre l'activitat física segons les zones verdes del barri, el clima i la qualitat de l'aire, empoderem a la població per aconseguir un procés d'envelliment saludable.

Com comenta l'estudi, es vol tenir en compte les condicions ambientals sobre la població, ja que l'exposició a nivells alts de contaminació de l'aire pot causar una varietat de resultats

adversos a la salut. Aquesta contaminació pot augmentar el risc d'infeccions respiratòries, malalties cardíaques, accidents cerebrovasculars i càncer de pulmó. Tant l'exposició a curt termini com a llarg termini als contaminants de l'aire s'ha associat amb impactes adversos a la salut. Els impactes més severos afecten les persones que ja estan malaltes. Els nens, els ancians i els pobres són més susceptibles. Els contaminants més nocius per a la salut, estretament associats amb la mortalitat prematura excessiva, són partícules fines PM_{2,5} que penetren profundament als conductes pulmonars.

Si bé en general, la qualitat de l'aire als països d'alts ingressos ha millorat en les darreres dècades, els efectes adversos de la contaminació de l'aire ambiental exterior a la salut per partícules (PM per les sigles en anglès) continuen sent un problema mundial de salut pública, fins i tot a nivells relativament baixos.

De tot això, Barcelona no se n'escapa. Avui en dia és notícia la quantitat de contaminació que hi ha a tot el món i sobretot a les ciutats més grans com Barcelona i és per això que ha sorgit la necessitat de realitzar millores tecnològiques amb les quals ajudar a la població a reduir l'exposició, o com a mínim fer-los conscients dels efectes nocius de la contaminació.

Com és el cas d'aquesta aplicació, que té la intenció d'ajudar als usuaris a fer exercici caminant/corrent per zones més verdes i no tan verdes com el centre de la ciutat, però informant dels nivells de contaminació i les condicions meteorològiques perquè les persones siguin conscients de quan és millor sortir a caminar. L'aplicació està orientada a gent gran, amb una experiència d'usuari el més simple possible, ja que la gent d'avançada edat sol tenir menys costum d'utilitzar les tecnologies. Per això, és una app amb indicacions clares i concises i sense un excés de vocabulari tècnic. Això no impedeix que pugui ser usada per gent més jove sense problema.

A més de fer recomanacions sobre les rutes menys exposades a la contaminació, també s'inclouen avisos sobre les condicions meteorològiques. Es considera que un passeig per Barcelona en plena onada de calor pot ser més perjudicial que beneficiós. Així mateix, tampoc s'aconsella caminar si hi ha previsió de pluja.

A part, l'usuari també disposa de l'opció de participar en els dos mòduls realitzats durant aquest treball: el mòdul d'avaluació de l'experiència de la ruta amb puntuació, on s'ofereix a l'usuari donar una valoració subjectiva de l'experiència que ha estat realitzar la ruta i el mòdul d'avaluació de l'estat de la ruta amb retroacció, on l'usuari pot aportar la seva valoració objectiva de la ruta i de l'estat d'aquesta.

1.2. Objectius

L'objectiu principal d'aquest treball és ampliar l'aplicació desenvolupada en el Treball Final de Grau titulat "*Definició de rutes saludables a partir de dades de geolocalització i contaminació*" [1] realitzat per l'estudiant Albert Pérez Costa.

A continuació es llisten les principals funcionalitats de l'aplicació en l'estat final del treball anterior [1] (més informació a l'apartat [2.2 Treball fet prèviament](#)):

- Registrar-se.
- Iniciar sessió / Tancar sessió.
- Modificar perfil.
- Entrar com a invitat.
- Veure tutorial.

- Visualitzar el mapa amb rutes i sense.
- Filtrar rutes per tipus de ruta (*proposta inicial, proposta verda, començar aquí*).
- Filtrar rutes per dificultat.
- Filtrar rutes per temps.
- Veure informació de la ruta com la distància, el nivell de contaminació i la previsió meteorològica.
- Iniciar una ruta.
- Aturar/continuar una ruta.
- Sortir d'una ruta.

L'ampliació d'aquest Treball Final de Grau consisteix a millorar algunes funcionalitats i afegir de noves dins de l'aplicació, així com la implementació de dos mòduls de ciència ciutadana: el mòdul d'avaluació de l'experiència de la ruta amb puntuació i el mòdul d'avaluació de l'estat de la ruta amb retroacció, amb l'objectiu de permetre als usuaris puntuar la ruta que hagin dut a terme i proporcionar retroacció sobre l'estat d'aquesta durant el seu transcurs.

Amb la integració d'aquests mòduls, s'espera involucrar activament als usuaris en el procés de millora de la recomanació de rutes i de l'aplicació. Mitjançant la seva participació, es recolliran dades addicionals que enriqueiran la base de coneixement de l'aplicació, cosa que permetrà proporcionar rutes més saludables, fiables i adaptades als usuaris.

Així doncs, la finalitat principal d'aquest treball és millorar l'aplicació prèvia amb nous mòduls per la col·laboració dels usuaris, creant una eina eficaç i útil per fomentar l'activitat física saludable i minimitzar els efectes negatius de la contaminació ambiental en la salut. Amb la implementació dels mòduls de ciència ciutadana, es busca establir una relació de retroalimentació constant entre l'aplicació i els usuaris, fomentant la participació activa de la comunitat en la millora dels serveis que ofereix l'aplicació.

En resum, els objectius d'aquest treball són les següents millores de l'aplicació:

- Afegir sensació tèrmica a la informació de la ruta.
- Modificar el codi de colors a "Informació de la ruta".
- Definició de reptes per als usuaris.
- Finalitzar ruta automàticament.
- Avaluació de l'experiència de les rutes per part de l'usuari seguint un sistema d'estrelles i comentaris.
- Avaluació de l'estat de les rutes amb retroacció sobre una ruta en forma de text i d'imatge.
- Creació de fins a 3 noves rutes fetes per l'usuari.
- Afegir avís de participació en els mòduls de ciència ciutadana.
- Afegir avís de confirmació de l'estat d'una ruta en finalitzar-la.

2. Anàlisi previ

A continuació s'explica la recerca prèvia sobre aplicacions ja existents, un resum del treball fet prèviament i l'elecció dels llenguatges de programació i que és i perquè fem servir la ciència ciutadana.

2.1. Aplicacions ja existents

Al treball anterior [1] es va realitzar una recerca sobre les aplicacions ja existents al mercat que poguessin ser semblants a aquesta, i les semblances i diferències que hi existeixen entre elles amb la nostra proposta. Aquestes són **Caliope**, **Tiempo iOS** i **AllTrails**, on es va veure que malgrat sí que tenien algun aspecte semblant amb la nostra, cap d'elles contemplava donar rutes amb el factor de la meteorologia i la contaminació.

Des de les hores, durant el desenvolupament d'aquest projecte ha sortit una nova funcionalitat (de moment només als EUA) a un dels productes més populars de Google, **Google Maps**, semblant a la nostra aplicació [4] que cal mencionar en aquest apartat.

Aquesta nova funcionalitat té l'objectiu de permetre als usuaris navegar d'una manera més senzilla els parcs naturals nord-americans, proporcionant les rutes més populars dins els parcs i així entrant una mica dins el territori d>AllTrails, malgrat que només sigui, de moment i com ja he dit, als parcs naturals.

Així doncs, aquesta nova funcionalitat tampoc és tan semblant a la nostra aplicació més enllà que proporcionen rutes. La nostra a part de proporcionar-les també disposa d'advertències de contaminació, sensació tèrmica i previsió meteorològica. També la nostra aplicació està centrada en la gent de ciutat, en particular Barcelona, perquè facin exercici diari.

2.2. Treball fet prèviament

Com ja s'ha mencionat prèviament, aquest Treball Final de Grau és la continuació d'un altre Treball Final de Grau titulat "*Definició de rutes saludables a partir de dades de geolocalització i contaminació*" [1] realitzat per l'estudiant Albert Pérez Costa. A la vegada, aquests dos treballs formen part d'un estudi més gran el qual es va decidir separar en diverses fases on es durien a terme diferents avenços.

El treball de l'Albert Pérez [1] va crear la base de l'aplicació, afegint funcionalitats a aquesta perquè complís uns objectius de funcionament. Les funcionalitats creades van ser les següents:

- **Registrar-se:** Permet a l'usuari crear un compte a l'aplicació i així poder disposar de totes les funcionalitats disponibles.
- **Iniciar sessió:** Un cop l'usuari té un compte, permet iniciar la sessió amb el compte creat.
- **Tancar sessió:** Opció per tancar la sessió.
- **Modificar perfil:** Permet a l'usuari modificar les dades del seu perfil.
- **Visualitzar mapa sense rutes:** Mostra un mapa centrat en la ubicació de l'usuari.
- **Visualitzar mapa amb rutes:** Permet veure algunes de les rutes de la base de dades.

- **Filtrar rutes per temps:** Escollir quan l'usuari vol fer la ruta.
- **Filtrar rutes per dificultat:** Permet escollir a l'usuari la dificultat (grup de distàncies) de la ruta.
- **Filtrar rutes per tipus:** Permet a l'usuari escollir el tipus de ruta a realitzar on les opcions són: Proposta inicial, proposta verda (maximitzant zones verdes) i comença aquí.
- **Veure informació de la ruta:** Ajuda a l'usuari a decidir quina ruta fer amb un modal que permet visualitzar informació sobre la distància, nivell de contaminació i previsió meteorològica. Aquest modal està representat en diferents colors (verd, groc, taronja, vermell) en funció del nivell de contaminació/previsió meteorològica.
- **Visualitzar advertències:** Informa l'usuari dels nivells de contaminació i meteorologia perillosos i desaconsella sortir a caminar en aquell moment.
- **Veure tutorial:** En registrar-se, a l'usuari li apareix un tutorial explicant el funcionament de l'aplicació.
- **Començar una ruta:** Permet a l'usuari iniciar una ruta.
- **Aturar / continuar una ruta:** Permet a l'usuari clicar un botó i aturar la ruta o continuar-la si ja estava pausada.
- **Sortir de la ruta:** Un cop acabada la ruta, pot clicar un botó i sortir d'aquesta.

Així doncs, el treball d'Albert Pérez es va quedar en un punt d'inici a l'estudi de la Dra. Maria Grau des del qual poder continuar amb noves fases per afegir noves funcionalitats a l'aplicació i arribar al seu objectiu final de millorar l'envelliment saludable de les persones.

Es va dur a terme una recerca de tecnologies per a conèixer les millors opcions per a implementar aquesta aplicació, en particular, es van veure els pros i contres de diferents llenguatges de programació populars en el desenvolupament del frontend, tenint com a prioritat que l'aplicació sigui compatible tant amb Android com amb iOS. Algunes de les opcions que es van estar considerant van ser: **Flutter**, **Ionic** i **React Native**, on finalment es va decidir fer servir React Native per sobre de les altres opcions per la facilitat de la qual disposa a l'hora de crear aplicacions natives tant per Android com per iOS, per fer ús del popular llenguatge de programació JavaScript, que facilita la corba d'aprenentatge i la facilitat que aporta el sistema de components.

Pel que fa al backend, des d'un principi es va decidir programar amb Python i crear una API REST, fent que el servidor fos l'encarregat de fer la major part de la feina del tractament de les dades.

Una API (interfície de programació d'aplicacions) és un conjunt de regles que defineixen com les aplicacions o els dispositius poden connectar-s'hi i comunicar-se entre ells. Una API REST (transferència d'estat representacional) és una API que compleix els següents principis de disseny d'una arquitectura REST:

- La comunicació entre client i servidor ha de ser sense estat, cosa que vol dir que cada sol·licitud ha de ser independent de les altres.
- Han de tenir una interfície uniforme, totes les sol·licituds d'API per al mateix recurs han de ser iguals. Per exemple, si volem inserir un usuari nou a la nostra base de dades, haurem d'enviar un JSON el qual disposi del seu correu, la contrasenya, l'edat i el nom, per tant, no ens serveix que ens donin altres dades per aquest recurs.

- El client i el servidor han de ser completament independents entre si, només es poden comunicar per HTTP.
- Han de poder guardar informació a la memòria cau (depenent el recurs).
- Han de seguir una arquitectura en capes, per tal que ni el client ni el servidor sàpiguen si estan parlant entre ells o amb un intermediari.

REST API Model

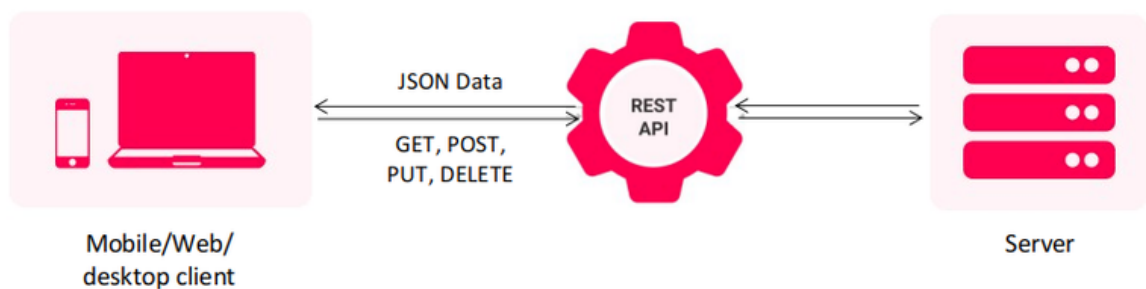


Figura 1: Model API REST

L'ús d'una API RESTful permet que el servidor faci la major part del treball, fent que les aplicacions siguin més escalables, fiables, visibles i simplificades. Així doncs, una aplicació mòbil es beneficiarà si les dades s'emmagatzemen a un servidor remot i no en local consumint més recursos del dispositiu on s'executa.

Es va decidir fer servir Python, ja que és un llenguatge molt popular i que disposa de bones eines per a realitzar API REST, com ara és el framework Flask usat durant el grau.

El procés de desenvolupament tant del frontend com de backend s'explica amb més detall en l'apartat [6. Disseny i implementació](#).

2.3. Ciència ciutadana

Una tema molt important d'aquest treball és la ciència ciutadana, una matèria molt recent i força desconeguda per ara.

El document “*Green Paper on Citizen Science: Citizen Science for Europe*” [3] defineix la ciència ciutadana com “*el compromís del públic general en activitats d'investigació científica, quan la ciutadania contribueix activament a la ciència amb el seu esforç intel·lectual o donant suport al coneixement amb les seves eines o recursos. Els participants voluntaris proveeixen dades experimentals o equips als investigadors. Els voluntaris, alhora que aporten valor a la recerca, adquireixen nous coneixements i habilitats. Com a resultat d'aquest escenari obert, col·laboratiu i transversal, les interaccions entre ciència, societat i polítiques investigadores milloren, conduint a una investigació més democràtica, basada en la presa de decisions fonamentada en evidències sorgides del mètode científic, totalment o parcialment, per part de científics aficionats o no professionals.*”

En els últims anys, la ciència ciutadana ha anat guanyant popularitat i terreny en diversos camps de recerca. Una de les raons principals és la disponibilitat de tecnologia i plataformes digitals que permeten als ciutadans connectar-se i participar en projectes científics. A més, la ciència ciutadana s'ha convertit en una eina poderosa per abordar reptes complexos i promoure la implicació de la societat en la presa de decisions basades en evidències científiques.

La participació dels ciutadans en la ciència ciutadana és essencial per diverses raons. En primer lloc, els ciutadans tenen un coneixement local i una perspectiva única sobre el seu entorn, la qual pot enriquir la recopilació de dades i observació de fenòmens científics. A més, la participació dels ciutadans permet augmentar la cobertura espacial i temporal de les dades, ja que hi ha més persones involucrades en la recopilació de dades i l'observació dels fenòmens. Aquesta ampliació de la cobertura pot ser particularment rellevant en el cas de la monitorització de l'estat de les rutes, perquè permet obtenir dades més representatives i precises.

La ciència ciutadana també beneficia els participants. A través de la seva implicació en projectes científics, els ciutadans adquireixen coneixements i desenvolupen habilitats de recerca. A més, la ciència ciutadana pot fomentar la consciència i el compromís ambiental, ja que els ciutadans es converteixen en part activa de la recerca i prenen consciència dels problemes ambientals locals. Aquesta presa de consciència pot conduir a canvis en el comportament i a l'adopció de pràctiques més sostenibles.

La ciència ciutadana, però, també té algunes limitacions i preocupacions sobre la validesa de les dades generades, com ara que alguns projectes poden no ser adequats per als voluntaris, per exemple quan s'utilitzen mètodes de recerca complexa o requereixen tasques àrdues o repetitives. En el nostre cas, aquesta preocupació no seria del tot vàlida, ja que només es requereix que l'usuari recorri les rutes suggerides per l'aplicació i durant el transcurs i/o final d'aquesta doni retroacció o la puntuï, per tant, no està sent una tasca àrdua o repetitiva, perquè l'usuari pot escollir quina ruta fer en funció de les seves capacitats ni repetitiva, ja que cada cop pot realitzar una ruta diferent.

Una altra preocupació és que la manca de formació dels voluntaris sobre els protocols d'investigació i monitoratge pot produir un biaix en les dades. Novament, aquesta no és una

preocupació que ens afecti, perquè la formació que es requereix és pràcticament nul·la, més enllà de saber interactuar amb l'aplicació i com accedir al mòdul d'avaluació de l'experiència de la ruta amb puntuació i al mòdul d'avaluació de l'estat de la ruta amb retroacció.

L'aspecte d'accés als mòduls d'avaluació s'ha tingut en compte a l'hora de dissenyar l'aplicació fent una interfície senzilla i informant l'usuari de com interactuar amb el mòdul d'avaluació de l'estat de la ruta amb retroacció en començar una ruta i sempre tenint accés al mòdul d'avaluació de l'experiència amb puntuació en finalitzar una ruta.

Finalment, també existeix la preocupació que els usuaris puguin mentir sobre les dades. Aquest és un risc que s'amplifica en el cas que s'ofereixi recompenses com a incentius per participar. En el nostre cas hem de reconèixer que és existent, però molt baix, ja que no hi ha incentius per participar i tampoc beneficia a ningú donar retroaccions ni valoracions de l'experiència de realitzar una ruta no acords amb la realitat.

2.3.1. Ciència ciutadana en aquest treball

Dins el context d'aquest projecte, la ciència ciutadana juga un paper clau en la promoció de l'activitat física saludable i la consciència ambiental.

L'aplicació desenvolupada permet als usuaris enregistrar les seves activitats físiques com caminar o córrer i ofereix recomanacions de rutes saludables basades en diverses variables, com ara la distància, el nivell de contaminació, la previsió meteorològica i la sensació tèrmica. Així doncs, l'aplicació compta amb dos mòduls de ciència ciutadana: el mòdul d'avaluació de l'estat de la ruta amb retroacció i el mòdul d'avaluació de l'experiència amb puntuació.

Aquests mòduls tenen com a objectiu involucrar els usuaris en el procés de millora de les recomanacions de les rutes. Un cop acabada una ruta, l'usuari té l'opció de puntuar-la i afegir un comentari a aquesta puntuació. També, durant el recorregut d'una ruta, els usuaris poden donar retroacció sobre l'estat de la ruta, facilitant informació com quin és el problema i donant més context amb una imatge o un comentari. Aquesta retroacció pot incloure informació sobre la qualitat de l'aire, la meteorologia, la presència de barreres arquitectòniques o qualsevol altre factor que pugui afectar la salut i el benestar dels usuaris o impedeixi realitzar correctament la ruta.

Les dades que s'aconsegueixin a partir de la col·laboració dels voluntaris, aniran a parar a una base de dades on seran analitzades per un equip de recerca amb l'objectiu de millorar la qualitat de les rutes i l'estat general de l'aplicació, i també serviran per a elaborar un atlas de rutes viables a Barcelona.

L'obtenció i explotació d'aquestes dades també ens permetran poder crear diverses noves fases d'aquest estudi, com ara crear un recomanador de rutes, ja que actualment gran part de les rutes estan preestablertes i sempre apareixen les mateixes en fer servir els filtres. També juntament amb els reptes que es defineixin, es podrà fer un “dashboard” on agafar aquestes dades i treure estadístiques i gràfiques, com fer anàlisis de si certa ruta a cert barri es fa més a l'estiu que a l'hivern o si la calor i el fred afecten més als barris pobres que als barris rics.

Aquestes noves possibles fases s'expliquen més a fons a l'apartat [9. Treballs futurs](#)

3. Anàlisi

3.1. Casos d'ús

Seguint el diagrama de casos d'ús dut a terme al treball previ [1], s'ha ampliat per a afegir les noves funcionalitats que s'han desenvolupat per a afegir tot el mòdul de ciència ciutadana. En aquest cas, en la figura 2 podem veure el diagrama de casos d'ús previ amb les noves modificacions afegides (en color vermell).

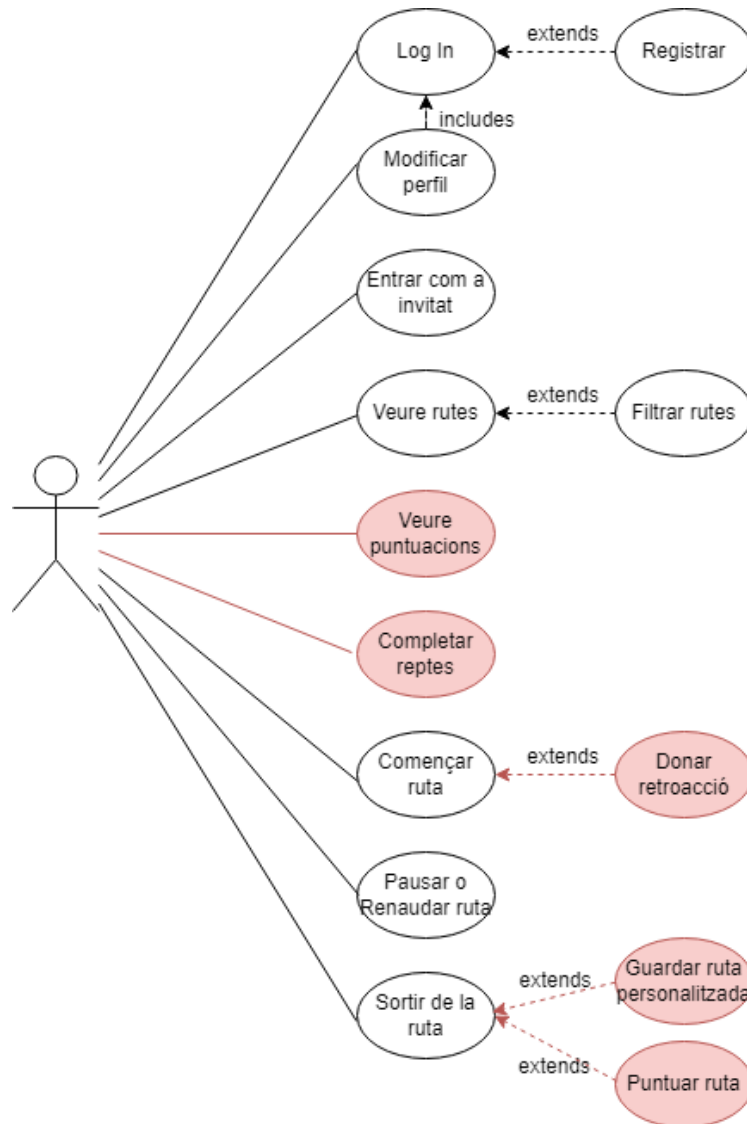


Figura 2: Diagrama de casos d'ús

A la figura 2, podem veure reflectits el diagrama de casos d'ús, on els casos d'ús que ja hi eren al treball anterior [1] en color negre, mentre que els nous han estat posats en color vermell.

3.2. User Stories

Una història d'usuari és una tècnica utilitzada en el desenvolupament de programari àgil per a comprendre i comunicar les necessitats dels usuaris o clients. Una història d'usuari és una breu descripció d'una característica, funcionalitat o requisit des del punt de vista de l'usuari final. S'usa per expressar què vol l'usuari i per a què ho vol, sense especificar com s'ha de fer.

L'estructura d'una història d'usuari es compon de tres elements principals: el títol, la descripció i els criteris d'acceptació.

- **Títol:** El títol de la història d'usuari és una frase concisa que resumeix el que l'usuari vol aconseguir. Ha de ser breu i significatiu perquè pugui ser fàcilment identificable.
- **Descripció:** La descripció amplia el títol i proporciona més detalls sobre la funcionalitat o característica desitjada. Ha de ser clara i específica, però al mateix temps concisa. S'ha de centrar en el que vol l'usuari, no en com s'ha de fer. La descripció pot incloure exemples o situacions concretes per ajudar a entendre millor el que es demana. Normalment segueix l'estructura següent: Tipus d'usuari -> Acció -> Objectiu o “**Com** a <tipus d'usuari>, **vull** <objectiu/desitjos> **per** <finalitat>”.
- **Criteris d'acceptació:** Aquests són els criteris específics que han de ser satisfers per considerar que la història d'usuari està completada amb èxit. Els criteris d'acceptació han de ser mesurables i observables. S'utilitzen per validar i provar la funcionalitat implementada.

Per tal de fer servir aquesta tècnica, es va crear un Product Backlog a l'eina Trello amb totes les històries d'usuari (US) creades durant el desenvolupament d'aquest treball. A continuació es mostren a la figura 3:

Títol	Descripció	Criteris d'acceptació
US1 Modificar codi de colors a “Informació de la ruta”	Com Usuari registrat i loguejat (o invitat) Vull veure el color de la ruta al mapa Per saber el nivell de contaminació, previsió meteorològica i sensació tèrmica que hi ha a una ruta.	Les rutes podrà tenir 3 colors; Verd, Taronja, Vermell. On cadascun indica un nivell: Verd → Si la ruta té contaminació < 50 ICA, no plou i sensació tèrmica entre 18 i 24 °C. Groc → Si la ruta té contaminació entre 50 i 150 i/o plou moderat i/o sensació tèrmica entre 10-18/24-30 °C. Vermell -> Si la ruta té contaminació > 150 i/o plou molt i/o sensació tèrmica

		>30 o <10 °C.
US2 Afegir sensació tèrmica a “informació de la ruta”	<p>Com a usuari registrat i loguejat (o invitat)</p> <p>Vull poder veure la sensació tèrmica d’una ruta</p> <p>Per poder decidir si fer-la o esperar a un altre moment o escollir-ne una altra.</p>	<p>Modal on es veu la distància que té la ruta, el nivell de contaminació ICA, la previsió meteorològica i la sensació tèrmica per al dia i hora seleccionat.</p> <p>A més, el color del requadre seguirà la mateixa lògica que el color de les rutes:</p> <p>Verd → Si la ruta té contaminació < 50 ICA, no plou i sensació tèrmica entre 18 i 24 °C.</p> <p>Groc → Si la ruta té contaminació entre 50 i 150 i/o plou moderat i/o sensació tèrmica entre 10-18/24-30 °C.</p> <p>Vermell -> Si la ruta té contaminació > 150 i/o plou molt i/o sensació tèrmica >30 o <10 °C.</p>
US3 Puntuar (i comentar) una ruta	<p>Com Usuari registrat i loguejat</p> <p>Vull poder donar feedback a una ruta completada</p> <p>Per poder compartir la meva experiència i opinar sobre la ruta.</p>	<p>L'usuari en sortir d’una ruta li apareixerà un modal per a puntuar una ruta.</p> <p>Aquest modal disposa d'un selector d'1 a 5 estrelles i l'opció d'afegir un comentari de text al respecte. Si clica el botó enviar, la puntuació es guarda, en cas de clicar el botó de cancel·lar, no s'envia.</p>
US4 Veure rutes puntuades	<p>Com Usuari registrat i loguejat</p> <p>Vull poder veure el feedback que he donat a rutes</p> <p>Per a tenir un registre de les</p>	<p>Amb sessió iniciada i des de la pantalla principal amb el mapa, fent clic al botó superior dret accedim al menú lateral i tindrem l'opció "Historial puntuacions". Allà podrem</p>

	meves experiències.	veure una llista de les puntuacions realitzades per l'usuari, amb el nom de la ruta, el nombre d'estrelles i el comentari que hagi donat.
US5 Acabar una ruta automàticament	Com a Usuari registrat i loguejat (o invitat) Vull poder acabar automàticament una ruta Perquè ja l'he completat	L'usuari en arribar al final d'una ruta, aquesta acabarà automàticament i donarà l'opció de puntuar-la si aquest té la sessió iniciada.
US6 Completar reptes	Com a usuari registrat i loguejat Vull poder completar reptes Per així motivar-me a fer més activitat física	Amb la sessió iniciada, quan l'usuari completa una ruta, aquesta automàticament s'afegeix a un comptador de rutes completades durant la setmana per l'usuari a la base de dades.
US7 Veure nombre de reptes completats	Com a usuari registrat i loguejat Vull poder veure el nombre de reptes completats Per així motivar-me a completar el següent repte.	Amb sessió iniciada i des de la pantalla principal amb el mapa, fent clic al botó superior dret accedim al menú lateral i veurem un text que informa del nombre de reptes completats. Allà podrem veure un text amb una medalla de bronze, plata o or depenent de si l'usuari ha completat 2, 4 o 6 reptes la setmana actual.
US8 Visualitzar advertències	Com Usuari registrat i loguejat (o invitat) Vull visualitzar les advertències Per ser conscient dels nivells de xafogor, contaminació i previsió meteorològica perillosos.	L'usuari ha fet clic sobre una ruta que té una contaminació elevada i/o previsió meteorològica i/o sensació tèrmica desfavorable. Li apareix un requadre al mig de la pantalla desaconsellant sortir a caminar en aquell moment.
US9 Donar retroacció sobre una ruta en forma de text	Com a usuari registrat i loguejat	Mentre l'usuari està caminant, pot clicar al botó superior dret per parar la ruta

	<p>Vull poder informar sobre l'estat d'una ruta</p> <p>Per a col·laborar a millorar-les</p>	<p>si vol donar retroacció a la ruta per qualsevol motiu i li apareixerà un modal on podrà escollir el tipus de retroacció que vol donar i afegir un comentari opcional.</p> <p>Les opcions disponibles al modal són: Ruta en bon estat, ruta en mal estat, ruta tancada, altres. En cas d'altres, l'usuari disposa d'un quadre de text per a indicar el significat d'altres.</p> <p>També disposa d'una opció per afegir un comentari al respecte en forma de text.</p> <p>Un cop seleccionat el tipus de retroacció que vol donar, s'activa el botó d'enviar, que de base està desactivat.</p>
<p>US10 Donar retroacció sobre una ruta en forma d'imatge</p>	<p>Com a usuari registrat i loguejat</p> <p>Vull poder fer una fotografia sobre l'estat d'una ruta</p> <p>Per a col·laborar a millorar-les per altres usuaris</p>	<p>Mentre l'usuari està caminant, pot clicar al botó superior dret per parar la ruta si vol donar retroacció a la ruta per qualsevol motiu i li apareixerà un modal on podrà escollir el tipus de retroacció que vol donar i afegir/prendre una foto opcional.</p> <p>Les opcions disponibles al modal són: Ruta en bon estat, ruta en mal estat, ruta tancada, altres. En cas d'altres, l'usuari disposa d'un quadre de text per a indicar el significat d'altres.</p> <p>També disposa d'una opció per afegir una fotografia per a donar més context.</p> <p>En clicar l'opció de fotografia, apareixen dos</p>

		<p>botons, un per fer servir la càmera i un altre per seleccionar una fotografia de la galeria d'imatges.</p> <p>Un cop una imatge ha estat presa o seleccionada, apareix en pantalla perquè l'usuari sigui conscient de la imatge que enviarà amb la retroacció.</p> <p>Una vegada seleccionat el tipus de retroacció que vol donar, s'activa el botó d'enviar, que de base està desactivat.</p>
<p>US11 Guardar ruta personalitzada</p>	<p>Com a usuari registrat i loguejat</p> <p>Vull poder guardar una ruta que hagi estat creada desde la meua ubicació</p> <p>Per a poder-la tornar a fer més endavant</p>	<p>L'usuari en sortir d'una ruta que ha estat creada amb el filtre de ruta "Començar aquí" li apareixerà un modal amb l'opció de guardar la ruta si li ha agradat, per a disposar de l'opció de tornar-la a fer.</p> <p>Aquest modal disposa d'un botó guardar i un botó cancel·lar, a part d'un text que informa de l'opció de guardar la ruta a la base de dades.</p>
<p>US12 Mostrar avís de ciència ciutadana</p>	<p>Com a usuari registrat i loguejat (o invitat)</p> <p>Vull rebre un avís</p> <p>Per estar informat que puc participar en el projecte de ciència ciutadana reportant problemes a les rutes</p>	<p>A l'usuari, un cop comenci una ruta, se li mostrarà un avís informant que disposa de l'opció de donar retroacció a una ruta i col·laborar així amb el projecte de ciència ciutadana està disponible clicant el botó que apareix a dalt a la dreta.</p> <p>Un cop inici una ruta, li apareix el modal amb aquesta informació, on haurà de clicar "D'acord" indicant que ha llegit l'avís.</p>

<p>US13 Mostrar avís de ruta correcte</p>	<p>Com a usuari registrat i loguejat</p> <p>Vull rebre un avís</p> <p>Per a tenir l'opció de donar retroacció a la ruta que he completat dient si estava en bon estat o no.</p>	<p>A l'usuari, un cop acabi una ruta, se li mostrarà un avís demanant si la ruta que ha realitzat ha estat en bon estat, on disposa de l'opció de contestar "sí", "no", "no contestar". En cas de clicar "no", a l'usuari li apareix el modal de retroacció per a omplir-lo amb els problemes que hagi pogut trobar. En cas de clicar "sí" o "no contestar" l'usuari continua a la pàgina on es poden veure les rutes.</p>
--	--	--

Figura 3: User stories creades

Totes aquestes US estan marcades amb prioritat de l'1 al 5. De tal manera que es prioritza les que es considerava que tenen un major impacte en el desenvolupament inicial de l'aplicació.

4. Planificació

Per tal de portar un procés de desenvolupament ordenat i on tant el desenvolupador com el client (equip de la Dra. Maria Grau) disposin de la informació que es planeja fer en cada moment i que s'està fent actualment, es va decidir des d'un primer moment fer servir la metodologia àgil Scrum, feta servir durant el grau. D'aquesta manera, es van crear les històries d'usuari amb tots els requisits que havia de complir l'aplicació.

4.1. Metodologia àgil Scrum

Durant el desenvolupament del projecte s'ha utilitzat la metodologia àgil Scrum (figura 4), que permet abordar projectes complexos en entorns més dinàmics i canviants d'una manera flexible.

L'estructura està basada en entregues parcials i regulars (Sprints) del producte final. Com podem veure a la figura 3, tenim un Product Backlog (les històries d'usuari creades a l'apartat [3.2 User Stories](#)), on es divideixen en Sprints, que en el meu cas duraven entre 2 i 3 setmanes, on conclouen amb una reunió amb les tutores.

En aquesta reunió s'exposava el que s'havia dut a terme durant l'Sprint d'una manera més visual i quines coses faltaven per fer. Durant la reunió podien sorgir nous requisits o modificacions per part del client, que s'afegien al Product Backlog i es realitzaven en un Sprint següent depenent de la prioritat d'aquestes modificacions.

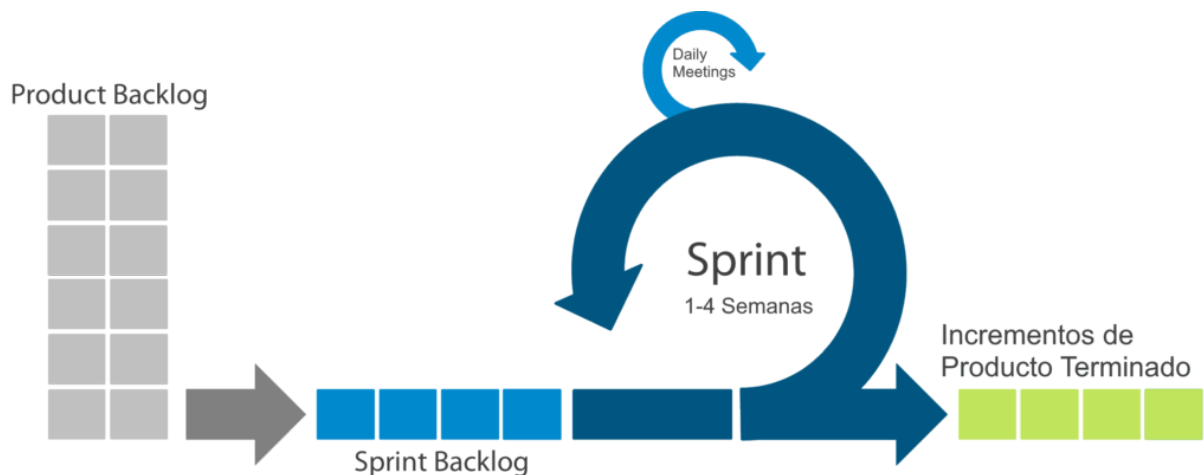


Figura 4: Procés Scrum

Els Sprints es van dividir de la següent forma (Figura 5):

Núm. Sprint	Data inici	Data final
1	3/03/2023	24/03/2023
2	24/03/2023	14/04/2023
3	14/04/2023	28/04/2023

4	28/04/2023	12/05/2023
5	12/05/2023	26/05/2023

Figura 5: Nombre de Sprints fets i dates

- **Sprint 1:**

En aquest primer Sprint, l'objectiu principal era acabar de deixar clar tots els requisits que s'havien d'afegir a l'aplicació i familiaritzar-me amb el codi amb el qual no havia tingut contacte prèviament.

Per aquesta raó es va decidir no posar molta càrrega en nombre d'històries d'usuari a completar i tan sols es va decidir afegir una única història d'usuari a completar, en concret la *US2 Afegir sensació tèrmica a “informació de la ruta”* (Figura 6).

Primerament, es va haver de preparar l'entorn de treball per a executar l'aplicació amb React Native, el qual no havia tingut contacte abans i entrava de nou. Un cop l'entorn de treball estava llest per a executar l'aplicació, vaig procedir a familiaritzar-me amb el codi abans de començar a desenvolupar la història d'usuari seleccionada.

Núm. User Story	Títol
US2	Afegir sensació tèrmica a “informació de la ruta”.

Figura 6: User stories del Sprint 1.

Aconseguir que l'aplicació funcionés en el meu entorn de treball va complicar-se una mica, però vaig solucionar-ho sense perdre molt de temps.

Per tal de familiaritzar-me amb el codi, vaig canviar la base de dades i la vaig col·locar a Google Cloud Platform, per a facilitar fer-la servir des de diferents estacions de treball i vaig falsejar les respostes de les API de meteorologia i contaminació, per tal de poder fer proves necessàries sense cap mena de problema amb els límits de les peticions disponibles (explicats a l'apartat [5. Cost del Projecte](#), dins la secció de desenvolupament).

Un cop familiaritzat, vaig afegir l'opció de la sensació tèrmica a “informació de la ruta” sense gaire problema.

- **Sprint 2:**

Ja en el segon Sprint, es va decidir centrar-nos pensar com es guardaria la nova informació que volem obtenir amb la creació del mòdul de ciència ciutadana, en concret el que volíem era realitzar la part del backend de les històries d'usuari seleccionades (*US3 Puntuar (i comentar) una ruta*, *US6 Completar reptes*, *US9 Donar retroacció sobre una ruta en forma de text*).

Núm. User Story	Títol
US1	Modificar codi de colors a “Informació de la ruta”.

US3	Puntuar (i comentar) una ruta.
US6	Completar reptes.
US8	Visualitzar advertències.
US9	Donar retroacció sobre una ruta en forma de text.

Figura 7: User stories del Sprint 2.

Un cop les noves taules creades i els endpoints de l'API REST també creats (explicats a [6.2.1 Estructura del codi](#), secció Models i [6.4 Endpoints](#) respectivament), es van realitzar tant la **US1** *Modificar codi de colors a "Informació de la ruta"*, com la **US8** *Visualitzar advertències* i així deixar tant la secció d'"Informació de la ruta" com les modificacions per afegir la sensació tèrmica enllestides.

- **Sprint 3:**

En aquest tercer Sprint, ens vam centrar a acabar algunes de les històries d'usuari iniciades al Sprint anterior, en concret la **US3** *Puntuar (i comentar) una ruta* i la **US6** *Completar reptes*, fent així la part de frontend que mancava perquè fossin completes.

Un cop va estar llest es va procedir a implementar el fet de poder acabar una ruta automàticament en completar-la (**US5** *Acabar una ruta automàticament*) i l'historial de puntuacions (**US4** *Veure rutes puntuades*).

Núm. User Story	Títol
US3	Puntuar (i comentar) una ruta.
US4	Veure rutes puntuades.
US5	Acabar una ruta automàticament.
US6	Completar reptes.

Figura 8: User stories del Sprint 3.

- **Sprint 4:**

Aquest Sprint va servir per a començar a abordar la secció del mòdul de ciència ciutadana relacionat amb la retroacció a les rutes, per aquesta raó es va iniciar amb la **US9** *Donar retroacció sobre una ruta en forma de text*.

Un cop va estar creat el modal amb les opcions per escollir quin tipus de retroacció es volia donar i amb l'opció de poder afegir un comentari, vam acabar d'abordar els reptes, fent la **US7** *Veure nombre de reptes completats* i es va afegir també la capacitat de guardar una ruta que hagi estat iniciada amb el filtre de ruta "**Començar aquí**" (**US11** *Guardar ruta personalitzada*).

Núm. User Story	Títol
US7	Veure nombre de reptes completats.
US9	Donar retroacció sobre una ruta en forma de text.
US11	Guardar ruta personalitzada.

Figura 9: User stories del Sprint 4.

- **Sprint 5:**

En aquest últim Sprint, es van acabar de fer les últimes històries d'usuari, en concret es va afegir la funcionalitat de poder afegir una imatge en comptes d'un comentari al modal que permet fer la retroacció sobre la ruta (**US10** *Donar retroacció sobre una ruta en forma d'imatge*). També es va afegir l'avís que indica a l'usuari en iniciar una ruta que pot participar en el mòdul de ciència ciutadana (**US12** *Mostrar avís de ciència ciutadana*) i l'avís que indica a l'usuari si la ruta ha estat correcta (**US13** *Mostrar avís de ruta correcte*).

Núm. User Story	Títol
US10	Donar retroacció sobre una ruta en forma d'imatge.
US12	Mostrar avís de ciència ciutadana
US13	Mostrar avís de ruta correcte

Figura 10: User stories del Sprint 5.

A part, també es va dedicar aquest últim Sprint a acabar d'arreglar alguns bugs que anaven sortint a l'hora de fer proves, deixar llesta la part pràctica d'aquest Treball Final de Grau i iniciar l'escriptura d'aquesta memòria.

Aquests Sprints corresponen al temps de desenvolupament del codi, però també, sobretot al principi, es va invertir molt de temps en l'estudi de les tecnologies, en particular a investigar com funcionava React Native, una plataforma que no havia utilitzat prèviament i a la que m'havia d'adaptar. Pel que durant les dues setmanes anteriors a començar els sprints i durant una gran part del primer sprint es va dedicar a l'estudi i recopilació d'informació de les tecnologies que s'anaven a fer servir.

4.2. Kanban

Durant el desenvolupament, igual que amb el treball anterior [1], també es va decidir fer ús de la metodologia Kanban.

Kanban ens permet dividir cada història d'usuari en diferents tasques més individualitzades amb l'objectiu de poder visualitzar i fer un seguiment del progrés dels Sprints d'una forma més detallada alhora que ajudava a definir el treball en progrés i el que faltava per fer. També ens ajuda a poder separar el treball, tal com s'ha fet amb alguna història d'usuari, que la part del backend i la del frontend entren a dos Sprints diferents.

El tauler Kanban el tenia localitzat al repositori de GitHub del treball. Aquest tauler disposava de les següents columnes:

- **To Do:** Llistat de tasques que queden per fer.
- **In Progress:** Llistat de tasques que s'estan fent actualment
- **MR - DEV:** Tasques acabades i fusionades a l'entorn de DEV, on és testeant amb la resta de codi que ja estava acabat.
- **MR - STG:** Tasques fetes i fusionades a l'entorn de STG, on estan totes les tasques ja provades i la versió de codi està llesta per ensenyar a les tutores.
- **Deployed:** Tasques que estan en l'entorn final PRO i han estat desplegades després del vist bo de les tutores.

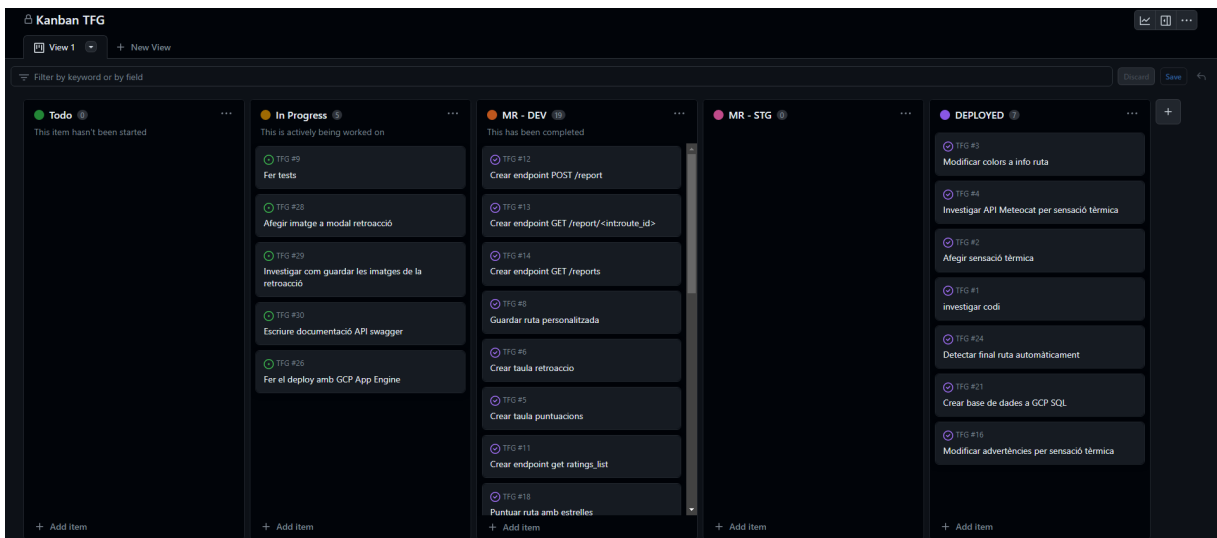


Figura 11: Tauler kanban durant l'Sprint 5

4.3. Diagrama de Gantt

Un diagrama de Gantt és una eina útil per planificar projectes. Aquest proporciona una vista general de les tasques programades, on totes les parts implicades sabran quines tasques s'han de completar i en quina data.

En la figura 12 es pot apreciar el diagrama format per totes les històries d'usuari realitzades durant els Sprints, per tal de poder visualitzar millor el temps que ha requerit cadascuna d'elles.

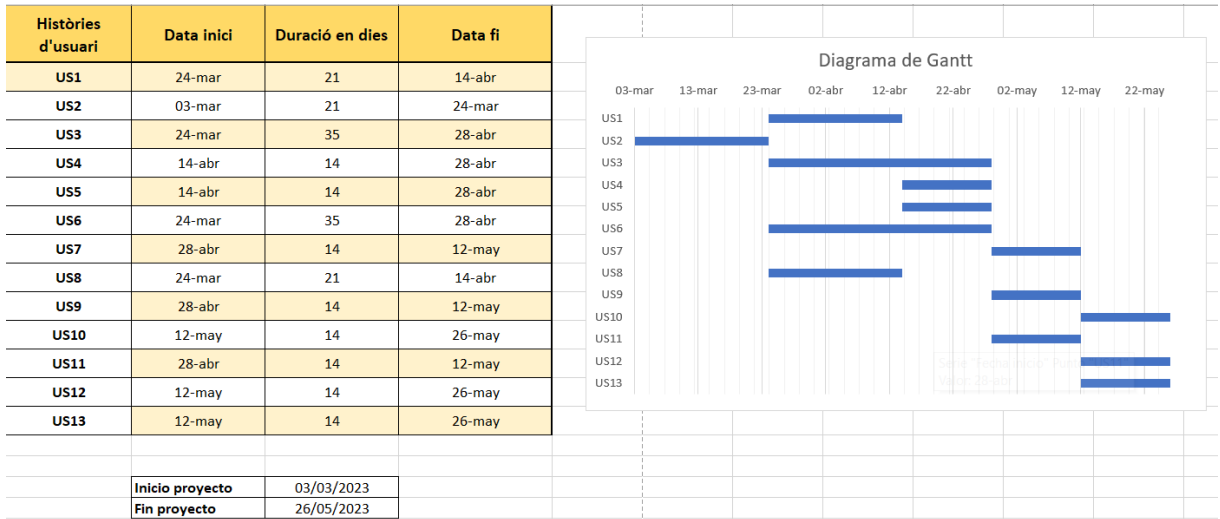


Figura 12: Diagrama de Gantt

5. Cost del projecte

En aquest apartat es du a terme una aproximació dels costos econòmics que suposaria fer aquest treball a escala empresarial.

Durant el desenvolupament de l'aplicació totes les tecnologies han estat fetes servides amb plans de dades gratuïts, i per això, com ara veurem, el cost de desenvolupament ha sortit gratuït. En canvi, en el moment de voler posar l'aplicació en producció, s'ha de buscar una alternativa per a tenir el contingut al núvol i sempre actiu, i amb uns plans de dades de les API superiors, ja que s'espera un nombre de peticions més elevat que en desenvolupament.

- Desenvolupament:

- **Base de dades MySQL [10] a Azure:** S'ha fet ús dels 100 dòlars gratuïts que proporciona el compte d'estudiant de la UB a Azure a poder provar la seva plataforma, per tant, ha ocasionat un cost de **0 €**.
- **Host de l'API:** Es va fer servir sempre en local durant el desenvolupament el servidor per a fer les proves necessàries en ser un sol desenvolupador, per tant, no hi havia necessitat de tenir el servidor al núvol, en conseqüència, té un cost de **0 €**.
- **Mapbox [8]:** Servei gratuït.
- **API contaminació (Ambee) [6]:** Límit de **500 peticions** amb el pla gratuït de trenta dies.
- **API meteorologia (Meteocat) [7]:** Igual que en el treball anterior [1], es va sol·licitar accés a l'API de *Meteocat* com a estudiant i em van facilitar accés amb **100 peticions al mes**.

Per a tenir el procés de desenvolupament completament gratuït, cal tenir en compte els requisits d'Azure per als crèdits gratuïts i els límits de les API, que per a no arribar-hi, es van crear valors de contaminació i meteorologia falsejats.

- Producció:

Cal mencionar que aquests costos són estimats per un ús de l'aplicació de 500 persones al dia, per tant, aquests costos poden variar depenent de l'ús que tingui l'aplicació. Per a calcular els preus de la base de dades MySQL d'Azure i el host a App Service, s'han calculat fent servir la calculadora que proporciona Azure [5].

- **Base de dades MySQL [10] a Azure:** En aquest cas no es podria fer ús del servei gratuït de prova i s'hauria de contractar una instància de la base de dades de tipus General Purpose i un nivell de computació D2as v4, que té 2 vCPU i un preu de **96,93 € al mes**, amb 50 GB d'emmagatzematge SSD.
- **Host de l'API a App Service [11] (Azure):** En aquest cas no podem tenir el servidor en local i caldria tenir-lo sempre actiu al núvol, cosa que es pot fer

contractant un altre servei d'Azure, en aquest cas App Service, on podem contractar una instància del tipus B2 per **32,61 € al mes**.

- **Mapbox [8]:** Mapbox continua sent un servei gratuït, per tant, suposa un cost de **0 €**.
- **API contaminació (Ambee) [6]:** La web de l'API no proporciona informació del cost, s'ha de contactar amb ells per explicar la intenció de l'aplicació i que et facin un pressupost. Aproximadament aquest pressupost surt a més de **500 € al mes**, ja que és una API privada i s'ha de consultar múltiples vegades per al càlcul de la contaminació de les rutes.
- **API meteorologia (meteocat) [7]:** Sempre que es consideri un projecte estudiantil, l'accés es permet gratuïtament. Però si es fa servir de forma professional, suposa un cost de **79 € al mes** per a un límit de 100.000 peticions.

Així doncs, ens quedaria un preu mensual de 708,54 € per a mantenir els serveis necessaris per al correcte funcionament de l'aplicació en producció per una mitja de 500 usuaris.

També cal tenir en compte el preu de tenir l'aplicació a Google Play, on s'ha de pagar 25 € un únic cop per a poder disposar d'un compte de desenvolupador. En el cas d'iOS, es requereix un compte de desenvolupador també que té un cost anual de 99 €.

Finalment, també cal considerar el cost humà que suposa dur a terme aquest projecte. Caldria tenir en compte més tipus de treballadors a part de mi, on el meu salari seria de Junior (uns 8 € l'hora), per tant, sortiria a uns 15 € l'hora aproximadament. Tenint en compte que el temps de desenvolupament del projecte està estimat en catorze setmanes, i sense comptar el temps per a redactar la memòria, amb un treball de 6 hores diàries en 3,75 dies.

Així doncs, el cost humà del desenvolupament quedaria de tal manera: El cost setmanal seria de 337,5 €/setmana (15 €/h * 6 h/dia * 3,75 dies/setmana), el **cost mensual** seria de **1350 €/mes** (337,5 €/setmana * quatre setmanes/mes), i finalment, el **cost total** quedaria en **4725 €** (337,5 €/setmana * catorze setmanes).

A continuació, a la figura 13 podem veure una taula resum amb tots els costos de producció:

Tipus de despesa	Preu
Base de dades	49,27 €/mes
API i servidors	708,54 €/mes
Google Play i App Store	115 € primer any, 99 € la resta d'anys → 8,5 €/mes (aproximació global)

Cost humà	4725 € en total → 1350 €/mes
Total	2116,31 €/mes

Figura 13: Cost de l'aplicació en producció

6. Disseny i implementació

Seguint el treball anterior [1], el desenvolupament de l'aplicació s'ha realitzat fent servir la tecnologia React Native per al frontend de l'aplicació i Python amb Flask per al backend.

En aquest apartat, primerament s'explica l'estructura del projecte i com funciona React Native, i a continuació es mostren els models de la base de dades, amb l'estructura que aquests comporten i els endpoints dels quals disposem per connectar amb l'API des de l'aplicació.

6.1. Arquitectura frontend

Per tal d'estructurar el codi a React Native per a fer la part del frontend de l'aplicació, s'ha seguit arquitectura que es va definir al treball anterior [1].

Aquesta arquitectura és la generalment recomanada dins del desenvolupament mòbil amb React Native, que consisteix a separar els fitxers principalment en tres capes:

- 1. UI o Capa de presentació:** Aquesta capa consta de tots els components, on un component és un element de la interfície d'usuari amb els quals l'usuari interactua.
- 2. Capa lògica:** Aquesta és la capa responsable de mantenir la lògica de l'aplicació, de gestionar els esdeveniments i les interaccions amb la capa de presentació.
- 3. Capa API:** Finalment, aquesta capa és la responsable de gestionar totes les interaccions amb el backend. És on l'aplicació fa les trucades a l'API, a la base de dades i als serveis web externs per a obtenir la informació sobre la previsió meteorològica, contaminació i sensació tèrmica.

Per tal de realitzar aquesta separació en tres capes, les carpetes del codi segueixen la següent organització (Figura 14):

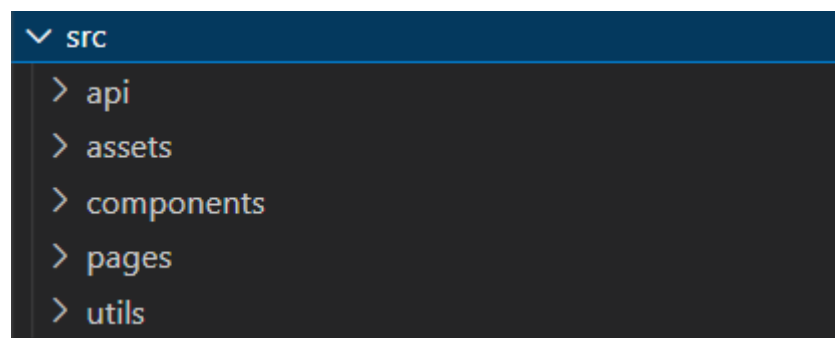


Figura 14: Estructura de carpetes a frontend

1. Components

Aquesta carpeta consta de tots els fitxers i carpetes on hi ha components reutilitzables. Es poden veure a la figura 15 els que s'han creat aquest treball ressaltats amb color blau.

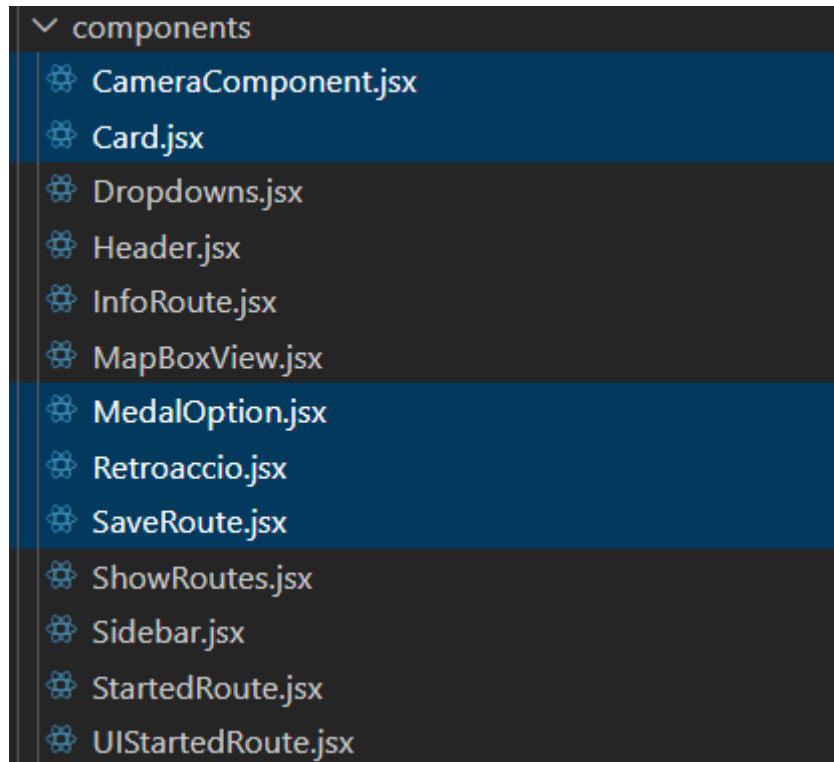


Figura 15: Components. En blau els creats aquest treball.

2. Assets

En aquesta carpeta es guarden tots els arxius estàtics, com ara poden ser les imatges o les icones que es facin servir a l'aplicació.

3. Pages

Aquí es guarden les pàgines de l'aplicació amb les quals interactua l'usuari, com ara la pàgina d'inici de sessió, la pàgina principal, la pàgina de registre, etc. Es pot veure les creades noves ressaltades de color blau, encara que també s'ha modificat "Main".

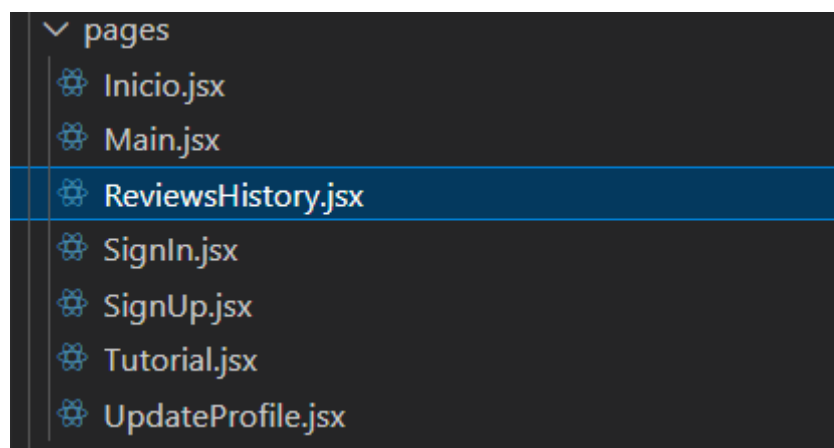


Figura 16: Pàgines creades

El conjunt de pàgines i components formen la capa de presentació de l'aplicació.

4. Api

Carpeta on són implementades les trucades a les API, tant la pròpia com les externes (meteorologia i contaminació). En blau es veuen els fitxers creats nous aquest treball, encara que s'han modificat la gran majoria.

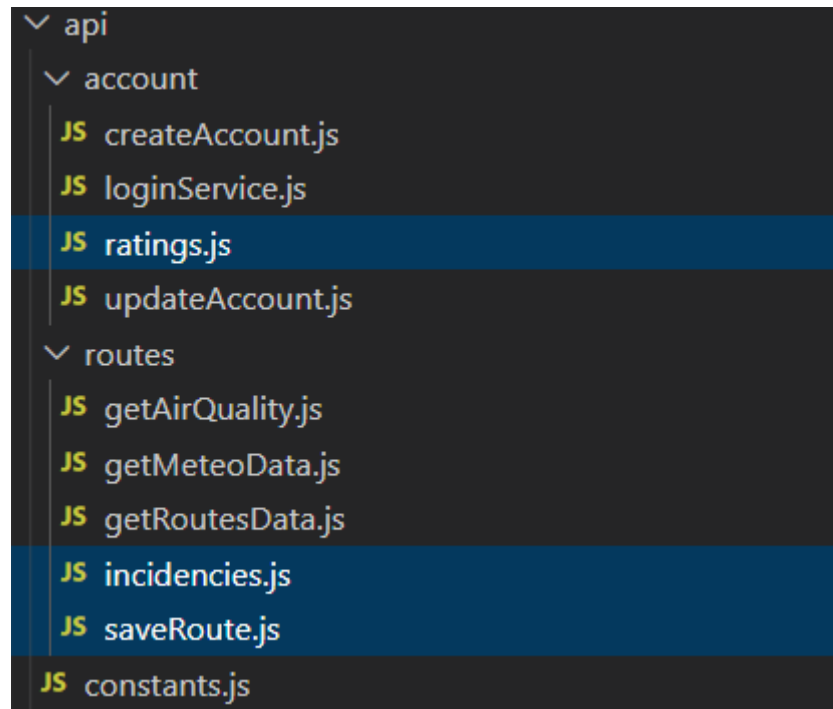


Figura 17: Fitxers de la capa API

Aquesta carpeta és l'encarregada d'implementar la capa API. Per a seguir una separació dels fitxers i una estructura més clara i ordenada, s'han separat les trucades per funcionalitat.

La primera carpeta “account” s'encarrega de tot el relacionat amb els usuaris, com ara crear un compte, iniciar sessió, actualitzar un compte existent o puntuar una ruta.

La segona carpeta “routes” conté la informació relacionada amb les rutes, com ara aconseguir les coordenades per mostrar rutes al mapa, l'obtenció de la temperatura i la sensació tèrmica, poder guardar una ruta o donar poder retroacció.

També hi ha un fitxer anomenat “constants.js”, que disposa de tota la informació que no varia mai, com ara els URL que serveixen per a connectar-se a l'API pròpia i les externes (meteorologia i contaminació) i els tokens d'accés necessaris per poder fer ús dels endpoints de les API externes.

5. Utils

Aquesta carpeta representa la capa lògica de l'aplicació. Els fitxers que hi apareixen són els encarregats de connectar la capa de presentació amb la capa d'API.

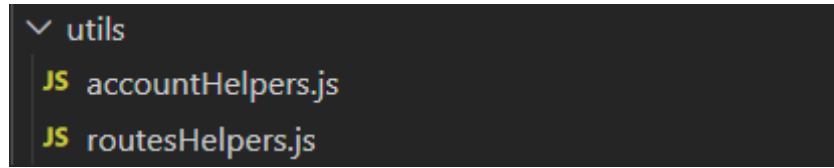


Figura 18: Fitxers de la capa lògica

En concret es disposa de dos fitxers (figura 18). El primer, “accountHelpers.js” s'encarrega de la lògica de dades relacionada amb l'usuari i la connexió amb els serveis de createAccount, loginService, updateAccount i ratings. El segon fitxer, “routesHelpers.js”, és l'encarregat de la lògica de dades relacionada amb les rutes. Concretament, s'encarrega de la connexió amb els serveis de retroacció i saveRoute, però també ajuden a calcular i obtenir tota la informació que necessiten els components per a mostrar les rutes i preparar els paràmetres a enviar als diferents endpoints de l'API relacionats amb les rutes.

En resum, tal com ja es va mencionar al treball anterior [1], seguir aquesta estructura va ser l'opció adequada, ja que continuar amb el desenvolupament de l'aplicació ha estat molt més senzill seguint un patró de disseny, en aquest cas el patró Capa de presentació - Capa lògica - Capa API, que si tot el codi hagués estat junt i sense cap separació.

6.1.1. Programació amb React-Native

Tal com s'ha mencionat a l'apartat [2.2 Treball fet prèviament](#), React Native és una popular plataforma de desenvolupament d'aplicacions mòbils que permet construir aplicacions per a iOS i Android utilitzant JavaScript i React. Una part important de React Native són els components, que són elements reutilitzables que permeten construir la interfície d'usuari de l'aplicació. Els cinc components més utilitzats a React Native són:

- **View**: És el component més bàsic i fonamental de React Native. Serveix per crear caixes (containers) que actuen com a contenidors per a altres components o per a elements de UI. La vista és comparable a un div d'HTML i proporciona funcions bàsiques de disseny, com ara disposició flex, estils, mida i posicionament.
- **Text**: Aquest component s'utilitza per a mostrar text. Permet personalitzar l'estil del text, com la font, el color, la grandària, etc. També és compatible amb estils en línia i CSS-like.
- **Image**: El component Image s'usa per mostrar imatges. Pot carregar imatges locals o remotes i proporciona funcionalitats com la càrrega progressiva, el control de la mida de la imatge i la gestió dels errors de càrrega.
- **TextInput**: Aquest component permet als usuaris introduir text en una aplicació React Native. Es fa servir per capturar dades d'entrada com ara noms d'usuari, contrasenyes, missatges, etc. Proporciona diverses opcions de personalització, com ara l'estil del text dins de la caixa de text, els controladors d'esdeveniments i la validació de dades d'entrada.
- **ScrollView**: A diferència de View, el component ScrollView permet desplaçar i fer scroll dels seus continguts quan són més grans que la mida de la pantalla. És útil quan es té un contingut llarg, com una llista o una pàgina amb molta informació. També proporciona controladors d'esdeveniments per gestionar l'acció de desplaçament.

Aquests són només alguns dels components més utilitzats a React Native. Hi ha molts altres components disponibles a la llibreria estàndard de React Native, així com a components personalitzats desenvolupats per la comunitat.

6.1.1.1. React Native Async Storage

En tenir una aplicació que consumeix dades de diverses API externes amb un pla de peticions limitades, el que es vol fer consisteix a guardar les respostes que dona l'API després d'haver fet una primera consulta, i així en la pròxima petició, en comptes de llançar la petició a l'API, es deriva a la memòria cau del dispositiu. D'aquesta forma es millora el rendiment de l'aplicació, ja que accedir a la memòria cau sempre és més ràpid que enviar una petició a una API.

Al treball anterior [1], per tal de dur a terme aquesta tasca es va decidir utilitzar React Native Async Storage, un sistema d'emmagatzematge de clau-valor no xifrat, asíncron, persistent i global per l'aplicació, però que no consta d'una funció d'expiració de les dades.

Així doncs, es va decidir implementar aquesta funcionalitat amb una data d'expiració de les dades de 24 h, però no va acabar de funcionar i, per tant, en aquest treball també he millorat aquest aspecte.

Vaig decidir canviar el temps que es guarden les dades de 24 h a només 6 h, ja que no és el mateix sortir al matí a caminar que a la tarda o a la nit. A l'hora de moure-ho a producció es pot canviar a 1 h per a ser tan realistes com sigui possible amb les dades del moment però mantenint limitat el nombre d'accessos a les API.

A la figura 19 es pot veure com queda el codi del servei de crides a l'API de Meteocat, un cop implementada la data d'expiració per a mantenir les dades de meteorologia durant 6 h.

```
export const getForecastData = async (time) => {

  let response = null;
  let rainValue = null;
  let tempXafogor = null;

  await AsyncStorage.getItem('meteoData3', async (err, value) => {

    response = (JSON.parse(value));

    // there is data in cache && cache is expired
    if (response !== null && response['expireAt'] && new Date(response.expireAt) < (new Date())) {
      //clear cache
      AsyncStorage.removeItem('meteoData3');
      //update data to be null
      response = null;
    } else {
      console.log('read meteoData from cache');
    }
  });

  //update cache + set expire at date
  if (response === null) {
    console.log('cache new Date ');
    const options = {
      method: 'GET',
      url: URL_METEO_DATA,
      headers: { 'X-Api-Key': meteoToken, 'Content-type': 'application/json' },
    };
    try {
      response = await axios.request(options);
      rainValue = response.data.dies[time.day].variables.precipitacio.valor[time.hour].valor;
      tempXafogor = response.data.dies[time.day].variables.tempXafogor.valors[time.hour].valor;
      let expireAt = getExpire6h()
      const forecastData = JSON.stringify({ expireAt, rainValue, tempXafogor });

      AsyncStorage.setItem('meteoData3', forecastData);

      return forecastData;
    } catch (error) {
      console.error(error.message);
      return null;
    }
  }
  return response;
}
```

Figura 19: Implementació servei Meteocat amb memòria cau i data d'expiració

6.1.2. Diagrama de components

El diagrama de components que queda finalment a l'aplicació és el següent, on podem apreciar la distribució dels components de les pàgines i l'ordre en el qual s'instancien. Igual que al diagrama de casos d'ús (figura 2), les pàgines noves i els nous components es poden veure de color vermell, a diferència dels ja existents que apareixen de color negre (figura 20):

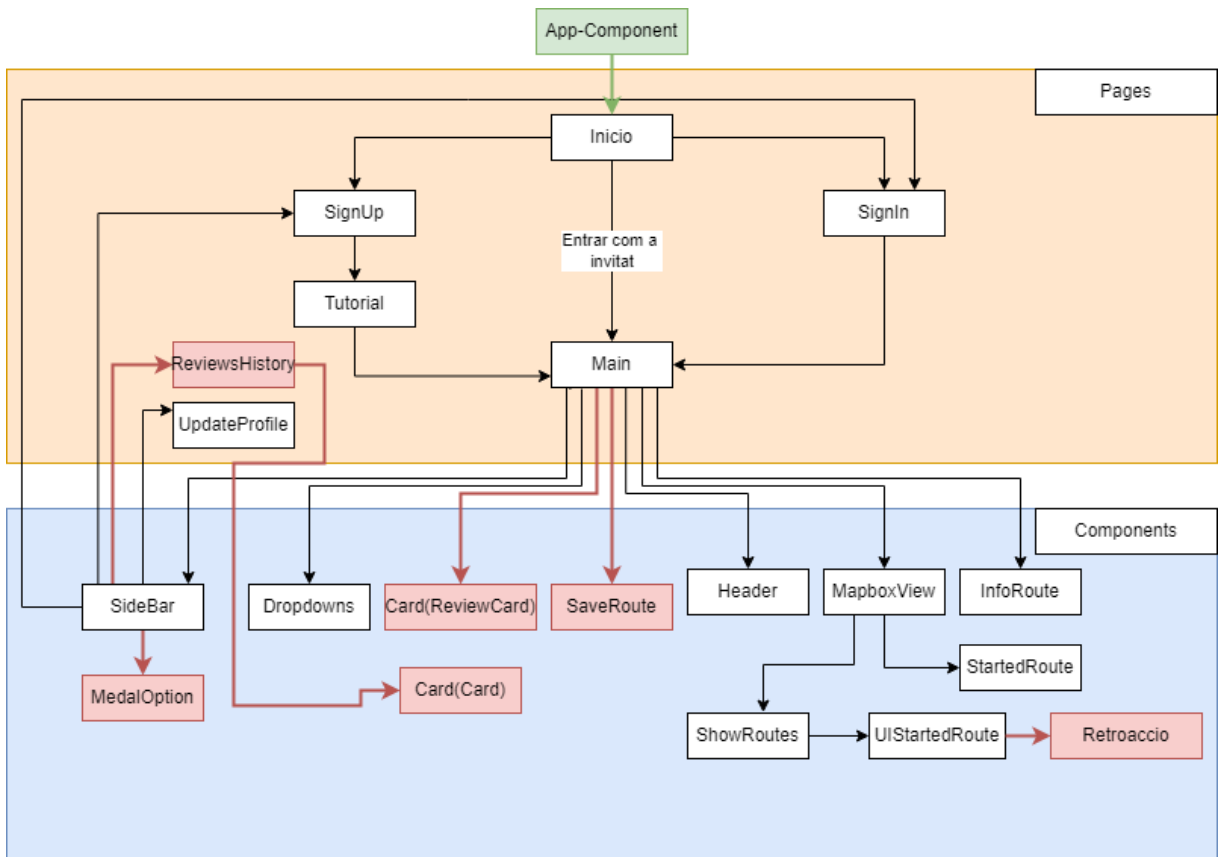


Figura 20: Diagrama de components i pàgines de l'aplicació

El diagrama el trobem dividit en dues seccions principals: “Pages” i Components. A la secció “Pages”, hi trobem totes les pantalles a les quals l'usuari pot navegar:

- 1) **Inici:** Primera pantalla, amb la que ens trobem en obrir l'aplicació. Ens permet escollir si volem iniciar sessió, registrar-nos o entrar com a invitat.
- 2) **SignUp:** Pantalla on podem registrar-nos a l'aplicació creant un compte personal.
- 3) **SignIn:** Pantalla on podem iniciar sessió amb un compte ja existent.
- 4) **Tutorial:** Un cop ens registrem, es mostra aquesta pantalla, on trobem una guia de com funciona l'aplicació.
- 5) **UpdateProfile:** Pantalla la qual ens permet modificar les dades del nostre compte personal.

- 6) **Main:** Pantalla principal de l'aplicació. S'hi accedeix un cop iniciem sessió, s'acaba el tutorial o entrant com a invitat.
- 7) **ReviewsHistory:** Pantalla que ens permet visualitzar l'historial de puntuacions realitzades pel compte personal.

A la secció “Components” hi trobem tots els components que ajuden a visualitzar l'aplicació i les seves funcionalitats. Funcionen complementant les pantalles. Primerament, es llisten els components ja existents (la majoria han estat modificats):

- 1) **SideBar:** Menú lateral on podem fer diverses coses en funció de si tenim la sessió iniciada o no. En cas negatiu, tenim l'opció de registrar-nos o iniciar sessió. En cas positiu, podem veure el nombre de reptes completats o l'historial de puntuacions que s'han realitzat. També tenim l'opció de modificar el perfil i de tancar sessió.
- 2) **Dropdowns:** Component amb el qual podem seleccionar les opcions per filtrar les rutes a buscar.
- 3) **MapboxView:** Component on es visualitza el mapa utilitzant Mapbox i sobre el que es visualitzaran les rutes.
- 4) **InfoRoute:** Un cop una ruta és seleccionada, aquest component és l'encarregat de mostrar la informació del recorregut com ara la distància, el nivell de contaminació, la temperatura o la sensació tèrmica.
- 5) **ShowRoutes:** Component que mostra sobre la vista del mapa (MapboxView) totes les rutes disponibles amb un codi de colors segons contaminació, previsió meteorològica i sensació tèrmica.
- 6) **StartedRoute:** Gràcies a aquest component podem visualitzar el mapa quan comencem el recorregut.
- 7) **UIStartedRoute:** Juntament amb el component anterior, quan la ruta ha començat, aquest component és l'encarregat de proporcionar-nos una interfície amb la qual interactuar per a poder veure el temps que ha passat des que ha començat, poder parar la ruta, sortir o donar retroacció sobre aquesta.

A continuació es llisten els components nous:

- 8) **Card (ReviewCard):** Ens permet puntuar una ruta d'1 a 5 estrelles i deixar un comentari opcional al respecte.
- 9) **Card (Card):** Component que serveix per a veure una puntuació donada. A la “page” de ReviewsHistory podem veure tants cops aquest component (en una llista) com cops hagi puntuat l'usuari rutes.
- 10) **MedalOption:** Component que ens indica quantes rutes hem completat la setmana actual juntament amb un missatge motivador per a continuar complint reptes.

- 11) **SaveRoute:** Aquest component ens mostra un modal que ens deixa guardar la ruta que hem acabat de fer si aquesta s'ha generat des de l'opció del filtre de ruta "Comença aquí".
- 12) **Retroaccio:** Finalment, aquest component ens permet donar retroacció a una ruta, podent escollir el tipus de la retroacció i afegir un comentari o una fotografia opcional.

Cal tenir en compte diversos aspectes del diagrama (figura 20), com ara que totes les relacions de les pàgines són 1 a 1, ja que aquest sistema utilitza les pàgines com un únic objecte. El cas dels components és diferent, perquè aquests sí que poden ser instanciats diverses vegades, com és el cas de la relació 1 a N entre MapboxView i ShowRoutes, perquè es poden veure diverses rutes sobre el mateix mapa o la relació 1 a N entre ReviewsHistory i Card (Card), ja que dins de ReviewsHistory podem arribar a trobar tants Cards com cops hagi puntuat l'usuari les rutes.

Com es pot veure, "App-Component" està fora de la secció de "pages", el qual s'executa i mostra per defecte a React Native i en aquest hi trobem el menú de l'aplicació que ens permet la navegació entre pàgines. La ruta inicial està assignada a la pàgina d'Inici des de la qual podrem anar a la resta de pantalles.

En l'apartat [7.1 Navegació](#) es podrà veure amb imatges la navegació completa entre pàgines i com aquestes interactuen amb els components.

6.2. Arquitectura backend

Per a realitzar la part del backend, s'ha fet servir l'organització de codi vista durant el grau, fent servir Flask, tal com es va decidir al treball anterior [1]. Aquesta arquitectura consisteix a separar per un costat els models de dades encarregats de definir l'estructura dels elements amb els quals treballa l'API (models), en aquest cas les rutes, els usuaris, les puntuacions de les rutes i les incidències per a guardar-los a la base de dades. Per un altre costat, hi ha els “resources”, encarregats de definir els endpoints que proporciona l'API i implementar els mètodes HTTP que aquests utilitzen i interactuen amb els models.

6.2.1. Estructura del codi

L'estructura del codi quedaria de tal manera (figura 21):

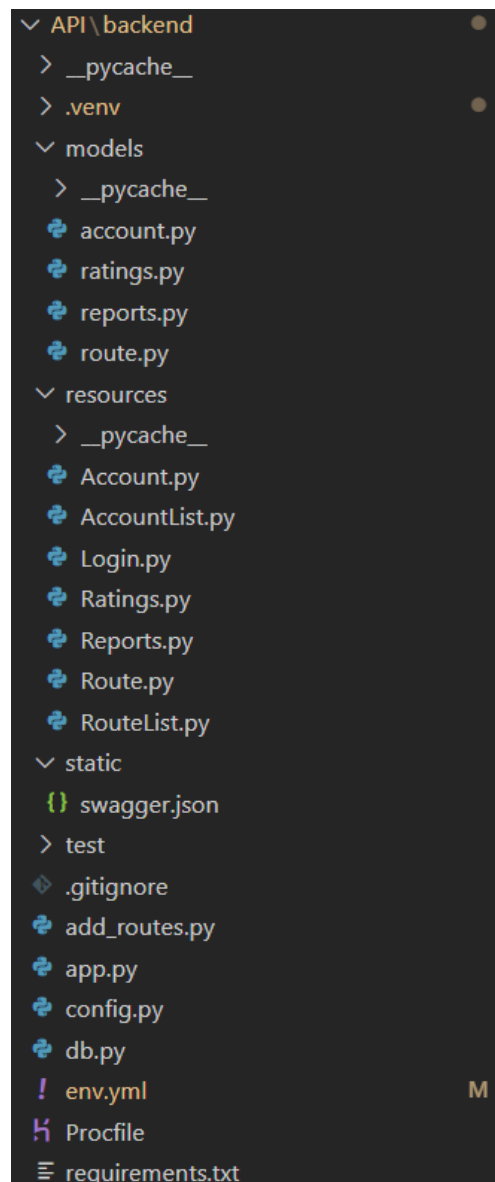


Figura 21: Estructura de fitxers del backend

Primerament, podem veure que tal com s'ha comentat, a la figura 21 podem veure que està dividit en diferents seccions:

- 1) **Models:** Un model ens dona l'estructura sota la qual operarà la lògica de l'aplicació, per tant, aquí és on donem un sentit a les diferents entitats amb les quals hem de comptar per poder exercir les nostres accions. Els models ens serveixen per a poder definir les nostres bases de dades fent ús de classes a Python.

A continuació es mostren les modificacions realitzades sobre els models ja existents (usuaris i rutes), i l'estructura dels que s'han creat per a abordar la ciència ciutadana.

- **Usuaris:** En aquest model (figura 22) ja es guardaven prèviament els següents camps:
 - **id:** identificador únic de l'usuari. És únic.
 - **email:** Correu de l'usuari.
 - **name:** Nom i cognoms de l'usuari.
 - **edat:** Edat de l'usuari.
 - **password:** Contrasenya amb què l'usuari s'ha registrat. Es guarda encriptada a causa de la protecció de dades.

I s'han realitzat les següents modificacions:

- **n_completed_routes:** Nombre de rutes completades per l'usuari en el període de temps d'una setmana. S'inicialitza a 0.
- **n_saved_routes:** Nombre de rutes guardades per l'usuari amb un màxim de 3. S'inicialitza a 0.
- **last_reset_date:** Data de l'última vegada que es va reiniciar el camp de *n_completed_routes*.

```
class AccountsModel(db.Model):
    __tablename__ = 'users'
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(100))
    name = db.Column(db.String(100))
    edat = db.Column(db.Integer)
    password = db.Column(db.String(400))
    n_completed_routes = db.Column(db.Integer, default=0)
    n_saved_routes = db.Column(db.Integer, default=0)
    last_reset_date = db.Column(db.Date, default=datetime.date.today())

    def __init__(self, name, email, edat, password):
        self.name = name
        self.email = email
        self.edat = edat
        self.password = password
```

Figura 22: Model de dades dels usuaris

- **Rutes:** En aquest model (figura 23), **no s'han realitzat canvis des del treball anterior** [1]. Els camps que es guarden són els següents:
 - **id:** Identificador únic de cada ruta.

- **name:** Nom de la ruta.
- **level:** Nivell de dificultat que té la ruta.
- **activity:** Tipus d'activitat per la qual està destinada la ruta. Passeig i/o córrer.
- **distance:** Distància en km del recorregut.
- **type:** Indica la forma de la ruta. Pot ser circular o d'anada i tornada.
- **coordinates:** Llistat de totes les coordenades que formen el recorregut.
- **elevation:** Camp on s'indica l'elevació de les rutes si n'hi ha.
- **health_type:** Tipus de ruta. "Normal" per les de proposta inicial i "Zones verdes" per les de Proposta verda.

```
class RoutesModel(db.Model):
    __tablename__ = 'routes'

    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100))
    level = db.Column(db.String(100))
    activity = db.Column(db.String(100)) # Paseo/correr
    distance = db.Column(db.Float)
    type = db.Column(db.String(100)) # Circular/ida y vuelta
    coordinates = db.Column(db.Text())
    elevation = db.Column(db.Float)
    health_type = db.Column(db.String(100))

    def __init__(self, name, level, activity, distance, type, coordinates, elevation, health_type):
        self.name = name
        self.level = level
        self.activity = activity
        self.distance = distance
        self.type = type
        self.coordinates = coordinates
        self.elevation = elevation
        self.health_type = health_type
```

Figura 23: Model de dades de les rutes

- **Puntuacions:** En aquest model (figura 24) es defineixen i es guarden els següents camps:
 - **id:** Identificador únic de cada puntuació.
 - **user_id:** Identificador de l'usuari que ha realitzat la puntuació.
 - **route_id:** Identificador de la ruta que s'ha puntuat.
 - **stars:** Nombre d'estrelles que s'ha donat a la ruta. Va d'1 estrella a 5.
 - **comment:** Comentari opcional per a donar més context sobre la puntuació donada.
 - **date:** Data en la qual s'ha dut a terme la puntuació.

```

class RatingsModel(db.Model):
    __tablename__ = 'ratings'

    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey(
        'users.id'), nullable=False)
    route_id = db.Column(db.Integer, db.ForeignKey(
        'routes.id'), nullable=False)
    stars = db.Column(db.Integer, nullable=False)
    comment = db.Column(db.String(500))
    date = db.Column(db.DateTime(), nullable=False, server_default=func.now())

    def __init__(self, user_id, route_id, stars, comment):
        self.user_id = user_id
        self.route_id = route_id
        self.stars = stars
        self.comment = comment
        self.date = datetime.utcnow()

```

Figura 24: Model de dades de les puntuacions

- **Retroaccions:** En aquest model (figura 25) es defineixen i es guarden els següents camps:
 - **id:** Identificador únic de cada retroacció.
 - **route_id:** Identificador de la ruta sobre la qual s'ha realitzat la retroacció.
 - **report_type:** Tipus de retroacció. (Ex: ruta tancada, en mal estat, etc.)
 - **comment:** Comentari opcional per a donar més context sobre la retroacció portada a cap.
 - **coordenadas:** Coordenades en la que s'ha fet la retroacció.
 - **image_url:** URL de la imatge a *Azure Blob Storage* [13]
 - **date:** Data en la qual s'ha fet la retroacció.

```

class ReportModel(db.Model):
    __tablename__ = 'reports'

    id = db.Column(db.Integer, primary_key=True, autoincrement=True)
    route_id = db.Column(db.Integer, db.ForeignKey(
        'routes.id'), nullable=False)
    report_type = db.Column(db.String(100), nullable=False)
    comment = db.Column(db.String(500))
    coordenadas = db.Column(db.Text())
    image_url = db.Column(db.String(150))
    date = db.Column(db.DateTime(), nullable=False, server_default=func.now())

    def __init__(self, route_id, report_type, comment, coordenadas, image_url):
        self.route_id = route_id
        self.report_type = report_type
        self.comment = comment
        self.coordenadas = coordenadas
        self.image_url = image_url
        self.date = datetime.utcnow()

```

Figura 25: Model de dades de les retroaccions

Al final de l'apartat [6.3 Base de dades](#) es pot veure el model relacional de tota la base de dades.

- 2) Resources:** Els resources són els fitxers encarregats de definir els mètodes HTTP GET, POST, PUT, DELETE i com aquests interactuen amb la base de dades i retornen les respostes a les sol·licituds entrants.

En aquest cas, s'han mantingut els resources referents als models “account” i “routes”, afegint les modificacions pertinents al resource de “account” per a poder tractar la nova informació que s'ha afegit al model i s'han creat els nous resources per als nous models “ratings” i “reports”.

En el cas de “ratings”, s'han creat els resources “Ratings”, que ens permet l'obtenció (GET), creació (POST) d'un “rating” i “RatingsList”, on hi ha definit només el mètode d'obtenció (GET) que ens retorna una llista de tots els “ratings” de la base de dades amb tota la informació del “rating”, com també de l'usuari i la ruta que estan relacionats.

En el cas de “reports”, se segueix una mateixa lògica. Hi ha un resource “Reports”, que ens permet l'obtenció (GET) i creació (POST) d'un “report” i “ReportsList”, que ens retorna el llistat complet de “reports” que hi ha a la base de dades amb tota la seva informació i la de la ruta sobre la qual està realitzat el “report”. També tenim “ReportsImage” que disposa d'un mètode POST per a guardar la imatge al núvol.

Adicionalment, podem observar a la figura 21 que tenim altres carpetes i fitxers que acaben de completar tota la part del backend i ajuden a complementar les parts principals:

- 3) Static:** Aquesta carpeta conté un fitxer *swagger.json* on hi ha definits els endpoints per a facilitar als usuaris l'enteniment de l'API. En l'apartat [6.4 Endpoints](#) s'explica en detall el swagger.
- 4) Test:** Aquesta carpeta conté els tests unitaris que hi havia al treball previ [1] i s'han ampliat per a afegir les comprovacions de les noves funcions disponibles a l'aplicació necessàries per a comprovar el correcte funcionament de l'API durant el desenvolupament.
Els tests fan ús dels endpoints creats enviant sol·licituds GET, POST, PUT i DELETE a l'API i comprovant que el resultat que retornen és l'esperat, de tal manera que un cop s'ha fet una modificació al codi del backend es pot comprovar que tot continuï funcionant correctament.
- 5) app.py:** Aquest és el fitxer principal de l'API, des del qual s'inicia l'execució i on es defineixen tots els endpoints possibles. S'utilitza com a controlador entre els models i els resources.
- 6) env.yml:** Aquest fitxer defineix els paquets necessaris per a crear un entorn de conda [12] i així sempre que s'hagi de desenvolupar l'aplicació des d'una localització diferent, pots fer ús d'aquest fitxer per a crear un entorn sempre idèntic i evitar problemes de software i versions de paquets diferents.

6.3. Base de dades

Com ja s'ha explicat prèviament en l'apartat [6.2.1 Estructura del codi](#), s'han definit 4 models de dades per a guardar la informació relacionada amb les rutes, els usuaris, les retroaccions i les puntuacions.

A diferència del treball anterior [1], ara la taula “users” consta de les noves columnes `n_completed_routes`, `n_saved_routes` i `last_reset_date`.

La taula “routes” s'ha mantingut sense cap modificació.

La taula “ratings” és l'encarregada de guardar les puntuacions que els usuaris hagin realitzat sobre les rutes. Per a poder representar això correctament, es va decidir afegir dues claus foranes que estiguin enllaçades amb la clau primària de les taules “routes” i “users”. En crear aquestes claus també es creen les relacions entre les 3 taules: una relació d'1 a N entre rutes i puntuacions, fent que 1 ruta pugui estar a N puntuacions i una relació d'1 a N entre usuaris i puntuacions, de tal manera que igual que passa amb les rutes, 1 usuari pugui estar a N puntuacions.

D'altra banda, la taula “reports” és l'encarregada de guardar les retroaccions fetes sobre les rutes. En aquest cas, aquestes no estan relacionades amb els usuaris i, per tant, són anònimes, però com també succeeix amb la taula “ratings”, tenim una clau forana amb la clau principal de la taula “routes” i una relació d'1 a N entre rutes i retroaccions, fent que 1 ruta pugui aparèixer a N retroaccions.

Respecte a l'estructura de la taula de retroaccions (“reports”), es va decidir que hi hagués una columna on s'especifiqués el tipus de retroacció, on a l'hora de guardar-ne una, hi ha els tipus preestablerts (“closed”, “bad”, “good”) i “altres”, on aquest “altres” és un text introduït per l'usuari, per tant, es facilita l'anàlisi d'aquestes retroaccions, que acaba sent un dels objectius de guardar aquesta informació. També es guarden les coordenades per a poder saber en quin punt de la ruta es troba l'usuari a l'hora de fer la retroacció, i en conseqüència poder localitzar el problema, igual que volem tenir més context sobre el problema, per tant, el comentari afegit per l'usuari es guarda a la columna *comment* o l'enllaç a la imatge facilitada es guarda a *image_url*. Addicionalment, ens interessa saber quan s'ha fet aquesta retroacció, per tant, guardem aquesta informació a la columna *date*.

Pel que fa a l'estructura de la taula de les puntuacions (“ratings”), volíem guardar la puntuació en forma d'estrelles (tal com es fa normalment) i un comentari en cas que l'usuari vulgui comentar algun aspecte de la ruta. En aquest cas també ens interessa saber quan s'ha realitzat la puntuació, per a poder veure més endavant si hi ha correlació entre les retroaccions i les puntuacions.

Tota aquesta informació la podem veure representada a la figura 26, on podem veure el diagrama relacional de la base de dades, on es poden apreciar les 4 taules (la informació nova de les taules existents i les noves taules es troben envoltades d'un requadre vermell), cadascuna amb els seus atributs, claus primàries, claus foranes i les relacions entre elles:

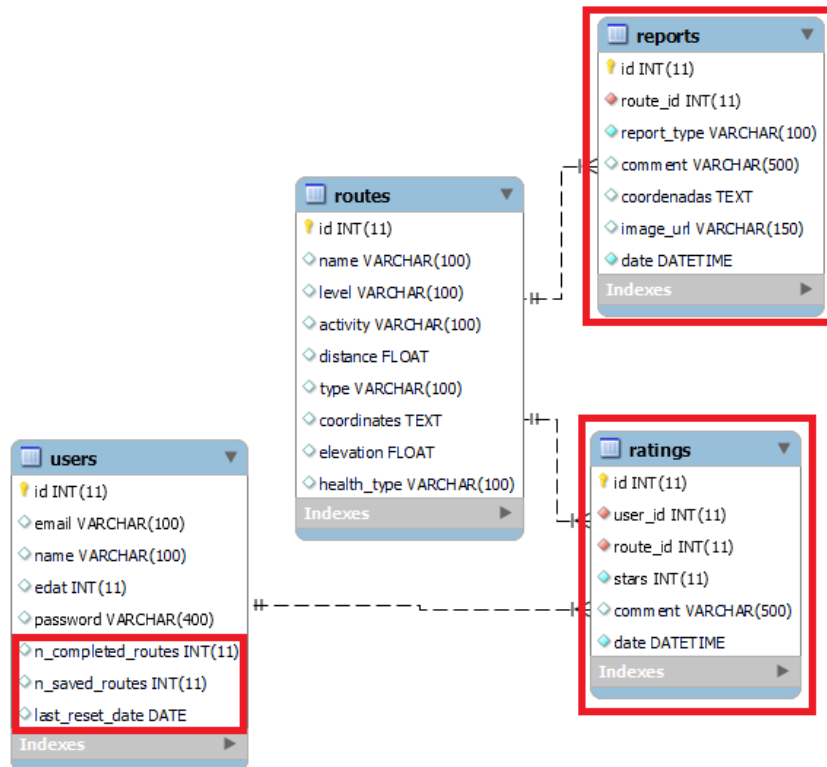


Figura 26: Diagrama relacional de la base de dades

6.3.1. Protecció de dades

La protecció de dades és un aspecte important a tenir en compte a l'hora de tractar amb dades personals de la població.

En el nostre cas, quan un usuari es registra se li demana el correu electrònic, el nom, l'edat i la contrasenya. En el cas de la contrasenya, aquesta és encriptada al moment de registrar-se i mai es tracta a l'aplicació de manera desencriptada.

Referent a les altres dades que es demanen, el correu, el nom i l'edat no són encriptats en cap moment, però no hi ha accés a aquestes per part de cap altra taula ni ningú extern a l'equip de desenvolupament ni ningú que no tingui accés a la taula users a aquestes dades.

En el cas de l'equip de recerca, no hauria de fer servir més que dades com l'edat o el nombre de rutes completades, que se li facilitarien sense comprometre les altres dades.

6.4. Endpoints

Els endpoints són els punts d'accés dels que disposa l'aplicació per a accedir a la base de dades mitjançant una API RESTful.

En el treball previ disposàvem de 7 endpoints [1], els quals ens permetien tractar amb usuaris i rutes. Ara s'ha afegit 7 endpoints addicionals que ens permeten tractar les retroaccions i les puntuacions (figura 27).

A continuació es llisten els 14 endpoints disponibles:

- 1) **POST /account:** Endpoint encarregat destinat a afegir usuaris nous a la base de dades. S'ha implementat amb un POST on s'envia al body de la petició totes les dades necessàries pels usuaris.
- 2) **GET o PUT /account/<string:email>:** Aquest segueix sent el mateix endpoint que abans, però s'utilitza pels mètodes HTTP GET i PUT. Aquest s'encarrega a partir de rebre per paràmetre el correu d'un usuari ja creat fer un GET per recuperar tota la informació de l'usuari. O també es pot realitzar un PUT per a 'actualització de l'usuari que s'indiqui, enviant la informació a actualitzar al body de la petició.
- 3) **GET /accounts:** Endpoint que retorna el llistat complet de les rutes que hi ha a la base de dades amb tota la informació de cadascuna d'elles.
- 4) **POST /login:** Endpoint per crear el login de l'usuari a partir de l'email i la contrasenya. A partir d'aquestes dades es verifica que l'usuari existeix a la base de dades i es retorna la informació de l'usuari.
- 5) **POST /route:** Endpoint encarregat d'afegir rutes a la base de dades si així es necessita, a partir d'un POST amb tota la informació necessària per a les rutes. Actualment, no s'utilitza, ja que com he explicat, totes les rutes s'afegeixen amb l'script `add_routes.py`. Però s'ha deixat creat per si és necessari en el futur.
- 6) **GET /route/<string:id>:** Endpoint que retorna la ruta a partir de l'identificador que té cadascuna.
- 7) **GET /routes:** Endpoint encarregat de retornar un llistat de rutes. Aquest llistat es pot filtrar segons les necessitats de l'usuari, a partir de tres paràmetres d'entrada; "health_type", "level" i "coords".
 - a) **health_type:** Tal com el nom indica, correspon amb el camp de la base de dades i s'utilitza per sol·licitar a l'API que retorni les rutes filtrades pel tipus de ruta (proposta inicial, proposta verda o automàtica).
 - b) **level:** El segon també correspon amb el camp level de la base de dades i fa que retorni rutes segons el nivell (fàcil, mitjà, difícil).
 - c) **coords:** Aquest camp s'utilitza conjuntament amb el health_type quan té el valor "auto" indicant que l'usuari vol obtenir una ruta automàtica. Llavors, en aquest camp s'envia les coordenades actuals de l'usuari per retornar la ruta a partir d'aquest punt. Per la resta de combinacions no s'utilitza.

Exceptuant el paràmetre “coords” que ja he comentat com i quan s'utilitza, els altres dos es poden utilitzar de forma individual o combinada. És a dir, si s'envia només health_type l'API retornarà les rutes filtrades per aquest camp i igual amb “level”, però si s'envien tots dos es retorna les rutes que compleixen totes dues condicions.

- 8) **POST /rating:** Endpoint encarregat destinat a afegir noves puntuacions a la base de dades. S'ha implementat amb un POST on s'envia al body de la petició totes les dades necessàries per a afegir una puntuació: user_id, route_id, stars, comment (opcional).
- 9) **GET /rating/<int:user_id>:** Endpoint destinat a obtenir el llistat de totes les puntuacions realitzades per l'usuari amb l'identificador que rep per paràmetre.
- 10) **GET /ratings:** Endpoint que retorna el llistat complet de les puntuacions que hi ha a la base de dades amb tota la informació de cadascuna d'elles.
- 11) **POST /report:** Endpoint encarregat d'afegir noves retroaccions a la base de dades. S'ha implementat amb un POST on s'envia al body de la petició totes les dades necessàries per afegir una puntuació: route_id, coordenades, report_type, comment (opcional).
- 12) **GET /report/<int:route_id>:** Endpoint encarregat d'aconseguir el llistat de totes les retroaccions realitzades sobre la ruta amb l'id (route_id) que es passa per paràmetre.
- 13) **GET /reports:** Endpoint que retorna el llistat complet de les retroaccions que hi ha a la base de dades amb tota la informació de cadascuna d'elles.
- 14) **GET /upload:** Endpoint que permet pujar una imatge al núvol.

```
@app.route("/")
def home():
    return "Hello, TFG!"

api.add_resource(Accounts, '/account/<string:email>', '/account')
api.add_resource(AccountsList, '/accounts')
api.add_resource(Login, '/login')
api.add_resource(Routes, '/route/<string:id>', '/route')
api.add_resource(RoutesList, '/routes')
api.add_resource(Ratings, '/rating/<int:user_id>', '/rating')
api.add_resource(RatingsList, '/ratings')
api.add_resource(Reports, '/report/<int:route_id>', '/report')
api.add_resource(ReportsList, '/reports')
api.add_resource(ReportsImage, '/upload')
```

Figura 27: Endpoints creats per a fer servir l'API

Igual que amb el treball anterior [1], s'ha complementat el swagger per a facilitar l'ús a futurs consumidors de l'API, on es pot veure de forma clara i amb exemples com queden els endpoints i com funcionen (figura 28). El swagger permet reproduir totes les peticions a l'API

existents per així poder-se fer una idea de com funciona i com implementar els serveis per a consumir-la:

The image shows a Swagger API interface with four main sections: Accounts, Routes, Ratings, and Reports. Each section contains a list of endpoints with their respective HTTP methods and descriptions.

Method	Endpoint	Description
Accounts Options about accounts		
GET	/accounts	Returns accounts list
GET	/account/{email}	Return account by email
PUT	/account/{email}	Update account
POST	/account	Create new account
DELETE	/account	Delete account
Routes Access to routes information		
GET	/routes	Returns routes list
GET	/route/{id}	Returns route by id
Ratings Options about ratings		
GET	/ratings	Returns ratings list
POST	/rating	Create new rating
GET	/rating/{user_id}	Returns all the ratings made by an user
Reports Options about reports		
GET	/reports	Returns reports list
POST	/report	Create new report
GET	/report/{route_id}	Returns all the reports of a route
GET	/upload	Returns all the reports of a route

Figura 28: Swagger creat a <http://everywalkyoutake.azurewebsites.net/swagger/>

7. Resultats

7.1. Navegació

Aquí es mostra el resultat final de les noves funcionalitats implementades a la interfície d'usuari. S'assumeix que es coneixen com arribar fins a iniciar sessió i a mostrar rutes i ens centrem a mostrar les noves funcionalitats.

- Sensació tèrmica:

Primerament, i com ja s'ha mencionat a les històries d'usuari ([3.2 User Stories](#)), es va afegir un nou paràmetre a l'hora de decidir que és una ruta saludable, i aquest és la sensació tèrmica. A continuació podem veure els diferents casos que existeixen:

● Cas molt de fred o molta calor:

Aquest cas contempla els intervals de temperatura de $< 10\text{ }^{\circ}\text{C}$ (figura 29) o més de $30\text{ }^{\circ}\text{C}$ de sensació tèrmica (figura 30). En aquest cas, en ser temperatura extrema, es mostren les corresponents advertències.

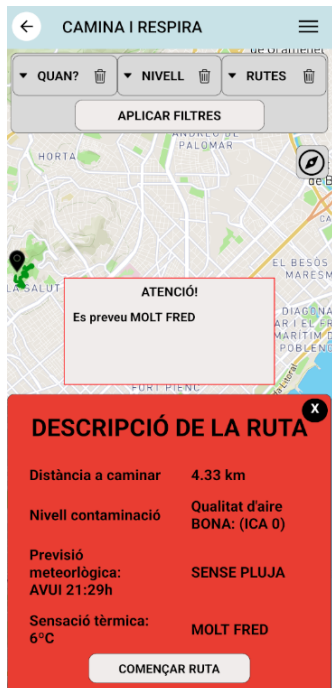


Figura 29: Sensació tèrmica amb molt fred

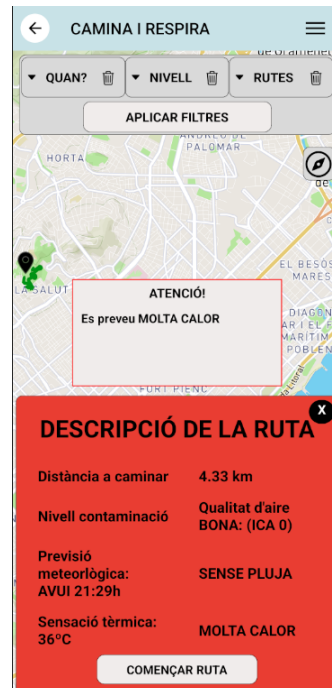


Figura 30: Sensació tèrmica amb molta calor

- **Cas fred o calor:**

Aquest cas contempla els intervals de fred d'entre 10 i 18 °C (figura 31) i de calor de 24 a 30 °C (figura 32).

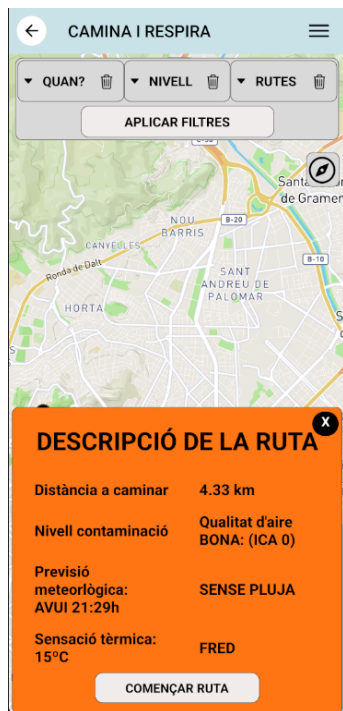


Figura 31: Sensació tèrmica amb fred

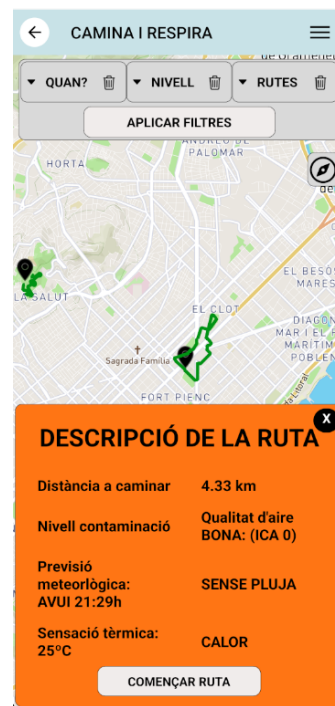


Figura 32: Sensació tèrmica amb calor

- **Cas confort tèrmic:**

Aquest cas contempla l'interval de temperatura d'entre 18 a 23 °C (figura 33).

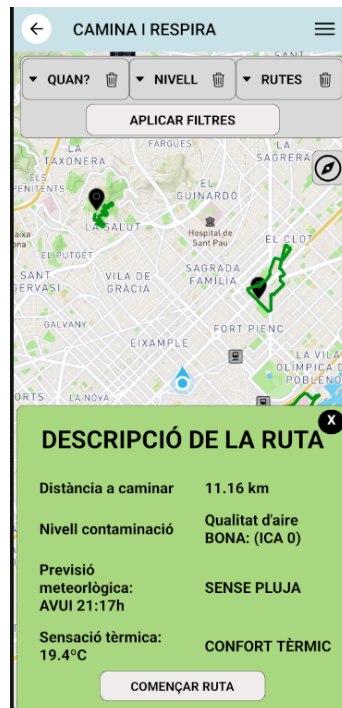


Figura 33: Sensació tèrmica ideal

- **Puntuar:**

També s’ha creat l’opció de puntuar una ruta que s’hagi completat. En aquest cas, primerament hem d’estar amb la sessió iniciada i haver començat i acabat una ruta. Un cop acabada, ens surt un modal on podem seleccionar d’1 a 5 estrelles (figura 34) i afegir un comentari (figura 35).

També disposem de l’historial de puntuacions realitzades per l’usuari, al qual podem accedir des del menú lateral (figura 36) clicant l’opció “*HISTORIAL PUNTUACIONS*” i ens mostra una llista (figura 37) de totes les puntuacions donades amb els comentaris respectius.

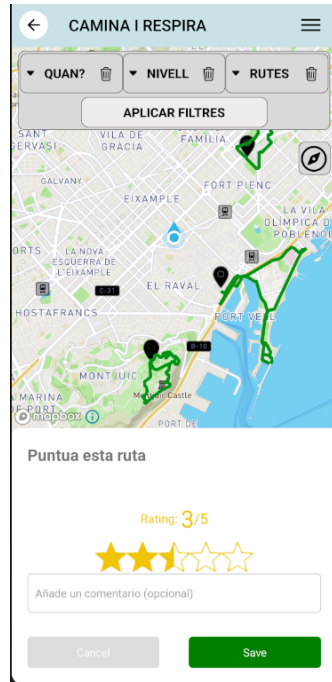


Figura 34: Cas puntuació sense comentari

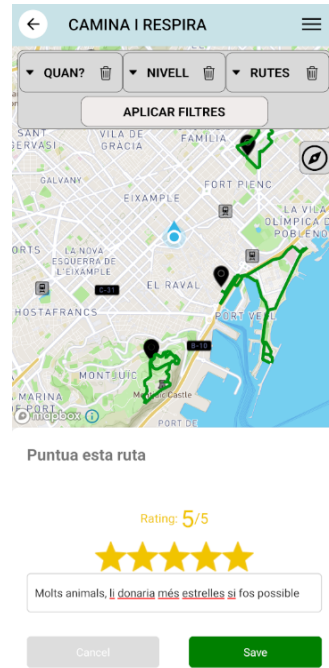


Figura 35: Cas puntuació amb comentari

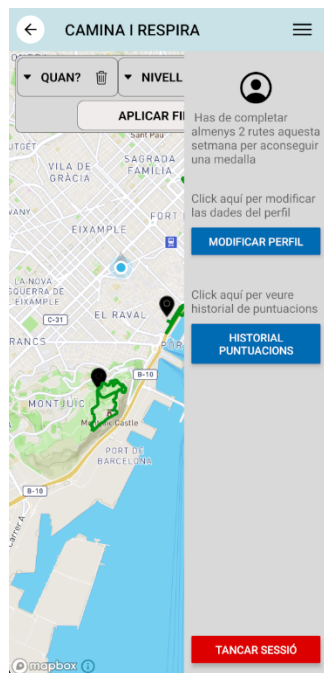


Figura 36: Menú lateral

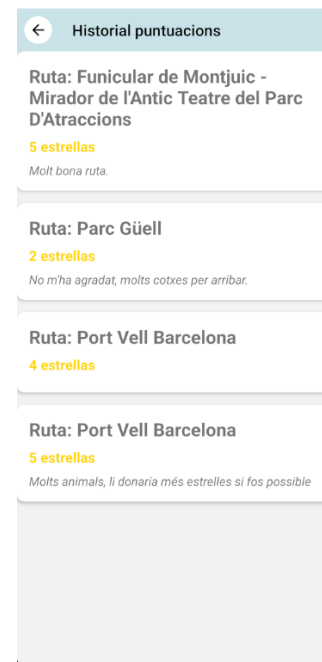


Figura 37: Historial de puntuacions

- **Reptes:**

Des del menú lateral també podem veure el nombre de rutes que ha completat l'usuari d'una manera intuïtiva gràcies als reptes. Els reptes estan dividits en 4 categories: **Cap medalla** (figura 38), **medalla de bronze** (figura 39), **medalla de plata** (figura 40) i **medalla d'or** (figura 41).

L'usuari no disposa de cap medalla (repte completat) en cas d'haver completat menys de 2 rutes durant la setmana, de bronze en cas d'haver completat 2 o 3 rutes, de plata si ha completat 4 o 5 i d'or si n'ha completat 6 o més.

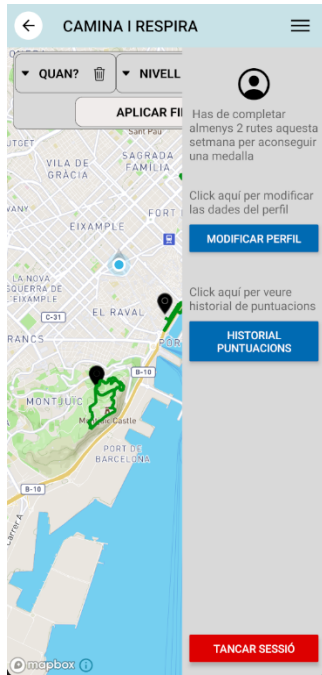


Figura 38: Cas amb cap medalla

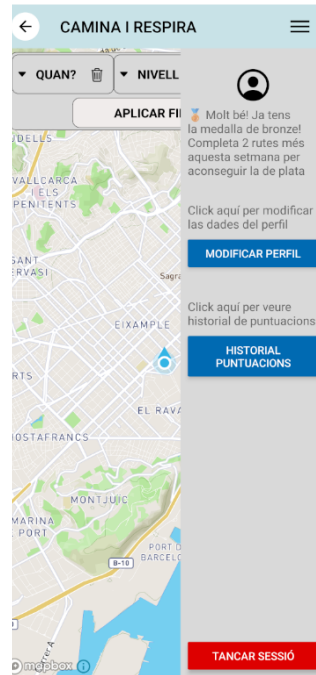


Figura 39: Cas amb medalla de bronze

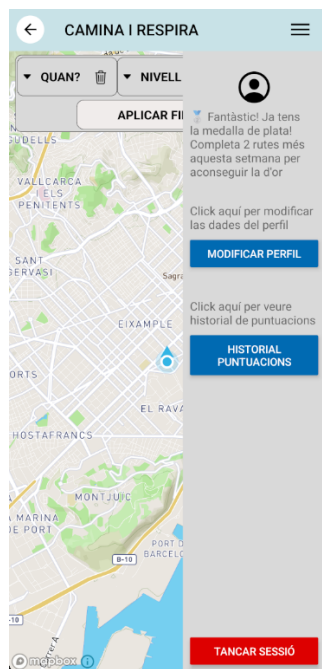


Figura 40: Cas amb medalla de plata

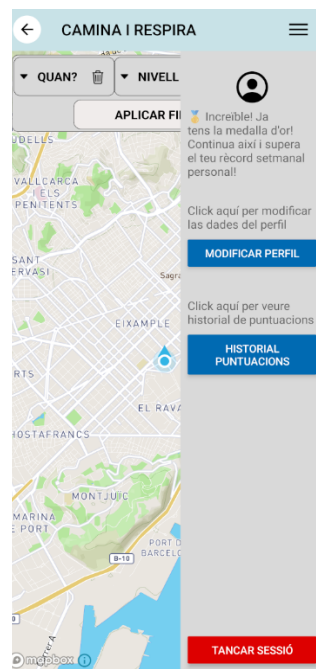


Figura 41: Cas amb medalla d'or

- **Retroacció:**

Per tal d'accedir al botó que permet l'usuari donar retroacció (figura 42), s'ha d'haver iniciat una ruta. Un cop es clica el botó “donar retroacció”, apareix un modal amb diferents opcions (figura 43). Entre elles podem veure que podem seleccionar un tipus de retroacció preestablert com ara “Ruta en mal estado” (figura 44) o afegir un tipus personalitzat a la situació actual (figura 45).

Un cop triat el tipus de retroacció, podem escollir si afegir un comentari (figura 45) o una imatge, on en aquest cas ens dona l'opció d'escollir una imatge de la galeria o de fer servir la càmera per a fer una fotografia al moment (figura 46). Una vegada s'ha seleccionat o pres una imatge, la podem visualitzar abans d'enviar la retroacció (figura 47).

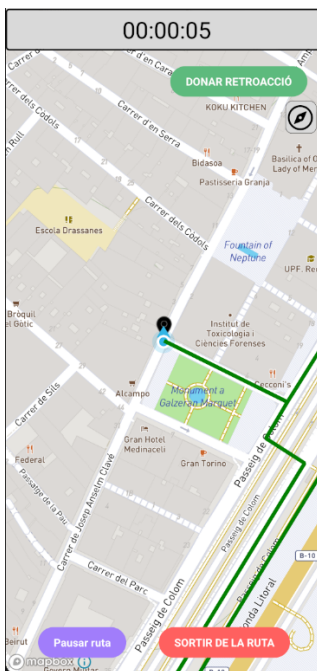


Figura 42: Interfície ruta iniciada

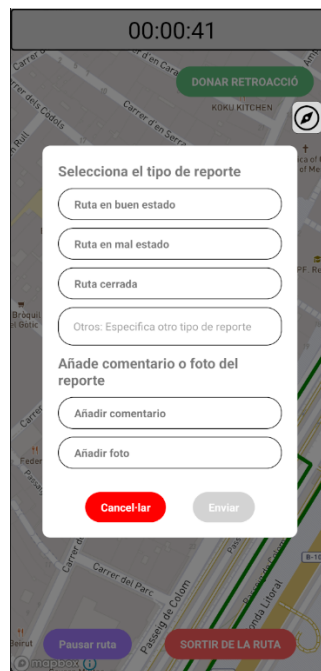


Figura 43: Modal de retroacció buit



Figura 44: Opció “ruta en mal estado” i comentari

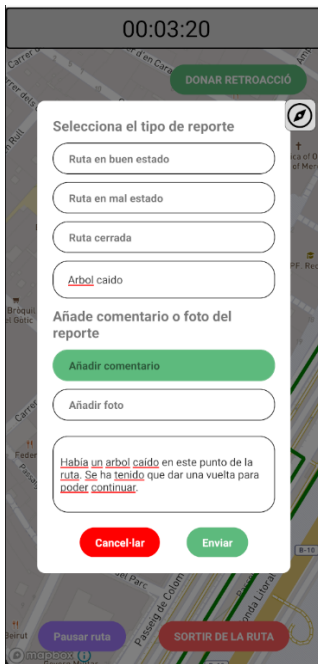


Figura 45: Opció personalitzada i comentari

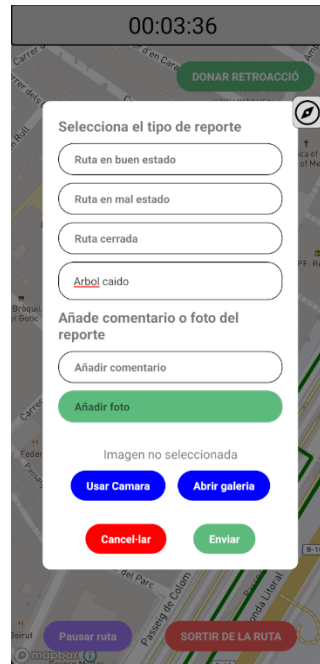


Figura 46: Opció imatge seleccionada

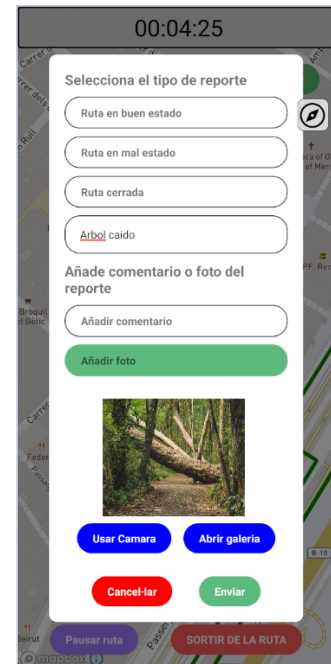


Figura 47: Després d'haver realitzat una fotografia

- **Avis ciència ciutadana:**

A l'hora d'iniciar una nova ruta, apareix sempre un avis informant que es pot participar en el mòdul d'avaluació de l'estat de la ruta amb retroacció (figura 48).

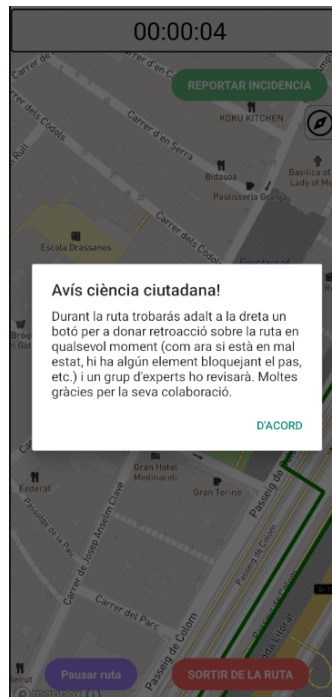


Figura 48: Avis ciència ciutadana

- **Guardar ruta:**

Per tal de tenir l'opció de guardar una ruta, cal tenir la sessió iniciada i fer servir el filtre de rutes amb l'opció "Comença aquí" (Figura 49). Un cop s'acaba la ruta, si l'usuari en qüestió no ha guardat ja 3 rutes personalitzades, li apareix un modal demanant si vol guardar la ruta que acaba de realitzar (Figura 50)

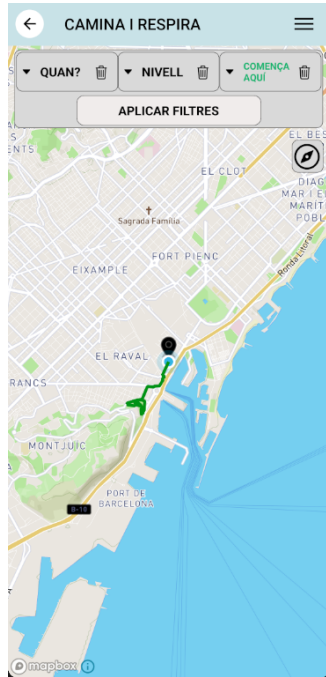


Figura 49: Filtre "Comença aquí" seleccionat

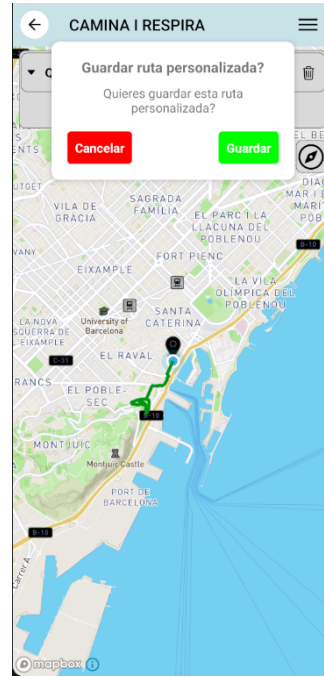


Figura 50: Modal per guardar ruta personalitzada

- **Avís ciència ciutadana final de ruta:**

En acabar una ruta, es demana a l'usuari si aquesta estava en bon estat en cas que no hagi publicat cap retroacció durant el transcurs d'aquesta, on té les opcions de contestar "sí", "no" o no contestar.

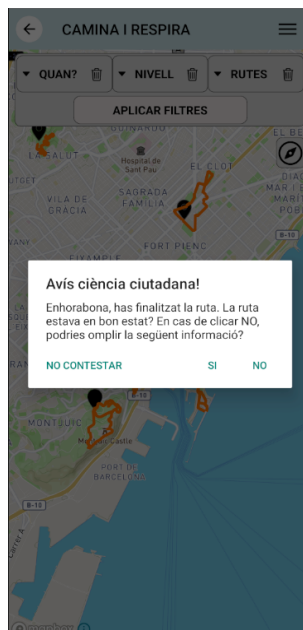


Figura 51: Avís ciència ciutadana final de ruta

7.2. Dades recollides i anàlisi de les dades

Les dades recollides durant l'ús dels mòduls d'avaluació de l'estat de la ruta amb retroacció i de l'experiència de la ruta amb puntuació han d'ajudar a crear un atlas de rutes saludables a Barcelona i a millorar les recomanacions d'aquestes rutes. Per fer-ho caldrà respondre a una sèrie de preguntes: com es veuen aquestes dades dins la base de dades? Com les veuran els científics que les analitzin? Prèviament, a la secció dels models de l'apartat [6.2.1 Estructura del codi](#) ja hem tingut una introducció a l'estructura d'aquestes dades, però ara veurem com queden a la base de dades i com els científics poden fer servir aquesta informació.

A continuació, a la figura 52 podem veure un exemple de les dades recollides a l'hora de fer servir el mòdul d'avaluació de l'estat de la ruta amb retroacció.

id	route_id	report_type	comment	coordenadas	image_url	date
1	5	good	La ruta està en bon estat	[2.16857,41.3874]		2023-05-25 09:35:13
2	5	Arbol caido		[2.16857,41.3874]	https://retroaccionsfotos.blob.c...	2023-05-27 21:55:07
3	4	closed	Hi ha una barrera que no permet el pas els divendres al vespre	[2.17874,41.37862]		2023-06-02 22:37:47
4	4	good	Bona ruta avui	[2.17874,41.37862]		2023-06-03 22:38:39

Figura 52: Dades recollides de retroaccions a la taula de retroaccions

Un cop tenim aquestes dades generades pels usuaris fent servir el mòdul d'avaluació de l'estat de la ruta amb retroacció, el nostre equip d'investigadors les podrà analitzar. En aquest cas, en podem analitzar els 4 exemples que trobem a la figura 52:

Principalment, podem veure que hi ha dues rutes puntuades, la 4 i la 5. En el cas de la ruta amb id 5, hi ha dos tipus de retroaccions, una amb el tipus "Good" i l'altre amb el tipus "Altres", concretament l'usuari que l'ha realitzat ha indicat que hi ha un arbre caigut, aportant una imatge. Si ens fixem, es pot observar com les dues retroaccions han estat fetes al mateix punt de la ruta, per tant, si observem la data quan es va fer, podem treure algunes suposicions, com ara que possiblement durant el dia 25 de maig a les 9 del matí i el 27 de maig a les 9 del vespre un arbre ha caigut a la via, així impeding el pas de la gent per la ruta i ens hauríem de plantejar que fer al respecte, però caldria mirar si tenim altres registres de la ruta 5, al voltant d'aquestes dates i coordenades per a poder treure conclusions en ferm i no fos un error de l'usuari que ha dut a terme la retroacció.

El cas de la ruta amb id 4 és el contrari. Primerament, tenim un registre que ens indica que la ruta està tancada, amb un comentari que ens dona més informació. En aquest cas ens indica que hi ha una barrera que no permet el pas els divendres al vespre, que és quan es va portar a cap la retroacció, en canvi, l'endemà al vespre tenim un altre registre que ens indica que la ruta està oberta, possiblement es deu al fet que és dissabte i no divendres tal com ens indicava el comentari de l'anterior retroacció, però igualment, abans de realitzar cap acció de pes, caldria consultar la base de dades de nou i veure si tenim més informació relacionada amb la ruta.

Per a treure conclusions amb més pes, podem agafar les dades de la taula de les retroaccions (omplerta amb la informació obtinguda a partir del mòdul d'avaluació de l'experiència de la ruta amb puntuació) i posar-les en comú amb la informació de la taula de les puntuacions (figura 53), on així en podrem treure més informació.

id	user_id	route_id	stars	comment	date
1	marc@reviews.co	2	5	Molt bona ruta.	2023-05-25 10:15:49
2	marc@reviews.co	3	2	No m'ha agradat, molts cotxes per arribar.	2023-05-25 10:15:51
9	joan@miguel.com	4	4	Ruta adequada i amb bona vegetació	2023-06-04 22:39:06
10	carles@robet.com	4	1	Ruta tancada	2023-06-02 21:13:23
11	marc@reviews.co	4	1	Ruta tancada	2023-06-09 22:34:52
3	marc@reviews.co	5	4		2023-05-25 10:15:54
8	marc@reviews.co	5	5	Molts animals, li donaria més estrelles si fos possible	2023-06-03 21:23:54

Figura 53: Dades recollides de puntuacions a la taula puntuacions

En aquest cas, ens hem de fixar en el número de la ruta, les estrelles, la data i finalment el comentari, que al ser opcional pot ser que no n'hi hagi.

Si ens fixem en la ruta amb id 4, on havíem vist que teníem dos registres, on un era fet un divendres i ens deia que la ruta estava tancada els divendres al vespre, i l'altre fet l'endemà on deia que la ruta estava correcta, veiem que tenim 3 registres: dos d'una estrella i un de quatre estrelles. En el cas d'una estrella, si mirem les dates podem veure com resulta que s'han fet dos divendres diferents del mes de juny, i ens indiquen que la ruta es troba tancada, i en el cas de les quatre estrelles, s'ha fet al vespre d'un diumenge i ens indica que li ha agradat la ruta i la vegetació d'aquesta.

Així doncs, en posar en comú les dues taules, ja podem estar més segurs que sembla que la ruta amb id 4 els divendres al vespre es troba tancada i, per tant, no l'hauríem de recomanar els divendres al vespre.

7.3. Every Walk You Take

El producte realitzat durant aquest TFG es validarà dins del Projecte “Every Walk You Take” que durarà fins a desembre del 2023. Aquest projecte ha estat finançat dins del Programa IMPETUS4CS [2] que ha estat promogut per la Unió Europea Horizon WIDERA 2021-ERA-01 Research and Innovation Programme (Grant Agreement No 101058677) i el lidera l'equip de la Dra. Maria Grau.

La validació de l'aplicació es farà en una comunitat de Stakeholders del barri de Trinitat Vella de Barcelona que inclourà: membres de l'equip directiu del Centre de Formació d'Adults del barri, investigadores del projecte, informàtics desenvolupadors de l'app, agents comunitaris i habitants del barri que actuaran com a científics ciutadans. A més, als mesos de setembre i octubre de 2023 es farà un estudi pilot durant el qual els científics ciutadans faran servir l'aplicació i recolliran informació sobre barreres i facilitadors per dur a terme un estil de vida saludable.

8. Conclusions

Un cop finalitzat el treball, considero que els objectius d'aquest projecte s'han assolit segons la llista mencionada a la introducció.

S'ha modificat l'aplicació de partida per a canviar el codi de colors i afegir la sensació tèrmica a la informació de les rutes. S'han definit reptes que ajuden a motivar a l'usuari a recórrer més rutes i així indirectament a augmentar el nombre d'interaccions amb els mòduls de ciència ciutadana.

S'ha fet que una ruta acabi automàticament i malgrat que no era als objectius, també s'ha arreglat el sistema de guardar dades a la memòria cau durant un dia i no permanentment.

S'han desenvolupat els mòduls d'avaluació de l'experiència de les rutes i de l'estat de les rutes per a permetre a l'usuari col·laborar amb el recull de dades d'una manera subjectiva i objectiva. I es deixen totes les dades recollides a una base de dades, llestes per a ser analitzades i explotades pel grup de recerca responsable d'aquest estudi.

De tal manera, també s'informa l'usuari sobre la possibilitat de col·laborar amb els mòduls de ciència ciutadana i en acabar una ruta es pregunta si hi ha hagut cap inconvenient durant el transcurs d'aquesta en cas de no haver realitzat cap retroacció.

Es permet a l'usuari guardar fins a 3 rutes personalitzades que s'afegeixen a la base de dades.

Així doncs, l'aplicació compleix amb tot el que s'havia proposat inicialment i ens porta un pas més enllà dins l'estudi de recerca "Every Walk you Take", des del que poder seguir amb noves fases per afegir noves funcionalitats i arribar al seu objectiu final de millorar l'envelliment saludable de les persones.

9. Treballs futurs

Tal com s'ha anat comentant en la memòria, aquest treball forma part d'un estudi de recerca més gran i les millores de l'aplicació i la creació dels mòduls de ciència ciutadana desenvolupats fins ara corresponen a la segona fase.

A continuació es detallen diverses tasques que es poden realitzar en les següents fases del projecte fent servir la informació recollida amb l'aplicació desenvolupada en aquest TFG:

- **Dashboard per ajudar a la visualització i anàlisi de les dades:**

Per tal d'ajudar als investigadors amb l'anàlisi de les dades, seria interessant desenvolupar un dashboard per ajudar a la visualització d'aquestes, on es mostressin diferents tipus d'informacions en format de taula o de gràfiques, el màxim personalitzable possible.

Amb aquesta eina seria més senzill fer un anàlisi de les dades i, entre d'altres, respondre preguntes com: Afecta la calor i el fred més a uns barris que a uns altres? Què ens diuen les dades quan es realitzen les rutes sobre l'estat del barri/ciutat?

- **Recomanador de rutes:**

Utilitzant les dades obtingudes amb l'aplicació d'aquest treball, es podria crear un recomanador per tal de maximitzar que les rutes que ens apareguin com a possibles a realitzar sempre fossin les millors en qualsevol moment.

Si s'acaba tenint moltes més rutes a Barcelona a part de les que existeixen actualment, també es podria mostrar només les 10/15 millors de manera visual i tenir un apartat que fos veure el top 10/20/50/100 millors rutes en funció dels filtres especificats per l'usuari.

També es podria crear alguna espècie de Chatbot amb recomanacions segons el temps i la contaminació, com per exemple, "Fes exercicis a casa quan hi hagi episodi d'alta contaminació".

- **Millorar algoritme per definir i guardar rutes personalitzades:**

Actualment, quan seleccionem l'opció de ruta "Comença aquí", l'aplicació busca les zones verdes més properes i ens porta des de la nostra ubicació, però això tendeix a què la ruta estigui molt estirada i quan arriba a la zona verda, directament es redirigeix a l'usuari cap al punt d'origen i acabi després de poc recorregut, donant així també una distància de ruta bastant curta.

Seria interessant poder aprofitar les rutes properes a la ubicació de l'usuari que ja existeixen per afavorir la creació de rutes que segueixin diverses zones verdes en funció de la dificultat, així doncs, no sent tan simples.

- **Afegir noves rutes a la base de dades:**

Una altra millora seria tenir noves rutes disponibles, amb unes característiques diferents de les que es van realitzar a la primera fase [1].

Es podrien proposar noves rutes utilitzant el catàleg d'arbres de Barcelona, el qual indica la posició de tots els arbres de la ciutat. D'aquesta manera es podria maximitzar la generació de rutes saludables passant pels carrers amb el nombre d'arbres més gran possible.

També seria interessant contemplar l'opció que els mateixos usuaris fossin els que poguessin crear les rutes per la ciutat al seu gust.

10. Bibliografia

- [1] Pérez Costa, A. (2023). *Definició de rutes saludables a partir de dades de geolocalització i contaminació* [Treball final de grau no publicat]. Universitat de Barcelona
- [2] Innovate in prevention. INNOVATE IN PREVENTION. (n.d.). <https://innovate-in-prevention.weebly.com/>
- [3] Socientize. *Green Paper on Citizen Science: Citizen Science for Europe Towards a better society of empowered citizens and enhanced research* (en anglès). European Commission, 22 de novembre de 2015, p. 1-52.
- [4] Curtes, M. (2023, April 11). *4 maps updates to help you explore U.S. national parks*. Google. <https://blog.google/products/maps/google-maps-update-national-parks-features/>
- [5] *Pricing calculator: Microsoft Azure*. Pricing Calculator | Microsoft Azure. (n.d.). https://azure.microsoft.com/en-us/pricing/calculator/?ef_id=_k_CjwKCAjwp6CkBhB_EiwAlQVyxdrSq1OAJJdlLxPFEXIL92_ZCB2UWTnQj9yWdsDCZZJ6UGj86zmKJBoCjTMQAvD_BwE_k_&OCID=AIDcmm68ejnsa0_SEM__k_CjwKCAjwp6CkBhB_EiwAlQVyxdrSq1OAJJdlLxPFEXIL92_ZCB2UWTnQj9yWdsDCZZJ6UGj86zmKJBoCjTMQAvD_BwE_k_&gclid=CjwKCAjwp6CkBhB_EiwAlQVyxdrSq1OAJJdlLxPFEXIL92_ZCB2UWTnQj9yWdsDCZJ6UGj86zmKJBoCjTMQAvD_BwE
- [6] *API Documentation*. Ambee. (n.d.). <https://docs.ambeedata.com/#aq-history-geospatial>
- [7] *Api De Dades Meteorològiques*. Meteocat. (n.d.). <https://apidocs.meteocat.gencat.cat/documentacio/dades-de-prediccio/>
- [8] *Mapbox Maps SDK: Android*. Mapbox. (n.d.). <https://docs.mapbox.com/android/maps/guides/>
- [9] *Core components and apis · REACT NATIVE*. React Native RSS. (2023, January 12). <https://reactnative.dev/docs/components-and-apis>
- [10] *Azure Database for MySQL: base de datos MySQL administrada: Microsoft Azure | google cloud. Cloud Computing Services*. (n.d.). <https://azure.microsoft.com/es-es/products/mysql/>
- [11] *App Service. Cloud Computing Services*. (n.d.). <https://azure.microsoft.com/es-es/products/app-service>
- [12] *Managing environments. Managing environments - conda 23.5.1.dev28 documentation*. (n.d.). <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>
- [13] *Azure blob storage. Microsoft Azure*. (n.d.). <https://azure.microsoft.com/es-es/products/storage/blobs/>

11. Annex

A continuació, es mostra la informació necessària per al traspàs del projecte als següents desenvolupadors que vulguin continuar el projecte.

Hosting:

S'ha de tenir en compte que per a fer el deploy del backend caldrà fer servir un compte nou d'Azure amb llicència d'estudiant o qualsevol altre servei de Hosting que es desitgi, ja que l'actual deixarà de funcionar un cop s'acabin els crèdits de la llicència.

API:

S'haurà de crear comptes per accedir a les API de contaminació "Ambee" per a obtenir una nova API KEY per versions gratuïtes o utilitzar qualsevol altre servei que es vulgui. Però un cop caduqui la KEY actual, les dades de contaminació deixaran de funcionar.

Per l'API del Meteocat, també s'haurà de demanar un nou accés, ja que la KEY actual té un ús de tres mesos i deixarà de funcionar arribat aquest punt.

Per Mapbox, la KEY és gratuïta i en principi sense caducitat, però es recomana crear un compte propi, per si un cas.

Totes les KEY i enllaços a canviar es poden trobar al fitxer *api/constants.js*.

Base de dades:

La base de dades, feta amb Azure també deixarà de funcionar un cop acabat el treball, per la qual cosa s'haurà de fer servir un altre compte d'Azure amb llicència d'estudiant o qualsevol altre base de dades que es prefereixi. Es poden canviar les dades a les quals apunta el Backend des del fitxer *config.py*.

Tot el codi es troba disponible al [repositori de GitHub](#) del projecte.