

Albert March Esteruelas

Desenvolupament d'una aplicació Android per el món de l'alimentació

“NUTRIAPP”

TREBALL DE FI DE GRAU

dirigit per Marc Ruiz Rodríguez

Grau d'Enginyeria en Sistemes i Serveis de Telecomunicacions



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

Resum

Cada cop, hi ha més gent que té en compte la seva salut i, per tant, està començant a cuidar-se amb una bona alimentació, mitjançant dietes equilibrades i saludables de manera regular.

Actualment comença a ser molt més habitual que la gent vagi a nutricionistes o tinguin l'interès en buscar alguna dieta saludable com a guia, per poder seguir-la. Així, doncs, el desenvolupament d'una aplicació enfocada al món de l'alimentació i la nutrició dins el món esportiu, pot resultar una eina molt important en les rutines diàries de qualsevol persona que estigui interessada en aquest món.

Aquest projecte consisteix en el disseny i la creació d'una aplicació per dispositius mòbils amb el sistema operatiu d'Android. L'objectiu principal és aportar als usuaris els mitjans necessaris perquè a través de l'aplicació puguin desenvolupar, emmagatzemar, afegir o eliminar les seves anotacions diàries referents a la nutrició. Tanmateix, els usuaris podran veure els seus progressos diaris amb el transcurs del temps i la seva progressió en versus els objectius individuals de cadascú. També tindran la possibilitat de poder agregar tots els aliments que vulguin menjar amb les seves dades nutricionals, sigui de manera manual o a través de la nostra app, que té la funcionalitat per a poder proporcionar-te els nutrients. A part, la nostra app constarà d'un apartat per a poder parlar amb un xat on tindràs a disposició la possibilitat de preguntar aquells dubtes que tens i tenir una resposta immediata. D'aquesta manera l'aplicació podrà ser incorporada a la vida diària de qualsevol persona, per obtenir un objectiu esportiu o bé per un benestar propi, per poder tenir un seguiment diari dels seus progressos amb controls detallats gràcies a tenir una eina al seu abast que podrà acompanyar-los en tots els processos que faci en el món de l'alimentació.

Resumen

Cada vez, hay más gente que tiene en cuenta su salud y por tanto está empezando a cuidarse con una buena alimentación, mediante dietas equilibradas y saludables de forma regular.

Empieza a ser muy habitual que la gente vaya a nutricionistas o tengan el interés en buscar alguna dieta saludable como guía, para poder seguirla. Así pues, el desarrollo de una aplicación enfocada al mundo de la alimentación y la nutrición en el mundo deportivo, puede resultar una herramienta muy importante en las rutinas diarias de cualquier persona que esté interesada en este mundo.

Este proyecto consiste en el diseño y creación de una aplicación para dispositivos móviles con el sistema operativo de Android. El objetivo principal es aportar a los usuarios los medios necesarios para que a través de la aplicación puedan desarrollar, almacenar, añadir o eliminar sus apuntes diarios referentes a la nutrición. Sin embargo, los usuarios podrán ver sus progresos diarios con el transcurso del tiempo y su progresión en versus los objetivos individuales de cada uno. También tendrán la posibilidad de poder agregar todos los alimentos que quieran comer con sus datos nutricionales, sea de forma manual oa través de nuestra app que tiene la funcionalidad para poder proporcionarte los nutrientes. Aparte, nuestra app constará de un apartado para poder hablar con un chat en el que tendrás a disposición la posibilidad de preguntar aquellas dudas que tienes y tener una respuesta inmediata. De esta forma la aplicación podrá ser incorporada a la vida diaria de cualquier

persona, para obtener un objetivo deportivo o bien por un bienestar propio, para poder tener un seguimiento diario de sus progresos con controles detallados gracias a tener una herramienta a su alcance que podrá acompañarlos en todos los procesos que realice en el mundo de la alimentación.

Abstract

More and more people are taking their health into account and are therefore starting to take care of themselves with good nutrition, through balanced and healthy diets on a regular basis.

It is becoming very common for people to go to nutritionists or have an interest in looking for a healthy diet as a guide, to be able to follow it. So, the development of an application focused on the world of food and nutrition in the sports world can be a very important tool in the daily routines of anyone who is interested in this world.

This project consists of the design and creation of an application for mobile devices with the Android operating system. The main objective is to provide users with the necessary means so that through the application they can develop, store, add or delete their daily annotations regarding nutrition. However, users will be able to see their daily progress over time and their progression against each individual's goals. They will also have the possibility to add all the foods they want to eat with their nutritional data, either manually or through our app that has the functionality to provide you with the nutrients. In addition, our app will have a section to be able to talk with a chat where you will have the possibility to ask those doubts you have and get an immediate answer. In this way, the application can be incorporated into anyone's daily life, to achieve a sporting goal or for their own well-being, to be able to monitor their progress daily with detailed controls thanks to having a tool at their disposal that will be able to accompany them in all the processes they do in the world of food.

Índex

1.-Introducció.....	1
1.1.-Descripció.....	1
1.2.-Motivació.....	2
1.3.-Objectius.....	2
1.4.-Planificació.....	3
2.-Justificació tecnologies emprades.....	5
2.1.-Tipus d'aplicacions mòbils.....	5
2.1.1 Aplicacions natives.....	6
2.1.2 Aplicacions web.....	7
2.1.3 Aplicacions híbrides.....	8
2.1.4 Elecció.....	9
2.2 Sistema operatiu.....	9
2.2.1 Android.....	10
2.2.1.1 Arquitectura Android.....	10
2.2.1.2 Avantatges i desavantatges.....	12
2.2.2 iOS.....	13
2.2.2.1 Arquitectura iOS.....	13
2.2.2.2 Avantatges i desavantatges.....	14
2.2.3 iOS VS Android.....	15
2.2.4 Elecció.....	16
3.-Requisits.....	16
3.1 Requisits funcionals.....	17
3.2 Requisits no funcionals.....	19
3.3 Casos d'ús.....	20
3.3.1 Diagrama dels casos d'ús.....	20
3.3.2 Especificació textual dels casos d'ús.....	21
4.-Entitats del projecte.....	32
5.-Disseny.....	33
5.1 Disseny de l'arquitectura Software.....	34
5.2.-Disseny de la interfície.....	35
A.- Entrada.....	36
B.- Login i Register.....	37
C.- Canvi de contrasenya.....	38
D.- Home.....	39
E.- Perfil.....	40
F.- Modificar el perfil.....	41
G.-Agenda i Afegir nota a l'agenda.....	42
H.- Calendari.....	43
I.- Dia seleccionat i Afegir notes al dia.....	43
J.- Càlcul IMC.....	44
K.- Registres IMC.....	45

L.-Solucions IMC.....	46
M.- Escàner Manual i API.....	46
N.-Llista menjars.....	48
O.- Ajuda.....	48
P.- Política de Privacitat.....	49
Q.- Xat.....	50
5.3.- Disseny de la tecnologies utilitzades.....	51
5.4.- Disseny de la base de dades.....	52
5.4.1.- Arquitectura del Room.....	53
5.4.1.1.- Avantatges del Room.....	53
5.4.1.2.- Implementació del Room.....	54
5.4.2.- Arquitectura del SQLite.....	54
5.4.2.1 Avantatges del SQLite.....	54
5.4.2.2.- Implementació del SQLite.....	55
6.- Implementació.....	56
6.1.- Estructura de la app.....	56
6.2.- Llibreries emprades.....	58
6.3.- Servei de escanejament manual.....	60
6.4.- Servei de escanejament API.....	61
6.5.- Seguretat als passwords.....	61
6.6.- Servidor.....	62
6.6.1.- Administrador de servidors.....	64
6.6.2.- Perfeccionament de la IA del servidor.....	66
7.- Proves d'ús de l'app.....	68
7.1.- Registre de l'usuari.....	69
7.2.- Login de l'usuari.....	69
7.3.- Canvi de contrasenya.....	70
7.4.- Home.....	70
7.5.- Càlcul del IMC.....	71
7.6.- Verificació Recordatoris.....	71
7.7.- Verificació Notes.....	72
7.8.- Escaner Manual.....	73
7.9.- Escaner API.....	74
7.10.- Verificació del RecyclerView (Tutorial).....	75
7.11.- Xat.....	76
7.12.- Comprovació finals dels botons.....	77
7.13.- Servidor.....	77
8.- Conclusions.....	78
9.- Annexes.....	80
Referències.....	82
Webgrafies:.....	82
Fotografies:.....	83
Diagrames:.....	84

1.-Introducció

En els últims anys he estat cursant els meus estudis en el grau d'Enginyeria en Sistemes i Serveis de Telecomunicacions. Tot i això, sempre he tingut una motivació extra a la part de l'esport i tota la part de l'alimentació saludable, per aquesta raó vaig posar-me com a objectiu poder portar a cap una aplicació on poder tenir tots els aliments i que et permetés recomanacions de dietes saludables.

En primer lloc, m'agradaria donar una breu descripció del meu projecte, les motivacions que m'han portat a cap a poder dissenyar i desenvolupar una aplicació com aquesta, i per finalitzar, tractar els diferents objectius principals del projecte.

1.1.-Descripció

NutriAPP és una aplicació exclusiva pel sistema operatiu d'Android. L'objectiu principal és poder tenir tota una gran llista d'aliments que es poden consumir, amb l'opció de poder utilitzar la càmera per escanejar els codis de barres dels aliments que estan als supermercats o els que tenim a casa i així poder observar les macros i els nutrients que comporta.

En primer lloc, està aplicació està destinada a qualsevol usuari que tingui ganes de tenir una alimentació saludable o aquelles que els agrada l'esport i volen tenir els seus hàbits alimentaris d'una manera més controlada i sana.

En segon lloc, els usuaris podran tenir una llarga llista d'aliments que estan disponibles als supermercats, on podran veure els seus components i la particularitat de poder afegir a una llista aquells aliments que trobem interessants, i amb la possibilitat de poder registrar les seves menjades durant el dia, gràcies a aquesta l'aplicació.

Aquells usuaris que els agrada portar un registre dels nutrients, consumits en cada un dels àpats, a part com hem comentat anteriorment, tindrà a la disposició l'opció de poder escanejar amb la càmera els diferents codis de barres dels aliments per poder veure quins nutrients aporta.

Entre altres coses, l'aplicació té com a objectiu final poder facilitar a tothom la informació nutricional dels aliments amb l'ajuda de tots els avantatges que aporten, amb la possibilitat de poder distribuir els seus àpats diàriament i poder tenir un registre, a través de les plantilles que cada usuari podrà generar per ell mateix.

Tenir en compte, que una aplicació com és NutriAPP, és una aplicació dissenyada i desenvolupada perquè el mateix usuari pugui tenir un control o seguiment de tots els seus reptes, les seves ingestes d'aliments i tenir una constància de quin patró alimentari està seguint. I sobretot tenint com a punt central que l'usuari pugui centrar-se en la seva pròpia salut i creixement personal.

1.2.-Motivació

L'objectiu principal en el moment d'escollir aquest projecte va ser dissenyar una aplicació que pogués ser suport del desenvolupament i seguiment de qualsevol alimentació incloent-hi els temes bàsics de qualitat alimentària.

Actualment, hi ha molta gent que està incorporant-se al món de l'alimentació saludable, començant a cuidar-se cada cop més i tenint més cura dels aliments que mengem.

La barrera d'entrada quan ens plantejem tenir cura de la nostra salut, és que ens adonem que la majoria de gent, tenen molt poca idea de la complexitat d'un món tan gran com és el de l'alimentació i, tot això, genera un seguit de dubtes, sobre quins consells alimentaris tenir en compte, i sobretot un dels problemes més grans és la falta d'informació creïble de tots aquests àmbits.

Aleshores, una de les motivacions més importants a l'hora de realitzar aquest projecte és poder trobar una solució a tots aquests problemes amb la creació d'un entorn (aplicació) on la persona interessada en el món de l'alimentació pugui trobar una guia. De la mateixa manera que li permeti tenir a l'abast tot el que sigui necessari per poder fer un seguiment, i un control per no trobar-se perdut i sense saber què fer.

No obstant això, l'aplicació desenvolupada té com a objectiu ajudar a tothom, ja sigui per una persona que just acaba de començar, com per una persona que ja fa temps que està immersa en aquest món i requereix unes necessitats diferents.

1.3.-Objectius

Més enllà dels objectius merament de l'aplicació, el meu primer objectiu i principal d'aquest projecte és aplicar tots els coneixements obtinguts al llarg de la carrera d'Enginyeria en Sistemes i Serveis de Telecomunicacions, concretament en la branca de telemàtica.

Durant l'últim curs he estat fent algunes petites aplicacions del sistema operatiu Android en diverses assignatures de la carrera, on una d'elles va ser la que vam realitzar el disseny i el desenvolupament d'una aplicació per poder realitzar el lloguer de patinets elèctrics.

Un dels objectius del meu projecte, és poder aprendre diferents tecnologies, llibreries, frameworks i a la vegada aprendre a utilitzar aquelles que són més emprades al sector TIC pel desenvolupament d'aplicacions mòbils pels dispositius amb sistema operatiu Android.

Per altra banda, crear un projecte, únicament per un sistema operatiu, en aquest cas Android, és un dels altres objectius a tenir en compte, ja que és el dispositiu que faig ús diàriament i puc fer tota la realització dels testos.

Finalment, realitzar un projecte d'una aplicació real i amb un objectiu concret em permet llançar l'aplicació al mercat, com per exemple "Google Play Store", podent convertir la resolució del meu problema o dels meus interessos en un producte exclusiu i que pugui ajudar a totes aquelles persones que també tenen dubtes a l'hora de l'alimentació.

1.4.-Planificació

En la creació de qualsevol projecte software és important realitzar una planificació del temps de desenvolupament i estimar el temps de dedicació, així com, la data de finalització del projecte per poder corregir possibles desviacions temporals i que aquestes no afectin la data de lliurament o acabament.

Per la planificació d'aquest treball utilitzaré "monday.com", ja que ofereix plantilles de gestió de projectes bastant fàcils d'utilitzar, així com altres funcions essencials per la gestió d'aquests, amb la possibilitat de poder fer uns esquemes per veure els passos a realitzar durant la realització d'aquest treball.

Com podem observar a continuació mostraré tota la planificació que he realitzat per distribuir en els següents mesos totes les parts del treball, des de la creació de l'app fins a la documentació:

<input type="checkbox"/>	Tarea	Responsable	Fecha INICIO ⓘ	Fecha FINAL	Estado ⓘ	Notas
<input type="checkbox"/>	Començar el projecte	+	✓ 25-mar.	28 mar.	Listo	Realització de les p...
<input type="checkbox"/>	Implementació de l'aplicació	+	🌙 28 mar.	9 may.	En curso	Creació de l'aplicaci...
<input type="checkbox"/>	Implementació de proves a l'a...	+	○ 25 abr.	24 may.	En curso	Testos a totes les p...
<input type="checkbox"/>	Documentació	+	○ 9 may.	26 may.	En curso	Fer tot l'escrit
<input type="checkbox"/>	Entrega del projecte Final	+	○ 3 jun.	7 jun.	En curso	
<input type="checkbox"/>	+ Agregar tarea					

25 mar. - 3 jun. 28 mar. - 7 jun.

Figura 1. Planificació del Projecte

Aquesta planificació està constituïda per diferents parts, on cada una té les seves subdivisions. A continuació mostraré quins objectius he planificat per a cadascuna d'ells.

- Començar el projecte: Aquí es detalla tota la planificació del desenvolupament del projecte, des de la cerca del professor (Marc) fins a la preparació del treball a la part del Git.
- Implementació de l'aplicació: Es distribueixen totes les parts a desenvolupar dins l'aplicació, des de la part de l'escàner fins als càlculs de l'IMC.
- Implementació de proves: Aquí s'expliquen les diferents proves que faré en acabar el projecte, per verificar que quan l'usuari faci ús de les seves funcions, no tingui cap problema.

- Documentació: En aquesta part, realitzaré tota la memòria de tot el que he anat fent durant la implementació de l'aplicació.
- Entrega del projecte: I finalment, l'última part és l'entrega del projecte al repositori de la universitat.

<input type="checkbox"/>	Tarea	Responsable	Fecha INICIO	Fecha FINAL	Estado	Notas
<input type="checkbox"/>	Començar el projecte 6		25 mar.	28 mar.	Listo	Realització de les part...
<input type="checkbox"/>	Subelemento		Owner	Estado 1	+	
<input type="checkbox"/>	Parlar am el tutor		Listo			
<input type="checkbox"/>	Presentar la meva idea		Listo			
<input type="checkbox"/>	Preparar Android Studio		Listo			
<input type="checkbox"/>	Penjar a un repositori el treball al GIT		Listo			
<input type="checkbox"/>	Decisió de parts funcionals		Listo			
<input type="checkbox"/>	Decisió parts no funcionals		Listo			
<input type="checkbox"/>	+ Agregar subelemento					

Figura 2. Principi del projecte

<input type="checkbox"/>	Implementació de l'aplica... 12	Responsable	Fecha INICIO	Fecha FINAL	Estado	Notas
<input type="checkbox"/>	20 mar.		9 may.	Listo	Creació de l'aplicació ...	
<input type="checkbox"/>	Subelemento		Owner	Estado 1	+	
<input type="checkbox"/>	Creació de l'aplicació		Listo			
<input type="checkbox"/>	Distribució de tots els layouts a fer		Listo			
<input type="checkbox"/>	Splash		Listo			
<input type="checkbox"/>	Login		Listo			
<input type="checkbox"/>	Register		Listo			
<input type="checkbox"/>	Tutorial		Listo			
<input type="checkbox"/>	Home		Listo			
<input type="checkbox"/>	Profile		Listo			
<input type="checkbox"/>	Part Principal		Listo			
<input type="checkbox"/>	Camara		Listo			
<input type="checkbox"/>	Explicacions per usuaris		Listo			
<input type="checkbox"/>	Registre d'aliments		Listo			
<input type="checkbox"/>	+ Agregar subelemento					

Figura 3. Implementació del projecte

<input type="checkbox"/>	Implementació de probes ... 2	Responsable	Fecha INICIO	Fecha FINAL	Estado	Notas
<input type="checkbox"/>	25 abr.		24 may.	Listo	Testos a totes les par...	
<input type="checkbox"/>	Subelemento		Owner	Estado 1	+	
<input type="checkbox"/>	Comprovació Usuaris Registrarse		Listo			
<input type="checkbox"/>	Comprovació Utilitats de la App		Listo			
<input type="checkbox"/>	+ Agregar subelemento					

Figura 4. Implementació de les probes

Documentació 2		9 may.	26 may.	En curso	Fer tot l'escrit
Subelemento	Owner	Estado 1		+	
Planificació parts		Listo			
Realització		En curso			
+ Agregar subelemento					

Figura 5. Documentació

Entrega del projecte Final 1		3 jun.	7 jun.	En curso
Subelemento	Owner	Estado 1		+
Verificacions de tot		En curso		
+ Agregar subelemento				

Figura 6. Resolució Final

2.-Justificació tecnologies emprades

En aquest apartat faig una breu anàlisi de les diferents tecnologies utilitzades per desenvolupar i dissenyar aplicacions mòbils, així com les motivacions que porten a utilitzar biblioteques, frameworks o tecnologies concretes per sobre d'altres.

No obstant això, un repàs dels diferents sistemes operatius per a dispositius mòbils i què em va motivar a triar el sistema operatiu Android.

2.1.-Tipus d'aplicacions mòbils

A continuació faig una breu explicació de les tres tipologies d'aplicacions mòbils existents, que són: aplicacions natives, aplicacions web i les aplicacions híbrides i finalment desenvoluparé els criteris que m'han dut a escollir l'aplicació que he fet servir en el disseny i desenvolupament del meu projecte.

2.1.1 Aplicacions natives

Les aplicacions natives són aplicacions de mòbils desenvolupades i especialitzades per a una plataforma o sistema operatiu específic, com iOS, Android, etc. Les aplicacions estan escrites en llenguatges de programació compatibles amb el sistema operatiu, això ens permet als desenvolupadors accedir-hi i així poder obtenir una comprensió més directa de les capacitats i característiques d'un dispositiu o sistema operatiu.

Les aplicacions natives també tenen SDK (kits de desenvolupament de programari) específics de la plataforma, incloses les API de desenvolupadors. Això significa que els desenvolupadors podem crear aplicacions que estiguin totalment integrades amb el sistema operatiu i altres aplicacions del dispositiu.

Tanmateix, el desenvolupament d'aplicacions natives pot ser més complex i requerir més temps i recursos de desenvolupament que altres formes de desenvolupament d'aplicacions, com ara aplicacions híbrides o aplicacions web progressives.

Moltes dels avantatges que aporten les aplicacions natives són:

- Millor rendiment i velocitat.
- Millor disseny d'interfície d'usuari.
- Accés a les funcions del dispositiu.
- Seguretat.

Malgrat els avantatges que presenten, les aplicacions natives també tenen alguns desavantatges que s'han de tenir en compte, com poden ser:

- Costos de desenvolupament.
- Dificultats de compatibilitat entre plataformes.
- Actualitzacions més complexes.
- Dependència de les botigues d'aplicacions.

Aquí podem observar un exemple d'apps i llenguatges que són natiu

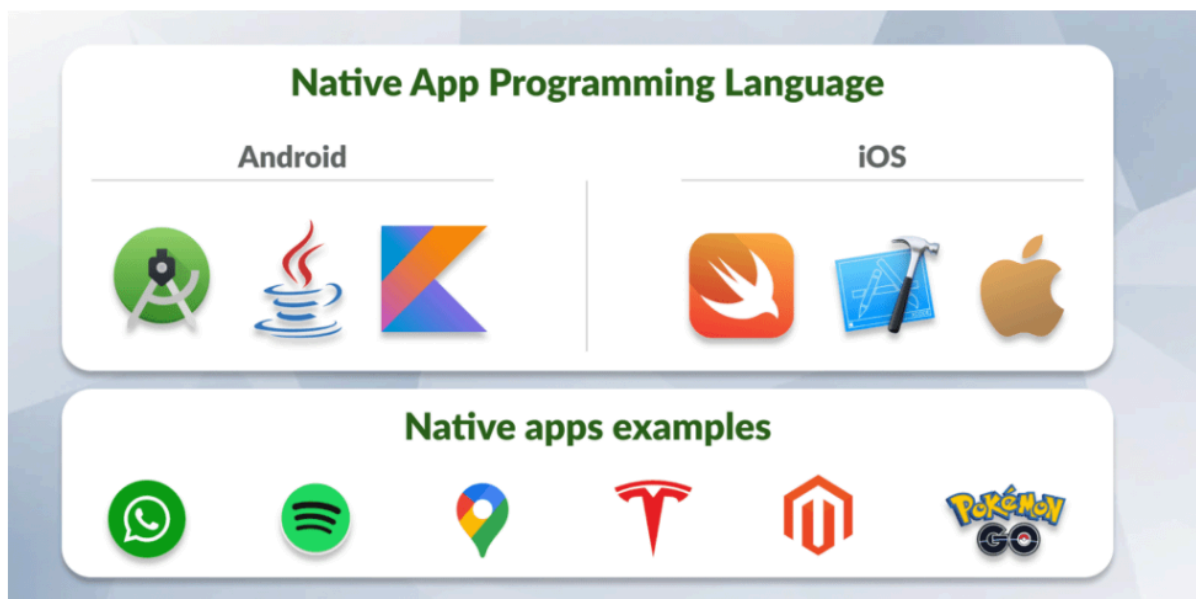


Figura 7. Aplicacions Natives

2.1.2 Aplicacions web

Les aplicacions web es poden programar independentment del sistema operatiu en el qual s'utilitza l'aplicació, cosa que fa diferència de les aplicacions natives.

Són aplicacions basades en web que no requereixen que l'usuari descarregui o instal·li res al seu dispositiu, aquestes són accessibles des de qualsevol lloc on l'usuari pugui connectar-se a Internet i no estan limitades, per la plataforma o el sistema operatiu que utilitza l'usuari.

A diferència de les aplicacions natives, que estan escrites en llenguatges de programació nativa per a una plataforma específica, les aplicacions web estan escrites en llenguatges de programació web com ara HTML, CSS i JavaScript.

El que indica que les aplicacions web, són més fàcils i menys costoses de dissenyar i desenvolupar perquè no es necessita desenvolupadors que coneguin la complexitat del sistema operatiu al qual van destinades.

Els avantatges que podem trobar en una aplicació web:

- Accessibilitat.
- Compatibilitat..
- Actualitzacions fàcils
- Desenvolupament més fàcil i menys costos.

Malgrat els avantatges que presenten, les aplicacions web també tenen alguns desavantatges que s'han de tenir en compte, com poden ser:

- Dependent de la connexió a Internet.
- Limitacions de rendiment.
- Menys funcionalitat en comparació amb les aplicacions natives.
- Problemes de seguretat.
- Dependència de tercers.

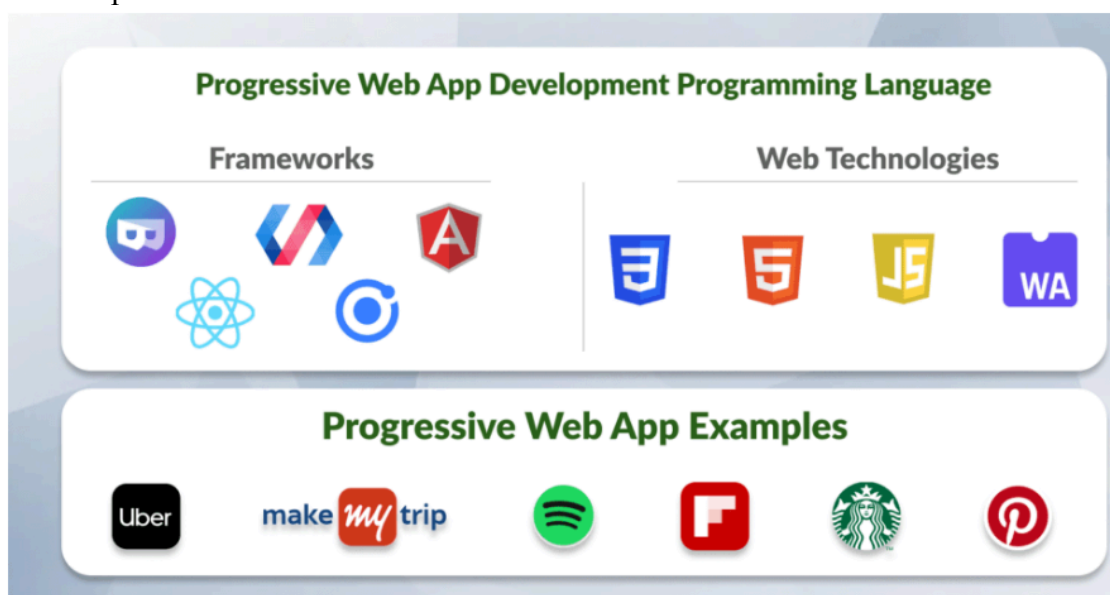


Figura 8. Aplicacions web

2.1.3 Aplicacions híbrides

Una aplicació híbrida és una combinació d'una aplicació web i una aplicació nativa, que s'executa en una plataforma o entorn nativa, però que està escrita utilitzant tecnologies web, com ara HTML, CSS i JavaScript.

Les aplicacions híbrides permeten als desenvolupadors escriure codi una vegada i després diferenciar/repartir l'aplicació per a múltiples plataformes, com ara incloent Android i iOS.

Algunes de les avantatges que presenten les aplicacions híbrides són:

- Desenvolupament més ràpid.
- Costos reduïts.
- Facilitat d'actualització.
- Funcionalitat multiplataforma.

Malgrat els avantatges que presenten algunes desavantatges que s'han de tenir en compte, com poden ser:

- Rendiment més lent.
- Dependència del marc de treball.
- Limitacions de personalització.
- La gran majoria d'aplicacions híbrides tenen les mateixes limitacions que les aplicacions desenvolupades amb tecnologies web

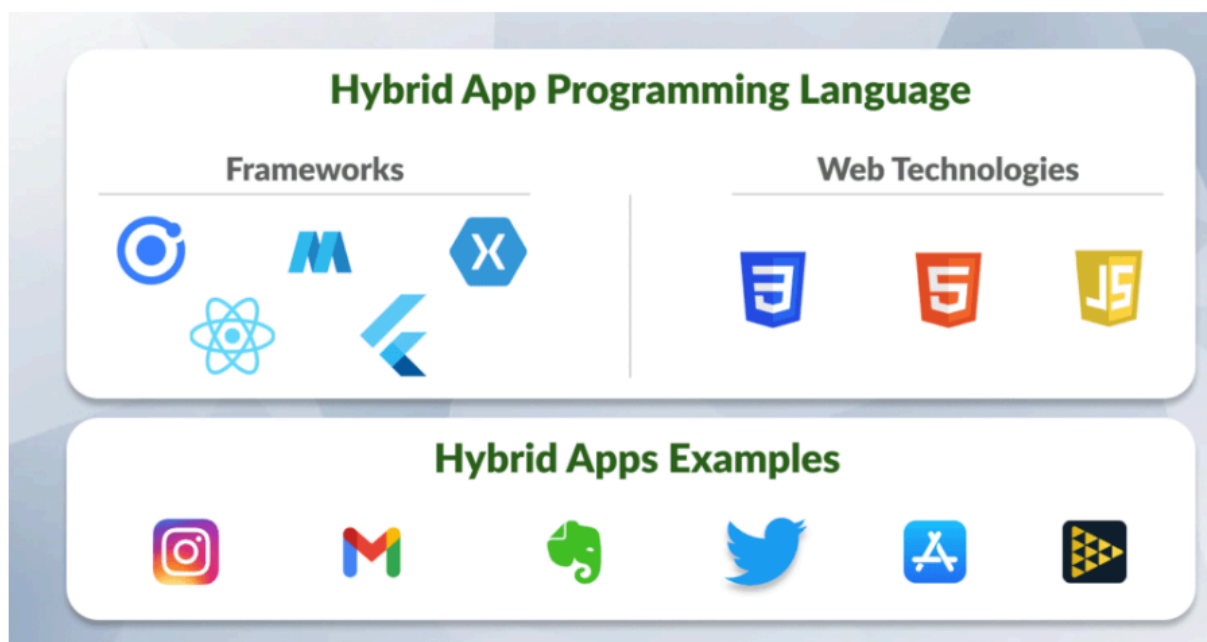


Figura 9. Aplicacions híbrides

2.1.4 Elecció

Abans de decidir-me a fer aquesta aplicació nativa per a Android, vaig estar fent un seguiment de tots els avantatges i desavantatges que aporten les diferents aplicacions, entre elles podia escollir entre aplicacions natives, híbrides o web.

Tot i que, el meu projecte podria realitzar-se amb qualsevol de les tres aplicacions, he decidit finalment quedar-me en la nativa per una qüestió de preferència personal, ja que la meva curiositat i interès personal es poder desenvolupar l'app per poder donar a tots els usuaris una gran experiència en el món de l'alimentació, gràcies als avantatges que comporten les aplicacions natives, com hem comentat anteriorment.

D'altra banda, el fet de desenvolupar una aplicació nativa m'ajuda molt, ja que em dona molts accessos en disposar d'un ordinador i mòbil que tenen aquests sistemes, i puc aprofundir més amb el seu disseny i desenvolupament.

La decisió de crear una aplicació nativa reflecteix la meua passió per aquest entorn de desenvolupament i el meu desig de crear una solució que pugui oferir el màxim valor als usuaris.



Figura 10. Aplicacions

2.2 Sistema operatiu

El sistema operatiu és un conjunt de programes que permeten la gestió i el control dels recursos d'un ordinador. Aquests recursos estan constituïts pel processador, la memòria, els dispositius d'entrada/sortida, el disc dur, la xarxa, etc. El sistema operatiu és l'encarregat de fer d'intermediari entre les aplicacions dels usuaris i el maquinari de l'ordinador.

Els sistemes operatius s'encarreguen de fer les següents tasques:

- Coordinar el hardware
- Planificació dels processos
- Gestió del sistema de fitxers
- Gestió d'entrada/sortida
- Gestió de memòria
- Interfície usuari

A mesura d'això hi ha sistemes operatius que ja estan creats exclusivament per a mòbils que tenen les capacitats per optimitzar els seus recursos. Es van crear exclusivament per als diferents dispositius mòbils, ja que tenen unes característiques de hardware més limitades comparades amb els ordinadors o altres dispositius. Actualment, el sistema operatiu més emprat per a mòbils és l'Android.

En aquesta secció es farà una descripció dels sistemes operatius més utilitzats actualment, on són l'Android i el iOS, i les motivacions per què he elegit un sistema operatiu i no l'altre.

2.2.1 Android

Android és un sistema operatiu que ha estat desenvolupat per la companyia de Google, i dissenyat per a utilitzar en dispositius mòbils. El sistema d'Android és de codi obert, ja que tots els desenvolupadors d'aplicacions poden modificar i personalitzar el codi font segons les seves necessitats.

El sistema operatiu està centrat a proporcionar una experiència als usuaris de primera classe, amb funcions com ara una personalització flexible, una interacció intuïtiva o com pot ser per tenir una gestió de la vida de la bateria intel·ligent.

Android té una gran diversitat de dispositius al mercat i una gran quantitat d'apps disponibles a través de la seva botiga oficial que tenen, anomenada Google Play Store, no obstant això, la fragmentació dels sistemes operatius pot arribar a ser un gran repte per als desenvolupadors d'aplicacions.

2.2.1.1 Arquitectura Android

Segons la documentació oficial d'Android, la seva estructura està basada en una arquitectura de capes, amb diferents components que treballen junts per així poder proporcionar una experiència a l'usuari coherent.

Una de les característiques més importants de l'Arquitectura d'Android és la seva modularitat, això significa que els diferents components del sistema operatiu poden ser actualitzats de forma independent, sense afectar a la resta del sistema.

Això permet que els fabricants de dispositius mòbils puguin personalitzar les seves implementacions d'Android i proporcionar funcions úniques als seus dispositius.

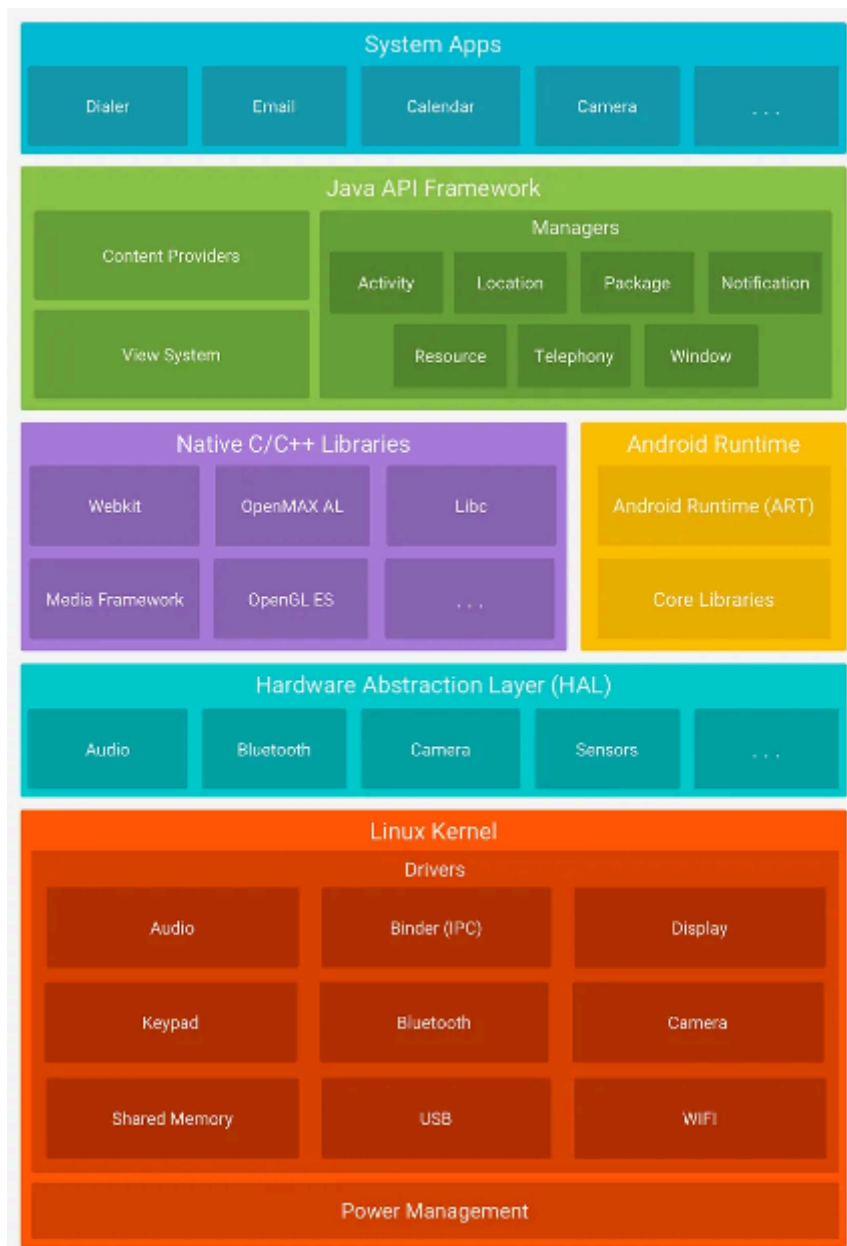


Figura 11. Capes de sistema Android

Les diferents capes que podem observar en la fotografia anterior són amb les que estan estructurades el sistema d'Android, que són les següents:

- **Aplicació:** Aquest és la primera capa, la que està localitzada a la part superior de l'arquitectura, i són les encarregades de mostrar a la pantalla dels dispositius les aplicacions corresponents.
- **Framework:** Aquesta és la següent capa després de la d'aplicacions, on està formada per un conjunt de components i serveis que les aplicacions poden utilitzar per accedir a les funcions del sistema. Aquesta capa inclou una gran

de gestos, com són, el gestor d'activitats, el gestor del cicle de vida de les activitats d'una aplicació, el gestor de serveis (encarregada de la gestió en segon pla de les aplicacions)

- **Biblioteques:** Són les encarregades de proporcionar les llibreries on estan totes les funcions que proporcionen, com poden ser els gràfics, depenent si és 2D o 3D, les bases de dades (per emmagatzemar informació), la comunicació entre diferents xarxes...
- **Runtime:** Aquesta capa inclou l'entorn executiu d'aplicacions en el sistema d'Android, que és la responsable de la seva execució.
- **Kernel de Linux:** Aquesta última capa, és l'encarregada de proporcionar l'accés als recursos de la maquinària de dispositiu, com ara seria amb la CPU, la memòria o els dispositius 'entrada/sortida. Android utilitza el nucli de Linux com a sistema operatiu subministrador.

2.2.1.2 Avantatges i desavantatges

Android, igual que després explicaré amb iOS, té una gran varietat d'avantatges i per la seva part alguns desavantatges que cal tenir en compte a l'hora de realitzar una aplicació per als seus dispositius mòbils.

Per la part d'avantatges, els sistemes Android tenen:

- **Diversitat de dispositius:** Android s'executa en una gran varietat de dispositius, des de telèfons intel·ligents fins a tauletes i fins i tot dispositius portàtils com ara rellotges intel·ligents. Això ofereix als desenvolupadors una àmplia base d'usuaris potencials.
- **Personalització:** Android permet una gran personalització tant per als usuaris com per als desenvolupadors. Els usuaris poden personalitzar el seu dispositiu amb diferents launchers, widgets i aplicacions, mentre que els desenvolupadors poden crear aplicacions molt personalitzades amb una gran varietat de funcionalitats.
- **Integració amb Google:** Android està integrat amb els serveis de Google, com ara Gmail, Google Maps, Google Drive, entre altres. Això ofereix als desenvolupadors l'oportunitat d'integrar les seves aplicacions amb aquests serveis per proporcionar una millor experiència d'usuari.
- **Software lliure i multiplataforma:** Android és un sistema operatiu de codi obert, el que significa que els desenvolupadors poden accedir al codi font del sistema operatiu i personalitzar-lo segons les seves necessitats. A més, les aplicacions d'Android es poden executar en una gran varietat de dispositius, independentment del fabricant.
- **Gestió de multitasca:** Android permet als usuaris executar diverses aplicacions alhora i canviar fàcilment entre elles. Això ofereix una millor productivitat i facilitat d'ús.

- Gran nombre d'aplicacions: L'ecosistema d'Android disposa d'una àmplia gamma d'aplicacions disponibles a Google Play Store. Això ofereix als usuaris una gran varietat d'opcions per trobar aplicacions per a les seves necessitats específiques.

Per altra part, els desavantatges, que comporta el sistema d'Android són:

- Tenir una gran varietat de versions d'Android, obliga que els desenvolupadors hagin d'estar constatant revisant les seves aplicacions i els seus funcionaments per a adaptar-los a totes aquelles versions.
- La gestió de multitasques, també pot ser considerada com a desavantatge, ja que en tenir aplicacions en segon pla, pot comportar que rallenteixi el dispositiu.

2.2.2 iOS

iOS és un sistema operatiu mòbil desenvolupat i utilitzat per Apple, pels seus desenvolupadors mòbils, com poden ser la varietat de dispositius que tenen, com són, iPhone, l'iPad, iVision...

Aquest sistema operatiu és conegut per la seva interfície d'usuaris intuïtiva, la seva seguretat, la privacitat i la seva integració amb els altres dispositius i serveis d'Apple.

2.2.2.1 Arquitectura iOS

En els dispositius d'iOS, la seva arquitectura està organitzada per una combinació de software i hardware dissenyats per Apple, per tal de poder oferir un rendiment òptim als seus dispositius.

A escala de software, està dissenyat en el Kernel de macOS, però també ha estat optimitzat per funcionar en dispositius mòbils. El Kernel és el component central del sistema operatiu que s'encarrega del control dels recursos de hardware i la comunicació dels diferents components del software entre ells mateixos.

A escala de hardware, els dispositius utilitzen processadors ARM basats en la tecnologia de RISC (Reduced Instruction Set Computing) que són dissenyats per oferir un rendiment elevat amb el consum d'energia reduïda.

L'arquitectura d'iOS es pot entendre com una arquitectura de capes, consta de diferents capes de software que interactuen entre si, per així poder dur a terme i proporcionar la funcionalitat necessària del sistema operatiu, segons es pot indicar amb la documentació de la pàgina d'Apple.

Les capes que configuren el sistema operatiu d'iOS és:

- Cocoa Touch: Aquesta és la capa superior de tot i és l'encarregada dels components d'interfície d'usuari per a les aplicacions i també inclou el frameworks que s'utilitzen per a les funcions comunes de les aplicacions.
- Media: Aquesta capa proporciona els frameworks per la gestió del contingut multimèdia, com ara el vídeo, l'àudio i els gràfics.
- Core Services: Aquesta capa proporciona funcions comunes que es poden utilitzar per a diverses aplicacions, com ara la gestió de xarxes, el control d'accés i l'emmagatzematge de dades.
- Core iOS: Aquesta última capa proporciona els components del sistema operatiu a nivell baix, com ara el Kernel

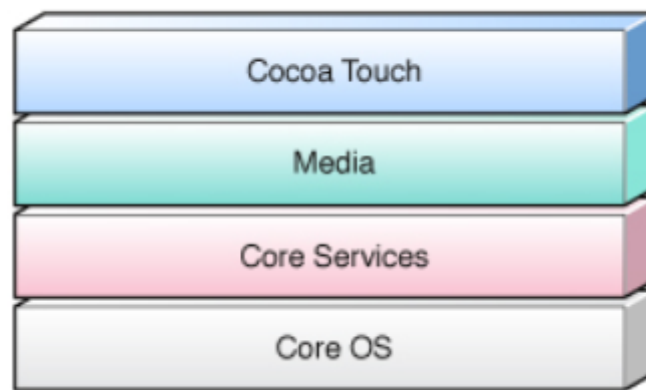


Figura 12. Capes de sistema iOS

2.2.2.2 Avantatges i desavantatges

iOS té una sèrie d'avantatges i desavantatges que cal considerar també a l'hora de realitzar un aplicació amb el seu sistema operatiu:

Els avantatges d'optar per iOS com a plataforma de desenvolupament d'aplicacions inclouen:

- Interfície d'usuari atractiva i intuïtiva: iOS ofereix una interfície d'usuari coherent i elegant que és coneguda per la seva simplicitat i facilitat d'ús. Això fa que les aplicacions d'iOS siguin atractives visualment i fàcils d'utilitzar per als usuaris.
- Mercat d'usuaris: Els dispositius iOS, com l'iPhone i l'iPad, tenen una base d'usuaris fidel i àmplia. Això ofereix als desenvolupadors una gran audiència potencial per als seus productes i un mercat estable per al llançament d'aplicacions.
- Plataforma segura: iOS és conegut per la seva seguretat i privadesa. El sistema operatiu té mecanismes integrats per protegir les dades dels usuaris i evitar la instal·lació de software maliciós.

- Bona integració amb hardware: Els dispositius iOS estan dissenyats específicament per treballar amb el software d'Apple, la qual cosa proporciona una integració perfecta entre el maquinari i el programari. Això permet als desenvolupadors crear aplicacions que aprofiten al màxim les funcionalitats dels dispositius iOS.
- Qualitat d'aplicacions: Els estàndards d'aprovació de l'App Store d'Apple són rigorosos, la qual cosa assegura que les aplicacions disponibles per a iOS mantinguin un alt nivell de qualitat i rendiment.

D'altra banda, els desavantatges d'optar per iOS com a plataforma de desenvolupament d'aplicacions inclouen:

- Preu elevat per desenvolupar aplicacions iOS: El desenvolupament d'aplicacions per a iOS pot ser més car en comparació amb altres plataformes, especialment a causa de les taxes d'inscripció al Apple Developer Program i el cost dels dispositius d'Apple per a proves.
- Únic entorn disponible per desenvolupar aplicacions amb aquest llenguatge, Xcode: El desenvolupament d'aplicacions per a iOS es fa principalment amb l'entorn de desenvolupament integrat (IDE) d'Apple, Xcode, que pot limitar les opcions per als desenvolupadors que prefereixen altres entorns de desenvolupament.
- Dependència de les eines d'Apple: Els desenvolupadors d'aplicacions per a iOS estan limitats a les eines proporcionades per Apple, la qual cosa pot limitar la flexibilitat i la innovació en el desenvolupament d'aplicacions.
- Limitacions de personalització: Les polítiques d'Apple sobre la personalització i la funcionalitat de les aplicacions poden ser més restrictives en comparació amb altres plataformes, la qual cosa pot limitar la creativitat dels desenvolupadors.
- Control exhaustiu de permisos: Apple imposa un control exhaustiu sobre els permisos de les aplicacions, el que pot limitar les funcionalitats disponibles per als desenvolupadors i els usuaris. Això pot afectar la capacitat de les aplicacions per accedir a determinades funcionalitats del dispositiu o integrar-se amb altres serveis.

2.2.3 iOS VS Android

Com són els dos sistemes operatius d'Android i iOS, tots dos tenen uns grans avantatges amb els seus punts forts i les seves parts febles, així que a partir d'aquí faré un breu resum d'aquestes diferències, les quals he trobat que són més importants a tenir en compte.

- Preus: Els dispositius d'Android solen ser més econòmics que els dispositius d'iOS, cosa que pot comportar que els consumidors busquen una opció més assequible a l'hora de realitzar els seus projectes de desenvolupament.
- Facilitat d'ús: Android ofereix la possibilitat d'una gran personalització i flexibilitat a l'hora de què l'usuari realitza els canvis que li vinguin de gust.

Per altra banda, iOS, és conegut més per tenir una interfície més intuïtiva i fàcil d'utilitzar amb icones més clares i senzilles.

- Integració de software: Android té un ecosistema més obert i té una gran varietat alhora del rendiment gràcies a la gran quantitat de versions que ofereix, en canvi, en iOS el seu hardware i software estan desenvolupats per la mateixa companyia d'Apple, cosa que permet més integració.
- Selecció d'aplicacions: En el cas d'Android, la botiga d'aplicacions té una selecció més àmplia d'aplicacions i jocs en comparació amb l'App Store d'Apple. Aquesta diversitat pot ser un avantatge per als usuaris d'Android, ja que tenen accés a una gamma més gran de productes i serveis. D'altra banda, l'App Store d'Apple es distingeix per tenir un control més gran sobre les aplicacions que s'hi ofereixen, la qual cosa pot garantir una major qualitat i seguretat. Tot i això, aquest control més estricte també significa que hi ha menys opcions disponibles per als usuaris.

En conclusió, els dos sistemes aporten una gran varietat d'avantatges i desavantatges respecte a l'altre sistema operatiu, cosa que permet una gran varietat als desenvolupadors per elegir el sistema que va més orientat al que vol desenvolupar.

2.2.4 Elecció

Primer de tot, cap dels dos sistemes operatius tractats anteriorment tenen avantatges clars respecte als seus competidors, tots tenen els seus pros i contres.

Una de les motivacions principals d'aquesta aplicació és permetre l'ús familiar de l'aplicació, per la qual cosa, em sembla més lògic el desenvolupament i disseny de l'aplicació pel sistema operatiu Android, ja que gran part de la meva família té aquest dispositiu.

Un altre motiu de pes per al desenvolupament d'aquest projecte, perquè sigui dirigit al sistema operatiu Android, és que desenvolupar una aplicació accessible per aquest SO és necessari la disposició d'un terminal Android cosa, que personalment dispo d'aquests dispositius.

A més, el sistema operatiu Android ofereix una gran quantitat de recursos i eines per als desenvolupadors, incloent-hi una extensa documentació i suport tècnic, així com una gran varietat de biblioteques i frameworks que simplifiquen el procés de desenvolupament.

3.-Requisits

A continuació exposaré els elements bàsics dels requisits funcionals i no funcionals de l'aplicació.

La realització d'aquests punts m'ajudarà a especificar amb molts més detalls les tasques a realitzar a l'hora del desenvolupament i disseny del software. I la seva anàlisi, dels casos

possibles que em permeten identificar els diferents comportaments de l'app que desenvolupo, i poder especificar més les accions que permeten que els usuaris interaccionin amb l'aplicació desenvolupada.

3.1 Requisits funcionals

A continuació realitzaré l'anàlisi dels requisits funcionals que ha de tenir integrats l'aplicació. Tots aquests objectius els vaig elegir abans de la realització del projecte, que són els objectius funcionals de l'app, els quals comporten totes les funcions presents que ha de tenir l'app desenvolupada.

A continuació farà una taula per cada una d'aquelles funcions que vaig enumerar abans de començar el projecte amb una breu explicació de cada un d'ells.

REQUISITS FUNCIONALS	EXPLICACIONS
1	Permetre que l'usuari es pugui registrar/identificar-se, si ja estàs autenticat a l'app. A través del correu i una contrasenya (que són els més comuns a l'hora d'autenticar-se a una aplicació)
2	La primera vegada que l'usuari es registri podrà posar el seu nom per a poder tenir un petit perfil d'ell
3	Si l'usuari ja té compte que pugui entrar, directament a l'aplicació
4	Si l'usuari no se'n recorda de la contrasenya, que pugui modificar-la
5	Que dins l'aplicació pugui veure les seves dades i poder-les modificar
6	Que pugui tancar la seva compte i posar una altra si vol
7	Permetre a l'usuari escollir entre les diferents funcions que té la nostra app
8	Permetre a l'usuari poder calcular el seu IMC, a través del seu pes i la seva alçada
9	Poder tindre un registre on surtin tots els càlculs que ha fet durant tot el temps que ha estat

10	Mostrar a l'usuari segons el seu IMC, unes explicacions de com està ell/a i uns comentaris d'ajuda o motivació
11	L'usuari podrà registrar en un calendari les activitats que va realitzant durant els dies o posar recordatoris
12	L'usuari podrà eliminar aquelles notes que ja no necessita
13	L'usuari podrà sempre entra a les seves notes per a la seva modificació, en cas d'algun error
14	L'usuari podrà entra a una pestanya on surten cada dia de l'any i els podrà seleccionar
15	L'usuari podrà veure les macros que ha consumit aquell dia
16	L'usuari dins de cada dia podrà afegir els nutrients consumits
17	L'usuari tindrà un botó d'ajuda per a poder veure el funcionament de l'app, per si tingués algun problema
18	L'usuari podrà escanejar els codis de barres dels aliments
19	L'usuari podrà agregar tots aquells aliments que hagi escanejat amb el seu nom i les seves macros
20	L'usuari tindrà un registre on podrà veure tots aquells aliments que té registrats
21	Si l'usuari ja té registrat aquells aliments, es mostrarà un missatge dient-li el nom de l'aliment, conforme ja el té
22	L'usuari podrà modificar tots els elements que ha afegit a la seva llista
23	També estira la possibilitat d'eliminar tots aquells aliments que ja no vulguis tenir a la llista o tinguis algun error
24	Tenir la possibilitat de veure la política de privacitat de la nostra app

25	L'usuari tindrà una IA a la seva disposició per enviar-li dubtes que tingui i li respondrà.
-----------	---

3.2 Requisits no funcionals

Els requisits no funcionals són aquells que ens especifiquen els criteris que es poden utilitzar per valorar així el funcionament del sistema desenvolupat.

Aquests requisits són els que s'organitzen a la necessitat de l'usuari, són els que defineixen les restriccions del sistema com ara la capacitat dels dispositius d'entrada/sortida i la representació de totes les dades utilitzades en la interfície de l'usuari.

REQUISITS NO FUNCIONALS	EXPLICACIONS
1	El disseny de l'aplicació és intuïtiu, fàcil de navegar per tota l'aplicació.
2	L'aplicació ha de tenir persistència de dades.
3	Algunes parts del sistema: creació de notes, visualització d'aliments. Ha d'anar en temps real.
4	Utilització de productes de Firestore (base de dades), per l'autenticació, els apunts, els aliments i l'IMC.
5	Implementació d'un hash en la contrasenya de l'usuari, per tenir una millora de seguretat
6	Connexió API a través d'una base de dades per detectar els aliments escanejats
7	Possibilitat que en un futur, pugui haver-hi la implementació d'una aplicació web.
8	Implementació servei web, per a poder connectar una IA amb la nostra App.

3.3 Casos d'ús

En aquest apartat de la documentació, parlaré i descriure els diferents casos d'ús, on a través d'un usuari, que seré jo mateix, i realitzaré totes aquelles opcions que permet l'app.

3.3.1 Diagrama dels casos d'ús

Aquí mostraré un diagrama de tots aquells casos que pot realitzar l'usuari dins l'app.

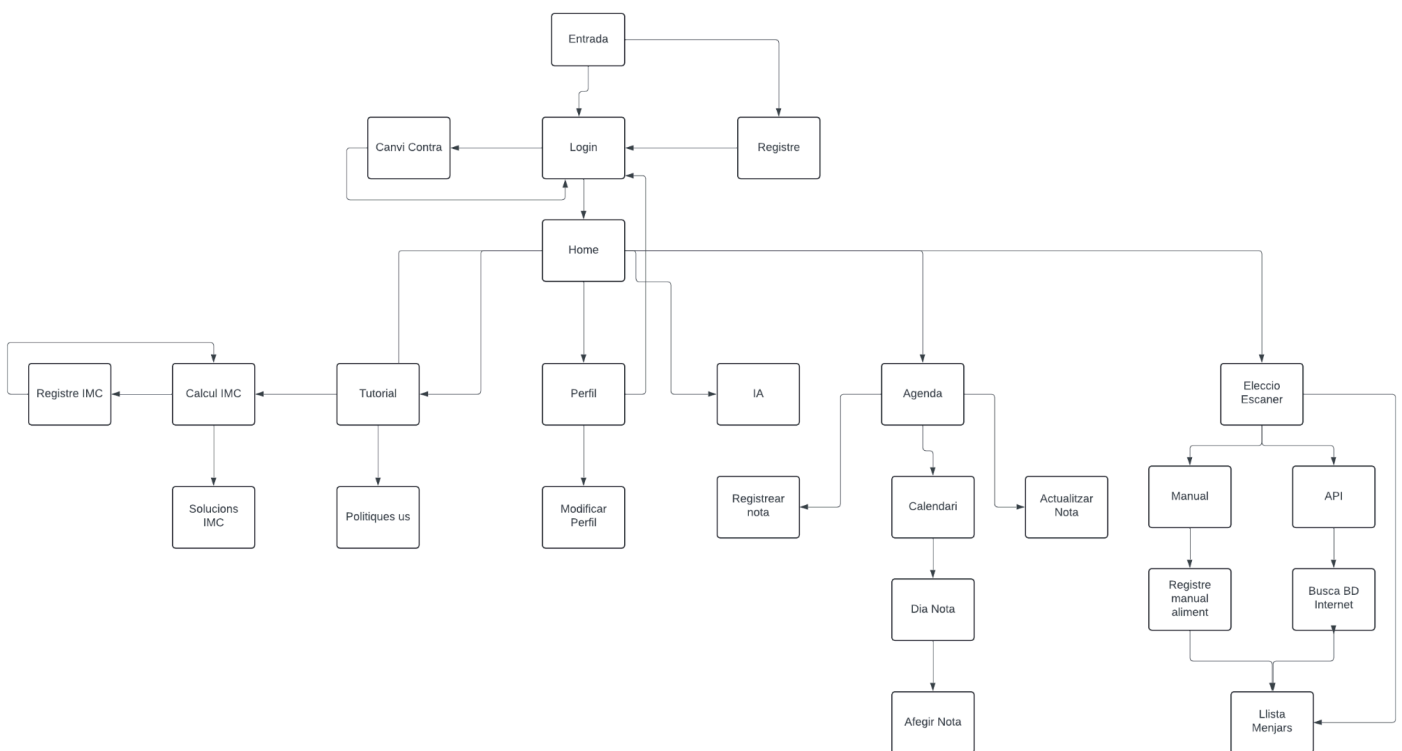


Diagrama 1.Estructura App

3.3.2 Especificació textual dels casos d'ús

Cas d'ús 01. Registre

FUNCIONALITAT	La funcionalitat és que l'usuari es pugui registrar a l'app amb les seves dades
PARAM. ENTRADA	Correu Electrònic, Nom complet, Contrasenya
PRECONDICIÓ	Ninguna
POSTCONDICIÓ	Usuari creat a l'app
ACTOR	Jo

- 1.-L'usuari es connecta al layout de registrar-se
- 2.-A continuació, l'actor ellegeix quines dades registrarà a l'app per donar-se d'alta
- 3.-L'aplicació es connecta amb la base de Dades i guarda el nou usuari
- 4.-Finalment després de fer el registre, connecta a l'usuari a la nova pàgina per entrar a l'app

Excepcions:

- 3.1.-Si l'usuari ja està registrat amb aquelles credencials, no deixa crear una altra compta i surt una notificació

Cas d'ús 02. Login

FUNCIONALITAT	La funcionalitat és loguejar a l'actor dins l'app
PARAM. ENTRADA	Correu Electrònic, Contrasenya
PRECONDICIÓ	Tenir un compte a la app
POSTCONDICIÓ	Inici del compte
ACTOR	Jo

- 1.-L'usuari entra a la pàgina d'entrada de l'app
- 2.-L'usuari si té el compte ja creat, ha d'introduir les seves credencials a l'app, per poder entrar
- 3.-Després l'app busca a la base de Dades, si existeix l'usuari introduït es dona accés a l'app, sinó no te'n deixa seguir
- 4.-Si l'usuari ha estat ben introduït entrem a la pàgina del Home

Cas d'ús 03. Canvi Contrasenya

FUNCIONALITAT	La funcionalitat és poder canviar la contrasenya si la vols canviar o per si t'has oblidat
PARAM. ENTRADA	Correu Electrònic, Contrasenya
PRECONDICIÓ	Tenir un compte a l'app
POSTCONDICIÓ	Canvi contrasenya
ACTOR	Jo

- 1.-L'usuari prem el botó de canviar contrasenya, que està situat al layout de l'inici de sessió
- 2.-Aquí primer et surt un quadre de text on verifica que existeixi l'email que n'introdueixes
- 3.-Mira si està a la base de dades aquest email, si és així t'apareixen al quadre de text per canviar la contrasenya, sinó no pots avançar de pàgina.
- 4.-Has de posar-la dos cops i ha d'estar igual, sinó no et deixa poder modificar-la
- 5.-Si compleix tots els requisits, es modifica a la base de dades la contrasenya i tornes al layout de inici de sessió

Cas d'ús 04. Pantalla Inicial

FUNCIONALITAT	La funcionalitat és poder observar totes les possibilitats que et pot oferir la nostra app
PARAM. ENTRADA	Cap
PRECONDICIÓ	Cap
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-La funcionalitat és que l'usuari pugui observar totes les opcions que ens permet poder fer la nostra app, amb els diferents botons que hi ha disponibles

Cas d'ús 05. Perfil

FUNCIONALITAT	La funcionalitat és que l'usuari pugui veure totes aquelles dades que ha introduït a l'app
PARAM. ENTRADA	Cap

PRECONDICIÓ	Cap
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-L'usuari a partir del botó, que està a la pantalla anterior del Home, pot anar a la pàgina del perfil
- 2.-L'app, càrrega de la base de dades, l'usuari que està iniciat en aquell moment
- 3.-L'usuari pot veure les dades que té a l'aplicació, excepte la contrasenya que és privada

Cas d'ús 06. Sortir Usuari

FUNCIONALITAT	La funcionalitat és que l'usuari pugui tancar el seu compte de l'aplicació
PARAM. ENTRADA	Cap
PRECONDICIÓ	Cap
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-L'usuari prem el botó situat a la pantalla del perfil, on tanca automàticament el compte i torna a la pàgina de inici de sessió.

Cas d'ús 07. Edició de l'Usuari

FUNCIONALITAT	La funcionalitat és que l'usuari pugui modificar aquelles dades que vol, de la seva compta
PARAM. ENTRADA	Cap
PRECONDICIÓ	Cap
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-L'usuari ha d'anar al botó de modificació, que està a la pàgina de perfil
- 2.-Així es dirigeix a la nova pàgina, on allí té tots els camps per modificar
- 3.-Aquí té la possibilitat de modificar les dades i contrasenya que té a l'app
- 4.-Si l'usuari decideix modificar alguna dada, llavors ha de clicar al botó de guardar

- 5.-L'app llavors busca a la base de dades, l'usuari i modifica a través de la funció que tenim les dades que estan registrades i les sobreesciu a les noves
- 6.-L'app torna a enviar a l'usuari a la pàgina de perfil, per poder observar els canvis aplicats

Cas d'ús 08. Ajuda

FUNCIONALITAT	La funcionalitat és que l'usuari pugui veure quines opcions ens permet fer l'app, d'una manera fàcil i intuïtiva
PARAM. ENTRADA	Cap
PRECONDICIÓ	Cap
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-L'usuari està a la pàgina de l'Home, i té l'opció de prémer el botó d'ajuda que està implementat al costat del perfil
- 2.-Llavors envia a l'usuari, a una pàgina on està creat un recyclerview, on mostra en cada una de les pàgines els botons que pot utilitzar l'usuari i quines funcions tenen cada un d'ells.

Cas d'ús 09. Càlcul del IMC

FUNCIONALITAT	La funcionalitat és que té aquest botó, és tenir la possibilitat de poder calcular el teu Índex de Massa Corporal, segons el teu pes i la teva altura
PARAM. ENTRADA	Pes i Altura
PRECONDICIÓ	Cap
POSTCONDICIÓ	Detalls segons el teu IMC
ACTOR	Jo

- 1.-L'usuari entra al botó de càlcul d'IMC, que està a la pàgina del Home
- 2.-Aquí dins s'han d'introduir les dades que et demana l'aplicació, que són el pes i l'altura
- 3.-A partir d'aquí l'app fas el càlcul que està implementat per l'IMC, i et mostra per pantalla el teu Índex de Massa Corporal.
- 4.-Llavors l'aplicació busca el teu usuari i afegeix en la base de dades el valor que ha sortit, perquè també el puguis veure al teu perfil i tenir un registre

Cas d'ús 10. Recomptes dels càlculs IMC

FUNCIONALITAT	La funcionalitat és que té aquest botó, és poder veure un registre continu de tots els càlculs que has fet sobre el teu IMC des del principi de l'app
PARAM. ENTRADA	Cap
PRECONDICIÓ	Haver calculat algun IMC
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-Primer l'usuari a la pantalla de càlculs ha de haver fet alguna operació per veure el seu IMC
- 2.-La base de dades busca llavors tots els càlculs que ha fet aquest usuari
- 3.-Llavors mostra una llista de tots els càlculs que ha fet anteriorment

Cas d'ús 11. Solucions del IMC

FUNCIONALITAT	La funcionalitat és que després de fer el teu càlcul, l'app et pugui donar uns comentaris i dir-te el rang de massa corporal on estàs situat
PARAM. ENTRADA	Cap
PRECONDICIÓ	Càlcul de l'IMC
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-L'usuari entra a la pàgina on li mostra una taula depenent dels rangs de l'IMC
- 2.-L'app busca l'IMC dins la base de dades de l'usuari
- 3.-Depenent del valor de l'IMC es mostra una persona dins la taula en la qual està l'usuari i uns comentaris complementaris per explicar els resultats

Cas d'ús 12. Agenda

FUNCIONALITAT	La funcionalitat és que puguis anar introduint tots aquells recordatoris que vulguis dins l'app, sobre el que vas fent
PARAM. ENTRADA	Cap

PRECONDICIÓ	Cap
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-L'usuari entra a la pàgina de Calendari, on mostra per pantalla un llistat dels recordatoris que n'has anat apuntant
- 2.-On tens la possibilitat de poder afegir o eliminar els recordatoris que vols

Cas d'ús 13. Registrar recordatoris

FUNCIONALITAT	La funcionalitat és per a poder anar afegint tots aquells recordatoris que necessites registrar per més endavant poder revisar
PARAM. ENTRADA	Títol i descripció
PRECONDICIÓ	Cap
POSTCONDICIÓ	Nota creada i guardada a la base de dades
ACTOR	Jo

- 1.-Primer l'usuari entra des de l'agenda al botó d'afegir per poder crear una nova nota
- 2.-A l'usuari li apareix una nova pàgina, on ha d'introduir un títol de la nota i després afegir els comentaris que en vulgui
- 3.-Posteriorment la nota es guarda en una base de dades diferent on va registrant totes les dades que anem posant a l'usuari
- 4.-Després d'haver-la guardat, torna a l'usuari a la pàgina de l'agenda per poder veure la nota ja a la llista

Cas d'ús 14. Editar recordatoris

FUNCIONALITAT	La funcionalitat és per a poder modificar aquelles notes que ja tens creades anteriorment
PARAM. ENTRADA	Títol i descripció
PRECONDICIÓ	Haver creat ja una nota anteriorment
POSTCONDICIÓ	Nota modificada i guardada a la base de dades
ACTOR	Jo

- 1.-L'usuari pot entrar a la modificació de la nota a través de la icona que està a cada nota, que té la aparença d'un bolígraf

- 2.-Allí l'app agafa de la base de dades la nota que ja està creada i te la mostra a la pàgina d'edició
- 3.-Allí l'usuari té l'oportunitat de modificar aquells comentaris que ha posat o el títol de la nota pel que ell en prefereix
- 4.-Després de fer les modificacions corresponent, es guarda de nou a la base de dades amb les noves dades ben registrades
- 5.-Es retorna a l'usuari a la pàgina d'agenda, per mostrar els nous canvis implementats

Cas d'ús 15. Eliminar recordatoris

FUNCIONALITAT	La funcionalitat és per a poder eliminar aquelles notes que ja tens creades anteriorment
PARAM. ENTRADA	Cap
PRECONDICIÓ	Haver creat ja una nota anteriorment
POSTCONDICIÓ	Nota eliminada de la base de dades
ACTOR	Jo

- 1.-L'usuari pot entrar a l'eliminació de la nota a través de la icona que està a cada nota, que té l'aparença d'una paperera
- 2.-Si l'usuari li dona, directament la nota és eliminada de la llista
- 3.-Per part de l'app, elimina directament de la base de dades aquella nota que havia registrat

Cas d'ús 16. Veure Calendari

FUNCIONALITAT	La funcionalitat és per poder veure tots els dies que hi ha a l'any
PARAM. ENTRADA	Cap
PRECONDICIÓ	Cap
POSTCONDICIÓ	Veure aquell dia
ACTOR	Jo

- 1.-A l'usuari se li mostra una pàgina on es mostra un calendari amb tots els dies de l'any i per cada mes
- 2.-L'usuari pot seleccionar el dia que vulgui introduir un comentari

Cas d'ús 17. Llista Dia

FUNCIONALITAT	La funcionalitat és per a poder veure aquelles macros que has apuntat a una llista
PARAM. ENTRADA	Cap
PRECONDICIÓ	Haver creat ja una nota anteriorment
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-L'usuari a l'entrar en un dia pot veure una llista amb tots aquells apunts que té d'aquell dia actualment
- 2.-Te l'opció de poder afegir nous comentaris

Cas d'ús 18. Afegir comentaris

FUNCIONALITAT	La funcionalitat és per a poder afegir comentaris dins d'aquell dia per veure'ns a la llista
PARAM. ENTRADA	Títol
PRECONDICIÓ	Cap
POSTCONDICIÓ	Nova nota
ACTOR	Jo

- 1.-L'usuari prem el botó de més a la pàgina del dia seleccionat
- 2.-Llavors es mostra una nova pantalla, on mostra un quadre per posar el teu títol (on posi les kcal diàries que has consumit aquell dia)
- 3.-A continuació es guarda a la base de dades
- 4.-Et retorna a la pàgina del dia per a poder veure la nova nota

Cas d'ús 19. Eliminar nota dia

FUNCIONALITAT	La funcionalitat és per a poder eliminar aquelles notes que ja tens creades anteriorment
PARAM. ENTRADA	Cap
PRECONDICIÓ	Haver creat ja una nota anteriorment
POSTCONDICIÓ	Nota eliminada de la base de dades
ACTOR	Jo

1.-L'usuari pot entrar a l'eliminació de la nota a través de la icona que està a cada nota, sembla una paperera

Cas d'ús 20. Escaner Elecció

FUNCIONALITAT	L'aplicació té la funció de poder elegir entre l'opció d'agregar un aliment manualment o poder buscar-lo a la base de dades de menja publicada a internet
PARAM. ENTRADA	Cap
PRECONDICIÓ	Cap
POSTCONDICIÓ	Cap
ACTOR	Jo

1.-L'usuari te a elecció dos botons un que és l'encarregat de buscar l'aliment a la base de dades d'internet i l'altre d'agregar manualment l'aliment nou

Cas d'ús 21. Escaner API

FUNCIONALITAT	L'aplicació té la funció de poder escanejar els codis de barres dels menjars i buscar a través de les API a un base de dades publicat a Internet
PARAM. ENTRADA	Codi de barres
PRECONDICIÓ	Cap
POSTCONDICIÓ	Nou aliment agregat
ACTOR	Jo

1.-L'usuari entra a la pàgina d'Escanejar, on mostra per pantalla un botó per començar a escanejar els codis de barres.

2.-A partir d'aquí, a l'escanejar el codi l'app, a través de les peticions envia el codi de barres a la base de dades de Food, on allí busca quin aliment és i retorna un GET amb els nutrients i nom d'aquell codi de barres.

4.-Llavors et mostra un missatge a la pantalla on apareix que ja està registrat i ben guardat a la base de dades l'aliment amb les dades introduïdes

Excepcions:

4.1.- Si el codi de barres ja està registrat a la nostra base de dades, llavors apareix un missatge a l'usuari avisant que ja té aquest aliment i es tanca la càmera directament.

Cas d'ús 22. Escaner Manual

FUNCIONALITAT	L'aplicació té la funció de poder escanejar els codis de barres dels menjars i així poder veure registrar-los
PARAM. ENTRADA	Nom, Calories, Proteïnes, Hidrats i Grases
PRECONDICIÓ	Cap
POSTCONDICIÓ	Nou aliment agregat
ACTOR	Jo

- 1.-L'usuari entra a la pàgina d'Escanejar, on mostra per pantalla un botó per començar a escanejar els codis de barres.
- 2.-Primer et mostra uns camps a emplenar, per poder introduir les dades de l'aliment que vas a introduir (Nom, calories, proteïnes, els hidrats i les grasses)
- 3.-Després et permet poder escanejar l'aliment, on en donar-li al botó s'obre la càmera.
- 4.-Allí has de posar el codi de barres dins el quadrat que et mostra i llavors et mostra un missatge a la pantalla on apareix que ja està registrat i ben guardat a la base de dades l'aliment amb les dades introduïdes

Excepcions:

- 4.1.- Si el codi de barres ja està registrat a la nostra base de dades, llavors apareix un missatge a l'usuari avisant que ja té aquest aliment i es tanca la càmera directament.

Cas d'ús 23. Registres Menja

FUNCIONALITAT	L'usuari té la possibilitat de poder veure tots aquells aliments que ha anat guardant a la base de dades
PARAM. ENTRADA	Cap
PRECONDICIÓ	Registrar algun aliment
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-L'usuari des de la pàgina per escanejar els codis de barres, té la possibilitat de poder veure tots els aliments que ja té registrat a partir del botó que hi ha implementat a la part de dalt a la dreta, que té la forma d'una llista
- 2.-Dins aquesta pàgina l'app busca a la base de dades, tots aquells aliments que n'han estat guardats
- 3.-Llavors es mostren en forma de llista tots els aliments
- 4.-Cada aliment té el seu nom i les macros que cada un d'ells porta

Cas d'ús 24. Política de privacitat

FUNCIONALITAT	L'usuari té la possibilitat de veure la política de privacitat de la nostra app
PARAM. ENTRADA	Cap
PRECONDICIÓ	Cap
POSTCONDICIÓ	Cap
ACTOR	Jo

- 1.-L'usuari entra dins el botó d'ajuda i seguidament al botó amb la figura d'Android
- 2.-Al pulsar el botó, l'app carregarà un fitxer HTML que tenim creta en local
- 3.-Mostrarà a l'usuari la nostra política de privacitat amb el seu títol corresponent

Cas d'ús 25. IA

FUNCIONALITAT	L'usuari té la possibilitat de parlar amb la IA que està implementada per poder preguntar els dubtes que tingui
PARAM. ENTRADA	Preguntes
PRECONDICIÓ	Cap
POSTCONDICIÓ	Respostes IA
ACTOR	Jo

- 1.-L'usuari entra dins el Home on tindrà el voto d'Android que el dirigirà a la pantalla de xat
- 2.-L'usuari formula la seva dubte a la nostra IA
- 3.-L'app es connectarà via IP al servidor que n'hem creat
- 4.-Envia la dubte que ha fet a través de la connexió establerta anteriorment
- 5.-Si el xatbot que hem creat al servidor sap la dubte li enviarà la resposta corresponent, si no és així li enviarà una llista de les dubtes que en sap contestar
- 6.-L'usuari veure per pantalla les respostes que es van trametent i el que ell pregunta (Com si fos un whatsapp)

4.-Entitats del projecte

Un cop acabada l'explicació dels casos d'aplicació de la nostra app, continuarem amb el punt d'exposar les entitats del projecte en qüestió.

Les entitats són les classes principals que representen els objectes importants en el projecte, aleshores són les peces clau del nostre model de dades i són essencials per al funcionament d'aquest.

Tenim totes les classes que representen les diferents entitats de la nostra aplicació, en diferents parts dins el projecte, cadascun al seu lloc respectivament. Aquestes classes defineixen les propietats i els comportaments específics de cada entitat i són utilitzades per interactuar amb les dades del sistema.

L'organització adequada de les entitats facilita la seva gestió i manteniment. A més, tenir una visió clara de les entitats ens ajuda a comprendre l'estructura del nostre sistema i les relacions entre elles.

És important tenir en compte les interaccions entre les entitats i com es relacionen entre si. Això ens permetrà dissenyar un sistema robust i escalable.

Les diferents entitats que hem anat registrant dins la nostra app, són les següents:

- User: És l'entitat més important del nostre projecte, ja que és l'encarregada de registrar tots els usuaris que anem afegint a la nostra app, amb les seves dades personals i la seva contrasenya respectiva, per poder iniciar sessió sempre que vulgui.
- IMC: És l'encarregada de guardar tots els registres d'IMC que té un usuari cada vegada que fa un càlcul del seu IMC.
- Notes Recordatoris: Té la funció de guardar per cada usuari els comentaris que té com a recordatoris, que és la primera pàgina que utilitzem d'Agenda.
- Notes per Dies: És l'encarregada de guardar per cada usuari i per cada dia dels anys les anotacions que anem fent.
- Escàner: És l'encarregat de guardar per cada un dels usuaris, els seus aliments que va registrant, i després poder mostrar-los tots.

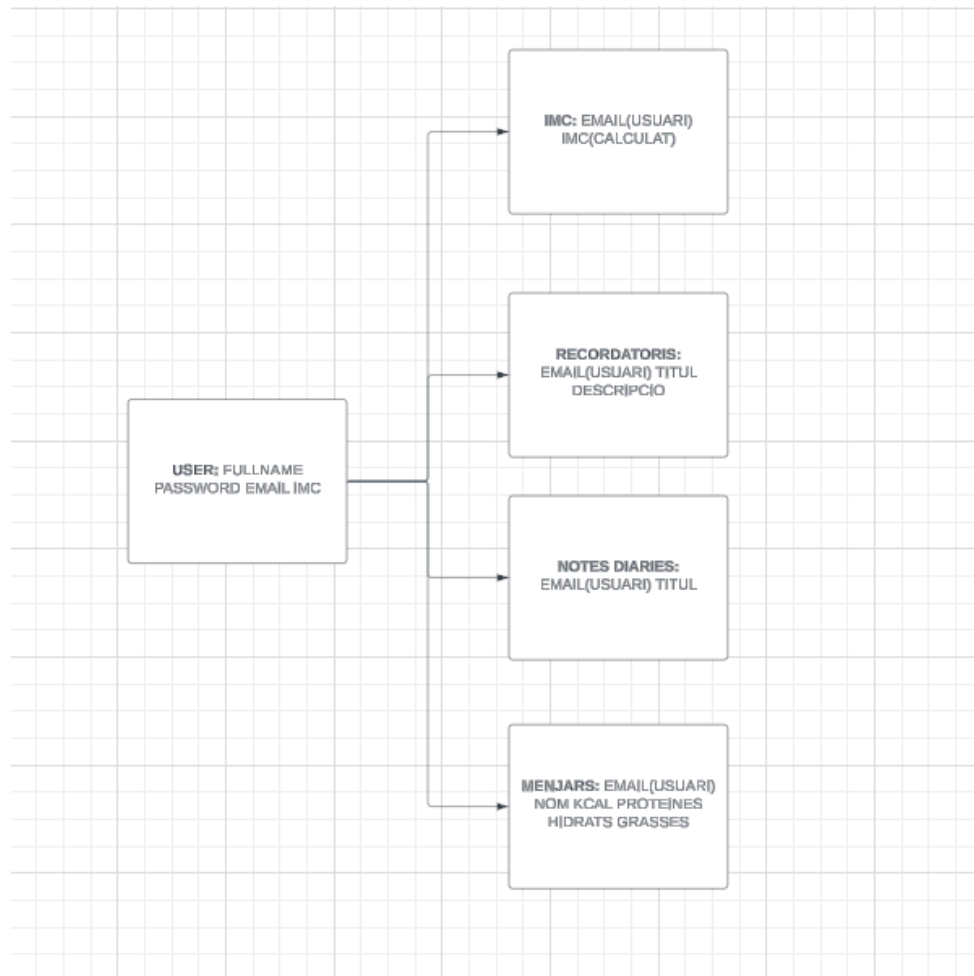


Diagrama 2. Entitats

5.-Disseny

Després d'haver fet una introducció amb les diferents tecnologies i tots els requisits que he implementat a l'aplicació, us començaré a explicar tota la part del disseny de l'aplicació.

L'aplicació està separada en tres parts diferents que són les següents:

- Disseny de l'arquitectura Software
- Disseny de la interfície
- Disseny de la base de dades

5.1 Disseny de l'arquitectura Software

Cada aplicació mòbil requereix una arquitectura de programació sòlida i ben definida des del començament del desenvolupament. Avui en dia, el desenvolupament d'aplicacions mòbils s'ha convertit en una activitat important en la indústria tecnològica. La creixent demanda d'aplicacions mòbils requereix el desenvolupament d'aplicacions que no només siguin funcionals, sinó també d'alta qualitat, escalables i fàcils de mantenir.

Per assolir aquests objectius, és crucial tenir una arquitectura de programació ben dissenyada i planificada des de les primeres etapes del procés de desenvolupament. Una arquitectura ben definida proporciona una estructura clara i cohesionada a una aplicació, cosa que facilita el desenvolupament, la comprensió i el manteniment al llarg del temps.

Una arquitectura per a aplicacions mòbils ha de tenir en compte diversos aspectes, com ara:

- **Divisió en capes:** Una estructura en capes ens permet diferenciar entre diferents parts de l'aplicació, com ara la interfície d'usuari, la lògica empresarial i l'accés a les dades. Això facilita la modularització i la reutilització del codi.
- **Patrons de disseny:** L'ús de patrons de disseny com MVC (Model-Vista-Controlador), MVVM (Model-Vista-ViewModel) així ajudant-nos a organitzar el codi i a mantenir-lo escalable i mantenible.
- **Gestió de dependències:** Les biblioteques externes es poden gestionar i integrar fàcilment amb el nostre projecte mitjançant un gestor de dependències com Gradle per a Android o CocoaPods per a iOS.
- **Seguretat:** Una arquitectura segura és important per protegir les dades dels usuaris i prevenir vulnerabilitats. Això inclou l'ús de protocols de comunicació segurs com els que hem implementat, el HTTP, xifratge de dades (hashing) i la gestió adequada d'autorització i autenticació que implementem.

L'arquitectura d'una aplicació mòbil és el conjunt de decisions sobre com organitzem el nostre programa. Aquestes decisions inclouen elements i interfícies estructurals i l'estil general del programa.

En resum, l'arquitectura d'una aplicació mòbil defineix l'estructura i el comportament del sistema.

Una bona arquitectura implica simplicitat i eficiència. Per tant, ens permet estalviar temps i esforç durant el procés de desenvolupament de l'aplicació. A més, fa que l'aplicació sigui més fàcil de modificar, provar i depurar.

Els beneficis d'una aplicació amb una bona arquitectura són molts:

- **Eficiència:** Una aplicació amb una arquitectura ben dissenyada pot realitzar tasques i funcions en qualsevol condició, garantint un sistema productiu i fiable.

- **Flexibilitat:** Això permet fer canvis sense que afectin negativament a les altres capes que ja tenim dissenyades.
- **Mantenibilitat:** Un bon disseny de software facilita la comprensió i modificació del codi, ja que està ben estructurat i organitzat. Això ajuda a mantenir el software amb menys errors i a fer canvis més eficientment.
- **Reutilització del codi:** Un bon disseny de programari afavoreix la reutilització de codi, permetent la creació de mòduls que es poden utilitzar en diferents parts del sistema, estalviant així temps en la creació de nou codi.
- **Extensibilitat:** Es poden afegir tantes funcions com sigui necessari a l'aplicació, amb un disseny que permeti adaptar-se fàcilment als canvis
- **Escalabilitat:** Un bon disseny de software ha de ser escalable, adaptant-se i ampliant-se per satisfer les nostres necessitats en el futur, fonamental per a empreses que volen créixer i expandir-se.
- **Testabilitat:** L'aplicació es pot provar fàcilment, amb menys errors i més confiança. Els elements o mòduls de l'aplicació són independents, facilitant les proves i el manteniment.
- **Comprensió:** El codi ha de ser entès per tantes persones com sigui possible per garantir la continuïtat del projecte i la col·laboració entre els diferents membres de l'equip de desenvolupament, si escau el treball en grup.
- **Facilita la col·laboració:** Un bon disseny de software facilita la col·laboració entre els membres de l'equip de desenvolupament, amb una estructura ben documentada que ajuda a comunicar clarament les idees i les expectatives, assegurant que tot l'equip treballi en la mateixa direcció.

En resum, una arquitectura ben planificada és essencial per a l'èxit d'una aplicació mòbil, garantint un funcionament eficient, mantenible i escalable a llarg termini.

5.2.-Disseny de la interfície

Dissenyar una interfície d'aplicació eficaç és molt important per proporcionar una experiència a l'usuari satisfactòria i fer que l'aplicació sigui fàcil d'utilitzar i entenedora per als nostres usuaris. Per això hem de seguir uns punts claus a l'hora de fer el disseny de la interfície de l'aplicació.

Per això per poder garantir una experiència d'usuari perfecta, la interfície ha de ser fàcil d'utilitzar, eliminant la necessitat que els usuaris facin referència a manuals o guies.

El disseny ha de prioritzar la simplicitat i la coherència a tot arreu. La coherència en la part del front-end i el comportament de tots els elements de l'aplicació és molt important per evitar confusions de l'usuari i facilitar la navegació. A més, el disseny ha de ser adaptable per

atendre els usuaris amb discapacitat visual, auditiva o motriu, garantint l'accessibilitat per a tothom.

Un disseny visualment atractiu i agradable pot millorar l'atractiu general de la nostra aplicació i fomentar el seu ús continu. En últim punt, l'aplicació ha de satisfer les expectatives dels usuaris i realitzar de manera eficient les tasques previstes.

Els avantatges d'un bon disseny d'interfície d'usuari inclouen una millor usabilitat, una major eficiència en l'ús de l'aplicació, una millor satisfacció de l'usuari i una reducció dels errors. Això ens permet tenir una millor retenció d'usuaris.

En resum, una interfície a la nostra aplicació ben dissenyada millora la satisfacció de l'usuari i la seva experiència amb l'aplicació, així com la percepció de la marca i la seva confiança amb l'aplicació, aspectes essencials per a nosaltres.

Aquestes bases van servir-me per fer els primers esbossos per començar a dissenyar l'aplicació i veure quines característiques hauria de tenir per poder ser fàcil d'utilitzar, a part de tenir un seguit de requisits els quals he mencionat anteriorment. A partir d'aquí vaig començar la construcció de l'app:

A.- Entrada

La següent imatge ens mostra l'entrada a la nostra app, indicant-nos que ja podem accedir a la nostra aplicació, i les diferents maneres que tenim per accedir-hi. Podem elegir entre registrar-nos si no tenim un compte ja creat a la nostra app o per l'altra banda el inici de sessió on pots accedir a l'aplicació tenint ja un compte creat.

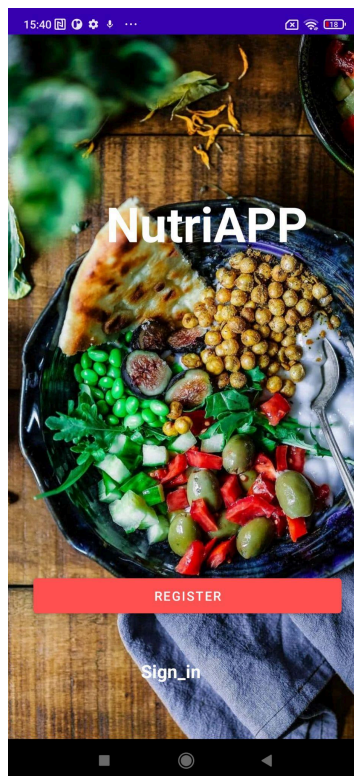


Figura 13. Entrada

B.- Login i Register

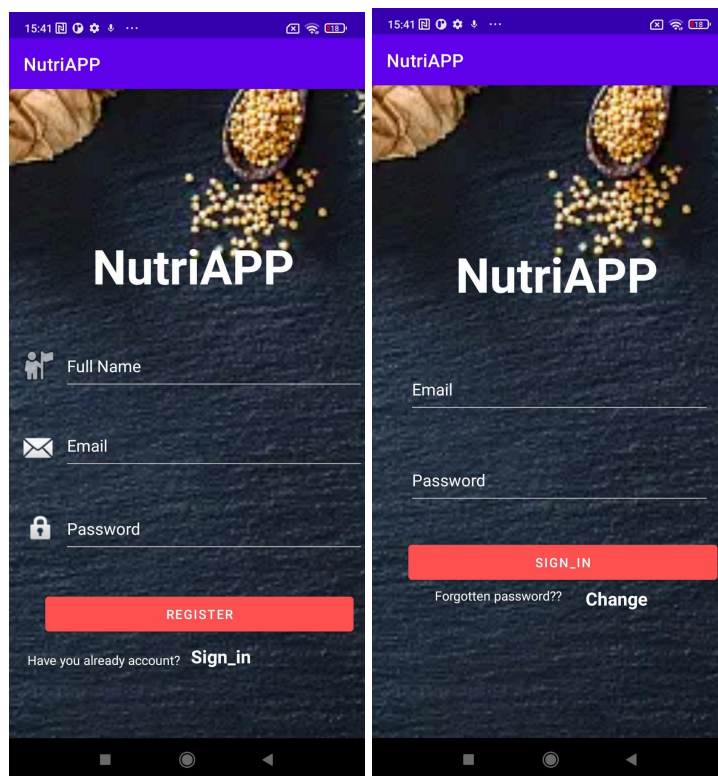


Figura 14 i 15. Login i register

En aquestes dues imatges podem observar, com hem dit anteriorment les dues maneres d'introduir-te a la nostra app.

la primera fotografia es la que utilitzarem per registrar.nos, en la qual podem observar els diferents camps.:

- Nom: Es el primer camp que podem observar. Té un disseny simple i es fàcilment accessible ja que destaca per tenir una mida adequada, i es on l'usuari posarà el seu nom que utilitza a l'app.
- Email: Aquest camp ens permet introduir el email, que es un camp crucial per la verificació del compte.
- Contrasenya: Aquest camp permet als usuaris establir una contrasenya segura per al seu compte.
- Botó de Registre: Aquest botó és gran i destacat, situat sota els camps de text. La seva mida més gran i el color destacat ajuden a atreure l'atenció de l'usuari, facilitant que sàpiga on ha de fer clic per completar el procés de registre.

A banda està l'opció d'anar directament a la pantalla d'inici, per si t'has equivocat i vols entrar d'una forma diferent..

Per l'altra banda, la segona foto ens mostra dos camps, els quals hem anat comentant anteriorment, et demana un email i la contrasenya del teu usuari, captant l'atenció de l'usuari e

indicant per poder entrar a l'app amb la teva sessió. A part té l'opció, de poder modificar la contrasenya, per si has tingut algun problema i no te'n recordes.

- Camp d'Email: Aquest camp es troba a la part superior de la pantalla d'inici de sessió i demana als usuaris que introdueixin la seva adreça de correu electrònic associada al compte. El seu disseny és clar i accessible, facilitant la introducció de dades.
- Camp de Contrasenya: Sota el camp d'email, aquest camp permet als usuaris introduir la seva contrasenya. La seva ubicació immediatament sota el camp d'email fa que sigui fàcil de trobar i utilitzar.
- Botó d'Inici de Sessió: Aquest botó és gran i de color destacat, similar al botó de registre, situat sota els camps de text. La seva mida i color ajuden a captar l'atenció de l'usuari i indicar clarament l'acció a seguir per iniciar la sessió.

C.- Canvi de contrasenya

En aquestes imatges podem observar que semblen dues pàgines diferents, però no és així, és la mateixa amb els camps ocults, per únicament mostrar-se si l'email que introdueixes existeix a la base de dades.

A través del botó lila que podem observar que té la funcionalitat d'enviar l'email que introduïm al quadre de diàleg, si existeix, llavors ens mostra el que veiem a la segona foto, on estan els dos quadre de diàleg per poder introduir la nova contrasenya i repetir-la, i amb el botó de guardar es modifica la contrasenya anterior per la nova introduïda.

A la primera imatge tenim els botons següents:

- Camp d'Email: L'usuari introdueix el seu email en aquest camp. Aquest camp és visible per defecte i permet als usuaris introduir les seves dades.
- Botó d'Enviar Email: Aquest botó lila permet enviar un correu electrònic de verificació. Si l'email existeix a la base de dades, es mostra una finestra emergent amb els camps per canviar la contrasenya.

A la segona imatge tenim els botons següents:

- Camp Nova Contrasenya: En aquest camp, l'usuari pot introduir la nova contrasenya que desitgi utilitzar. Aquest camp és visible únicament després de verificar l'email.
- Camp Repetir Nova Contrasenya: Aquest camp permet a l'usuari confirmar la nova contrasenya introduint-la de nou. Aquest camp és visible només quan l'usuari ha introduït una nova contrasenya.
- Botó de Guardar: Aquest botó permet desar els canvis i modificar la contrasenya anterior per la nova introduïda. La seva mida i color ajuden a captar l'atenció de l'usuari i indicar clarament l'acció a seguir per guardar la nova contrasenya.

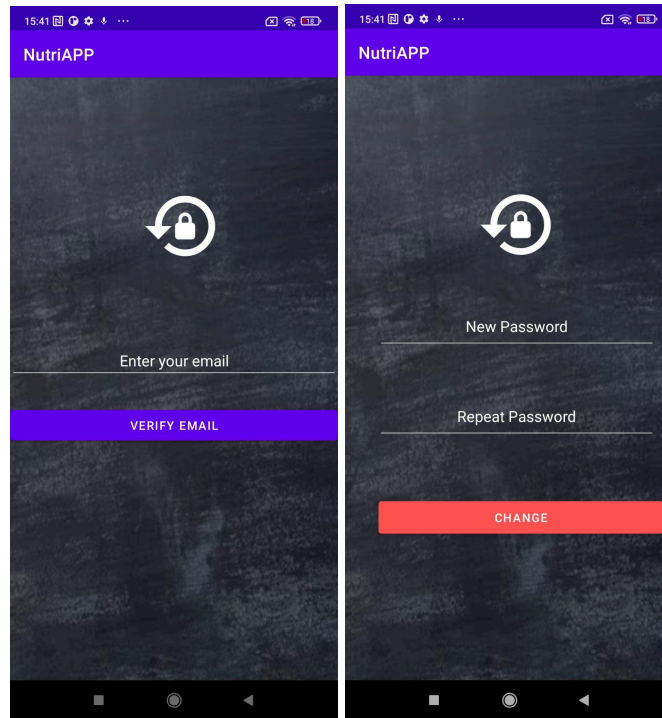


Figura 16 i 17. Canvi de contrasenya

D.- Home

A la figura podem apreciar que és la pantalla principal on es visualitzen les funcions que te l'usuari a l'entrar dins de l'aplicació. Podem observar com la pantalla principal consta d'un menú de navegació de totes les opcions a escollir per l'usuari, on es mostren les diferents opcions disponibles de l'aplicació, cadascuna amb una imatge representativa i un text instructiu que indica la funcionalitat o destinació d'aquella opció. Aquest disseny facilita la comprensió i utilització de l'aplicació pels usuaris, ja que ofereix una guia visual clara de les funcionalitats disponibles i com accedir-hi.. També tenim disponible a dalt a la dreta dos botons en forma de persona i d'interrogant, que són els encarregats de dirigir cap a la vista de perfil o d'ajuda.

Podem observar les diferents opcions que ens mostra l'app, on està diferenciat per cada imatge amb el seu text instructiu davall, on fa referència a on dirigeix aquest botó, on són els camps del càlcul d'IMC, a l'agenda i calendari personals, i finalment a la llista dels menjars i la càmera per poder registrar aliments.



Figura 18.Home

E.- Perfil

Aquí podem observar com es mostra cada camp que hem introduït en registrar-nos a l'app menys la contrasenya ja que és privada, podem veure també el nostre últim IMC que hem calculat i ha quedat registrat.

Al final de la pàgina tenim un text on posa "Modificar" que és l'encarregat de enviar-te a la nova pàgina per modificar les teves dades personals. Aquesta funcionalitat proporciona flexibilitat i control a l'usuari sobre la seva informació de perfil.

A part, podem observar que dalt a la dreta tenim dos botons, un amb forma de casa i l'altre amb forma de persona amb una ratlla, la primera és per poder tornar al home i continuar fent el que vulguem i l'altra té l'opció de sortir de la nostra sessió i poder posar una altra o per tancar-la únicament.

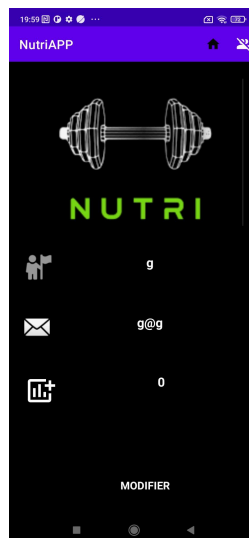


Figura 19.Perfil

F.- Modificar el perfil

A la fotografia següent podem observar que ens mostren tots els camps que podem modificar de la nostra sessió, on aquí si es pot modificar la contrasenya. Amb el botó de guardar, quan premem es registraran les noves dades que hem introduït i les modifica canviant-les i esborrant les anteriors.

Els diferents botons que tenim implementats son:

- Nom: Es el primer camp que podem observar. Té un disseny simple i es fàcilment accesible ja que destaca per tenir una mida adequada, i es on l'usuari posarà el seu nou nom que utilitza a l'app.
- Email: Aquest camp ens permet modificar el email, que es un camp crucial per la verificació del compte.
- Contrasenya: Aquest camp permet als usuaris canviar la contrasenya per al seu compte.
- Botó de Guardar: Aquest botó és gran i destacat, situat sota els camps de text. La seva mida més gran i el color destacat ajuden a atreure l'atenció de l'usuari, facilitant que sàpiga on ha de fer clic per completar el procés de modificació.

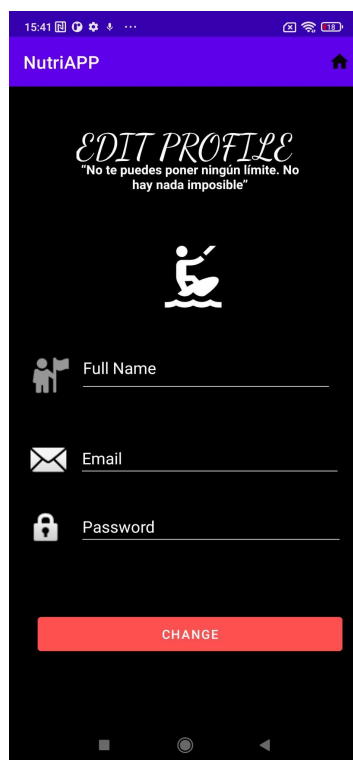


Figura 20. Modificacions del perfil

G.-Agenda i Afegir nota a l'agenda

Aquí podem observar que hi ha una nota, i a mesura que anem afegint anotacions es mostren de forma consecutiva a la nostra pàgina les quals les podem anar guardant a la base de dades.

A la part superior podem observar dues icones, una té la forma d'una casa i l'altra té la forma d'un calendari, allí podem decidir si volem tornar al home per fer altres coses o anar al calendari que hi ha.

Finalment, a la part inferior, podem veure que hi ha un botó d'afegir amb el símbol d'un més, que té la funció de poder agregar noves notes a la pàgina actual que estem, si premen ens dirigeix a la figura "22", on veiem que podem introduir un títol i uns comentaris per cada una de les notes que fem, quan això ja està introduït hi ha un botó, que té la forma d'un tick, el qual té la funció de guardar la nova anotació a la base de dades.

Aquest botons han estat implementats per ajudar visualment als usuaris:

- Botó d'afegir: La presència d'un botó d'afegir amb una icona clara en forma de suma (+) destaca visualment i crida l'atenció de l'usuari, indicant clarament la funció d'afegir noves notes.
- Estructura de Camps: L'ús de camps separats per al títol i els comentaris proporciona una estructura clara i ordenada al formulari, facilitant a l'usuari la introducció de la informació necessària.
- Botó de Guardar: La inclusió d'un botó de guardar amb una icona reconeixible en forma de marca de verificació (tick) indica clarament a l'usuari la funció d'aquest botó i com confirmar i finalitzar l'acció d'afegir una nova nota.

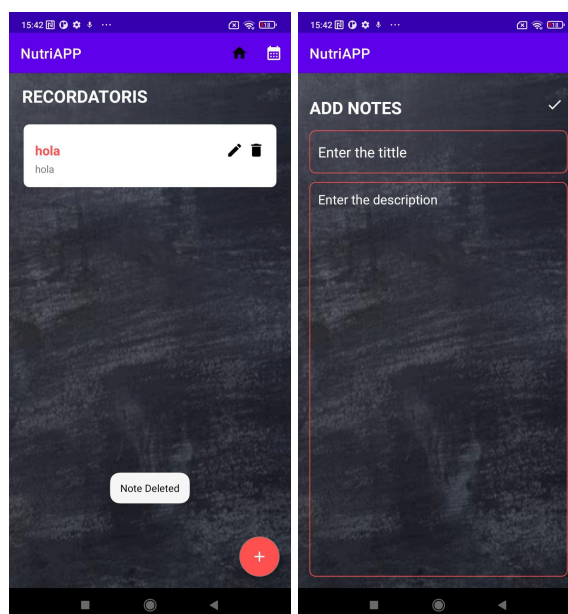


Figura 21 i 22. Recordatoris i Afegir recordatoris

H.- Calendari

A l'entrar aquesta pàgina podem veure un calendari que ens mostra cada dia de l'any i el seu més corresponent, tenint l'opció de poder seleccionar el dia que nosaltres vulguem.

Amb la peculiaritat, que quan els usuaris seleccionen una data, la casella del dia es destaca visualment amb un color blau, proporcionant una retroalimentació visual immediata per confirmar que la data ha estat seleccionada amb èxit. Aquesta característica ajuda a millorar la claredat i la comprensió de la interacció de l'usuari amb el calendari.

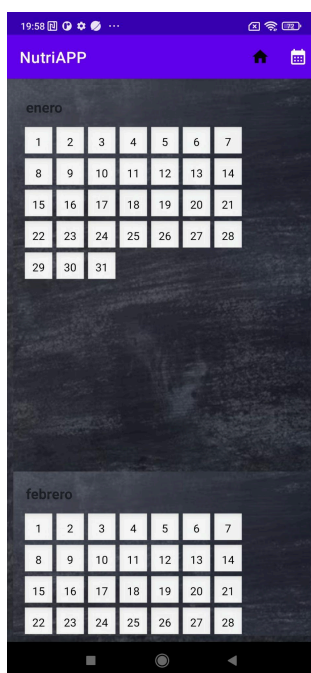


Figura 23. Calendari

I.- Dia seleccionat i Afegir notes al dia

Aquí podem observar que hi ha una anotació, i a mesura que anem afegint notes les podem veure de forma consecutiva a la nostra pàgina, els nous comentaris que anem guardant a la base de dades.

Finalment, a la part inferior, podem veure que hi ha un botó d'afegir amb el símbol d'un més, que té la funció de poder agregar noves notes a la pàgina actual que estem, si premen ens dirigeix a la figura "25", on veiem que podem introduir un títol per cada una de les notes que fem, quan això ja està introduït hi ha un botó, que té la funció de guardar la nova nota a la base de dades.

Aquests botons han estat implementats per ajudar visualment als usuaris:

- Botó d'afegir: La presència d'un botó d'afegir amb una icona clara en forma de suma (+) destaca visualment i crida l'atenció de l'usuari, indicant clarament la funció d'afegir noves notes.

- Estructura de Camps: L'ús de camps separats per al títol i els comentaris proporciona una estructura clara i ordenada al formulari, facilitant a l'usuari la introducció de la informació necessària.
- Botó de Guardar: La inclusió d'un botó de guardar amb una icona reconeixible en forma de marca de verificació (tick) indica clarament a l'usuari la funció d'aquest botó i com confirmar i finalitzar l'acció d'afegir una nova nota.

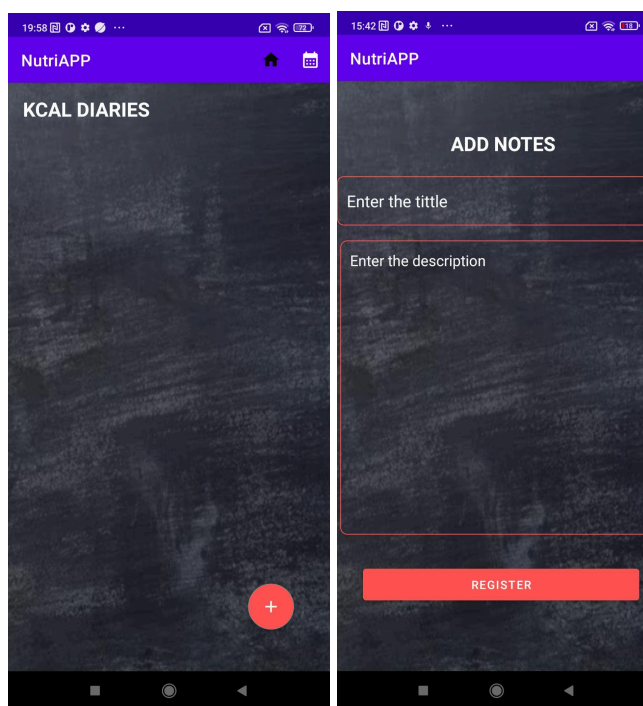


Figura 24 i 25. Notes i Afegir notes

J.- Càlcul IMC

A la següent figura podem observar dos textos des dels qual podem obtenir la funció de demanar els camps del pes i l'altura, on nosaltres mateixos hem de introduir-los, a prop de cada camp de text, es mostra un text descriptiu que indica clarament què ha de ser introduït en cada camp. Aquesta retroalimentació visual ajuda els usuaris a comprendre quina informació s'espera que introdueixin en cada cas, situat just a baix dels textos mencionats anteriorment hi ha un text on al costat surt el resultat del càlcul realitzat anteriorment cosa que permet la visualització immediata del resultat permeten als usuaris veure la seva informació de manera clara i immediata.

A la part inferior hi ha dos botons, un d'ells és l'encarregat de fer els càlculs al pressionar i l'altre l'encarregat de passar a la següent pàgina, un cop finalitzats els càlculs corresponent. A la part superior, també tenim dues icones, una d'elles representa una casa, on té la funció de torna al home i l'altra que té la forma d'unes estadístiques, que té la funció de dirigir-te al registre dels càlculs fets anteriorment de l'IMC

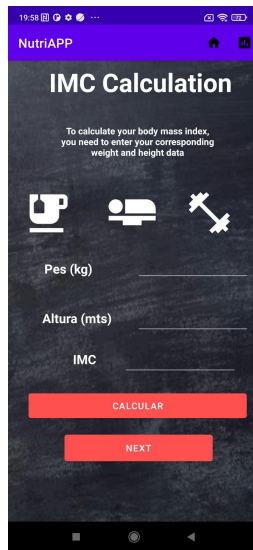


Figura 26.IMC càlcul

K.- Registres IMC

Com podem observar a la fotografia, únicament podem veure en aquesta pàgina un registre continu de l'IMC que hem anat calculant durant tot el temps que hem utilitzat l'app, aquesta presentació permet als usuaris veure clarament com han canviat els seus resultats al llarg del temps. Aquesta retroalimentació visual pot motivar als usuaris a mantenir hàbits de vida saludables i seguir millorant el seu estat de salut. fent que els usuaris puguin fer un seguiment del seu progrés i identificar tendències a llarg termini en la seva condició física

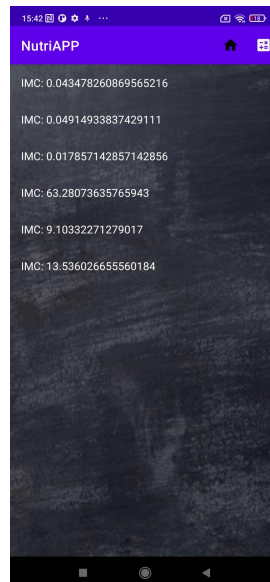


Figura 27.IMC

L.-Solucions IMC

Aquí podem observar una taula amb els diferents tipus d'IMC segons el valor que has donat. La funcionalitat d'aquesta pàgina es mostra la persona situada a la franja corresponent del seu IMC, de manera clara i ordenada facilitan als usuaris identificar ràpidament en quin rang d'IMC es troben i quina és la seva classificació de pes actual, en el quadre de text situat a la part inferior es conta en detall la posició en la franja i proporciona una breu explicació dels canvis que hauries de fer segons la franja on et trobes.



The screenshot shows the 'NutriAPP' interface with a dark background. At the top, the status bar shows the time 19:58 and various icons. Below the app name 'NutriAPP' and a home icon, the title 'IMC' is displayed in large white letters. A small icon of a person is positioned above the first row of the table. The table lists BMI categories with their corresponding ranges. The bottom part of the screen is a dark, textured area.

Tipus d'IMC	Rang
BAJO PESO	<18.5
PESO NORMAL	18.5-24.9
SOBREPESO	25-29.9
OBESIDAD TIPO 1	30-34.9
OBESIDAD TIPO 2	35-39.9
OBESIDAD TIPO 3	≥40

Figura 28.IMC solucions

M.- Escàner Manual i API

Com podem observar en aquesta fotografia, hi ha cinc camps a la part inferior, on cada un d'ells té la funció de poder posar el nom de l'aliment que vols registrar, i els seus nutrients (els carbohidrats, proteïnes, grasses i calories), quan aquests camps estan omplerts, s'activa el botó de "SCAN", on el qual al pressionar-lo s'obre la càmera i surt una sol·licitud per poder utilitzar la càmera i seguidament quan acceptes té mostra un quadre on has de situar el codi de barres per poder escanejar-lo i guardar-lo.

A la següent imatge tenim únicament un botó on tens l'opció d'escanejar el nou codi de barres, el qual en pressionar-lo escaneja directament el codi de barres que enfoques amb la càmera, i en detectar-lo et surt un missatge informant del nou aliment registrat.

A la part superior també tenim dues icones on un és la casa que com hem anat mencionant anteriorment, te la funció de tornar-te a la pàgina inicial i l'altra icona és una llista on et porta al layout on es van mostrant tots els aliments que vas registrant.

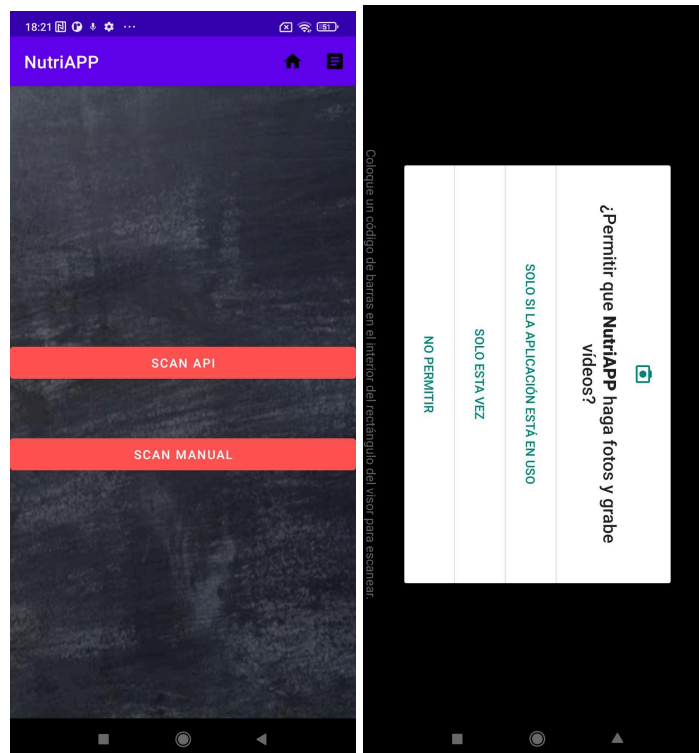


Figura 28 i 29. Elecció i permisos camara

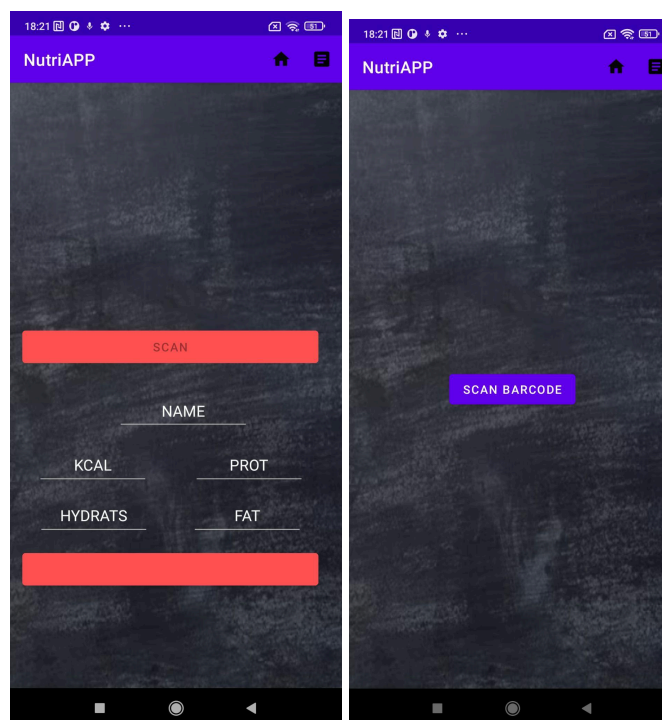


Figura 30 i 31. Escaner manual i API

N.-Llista menjars

Aquesta pàgina és un recyclerView, com podem observar a la fotografia, et va mostrant per pantalla tots aquells aliments que vas registrant a la pantalla anterior mencionada anteriorment (pàgina d'escanejar), on cada un dels aliments mostra el seu nom que li hem agregat i seguidament els nutrients que has agregat per aquelles quantitats que tu necessites

Això permet als usuaris navegar eficientment per la llista d'aliments, desplaçant-se cap amunt i cap avall per veure els aliments registrats i així oferint una disposició ordenada dels aliments facilitan als usuaris trobar ràpidament els aliments específics que estan buscant.

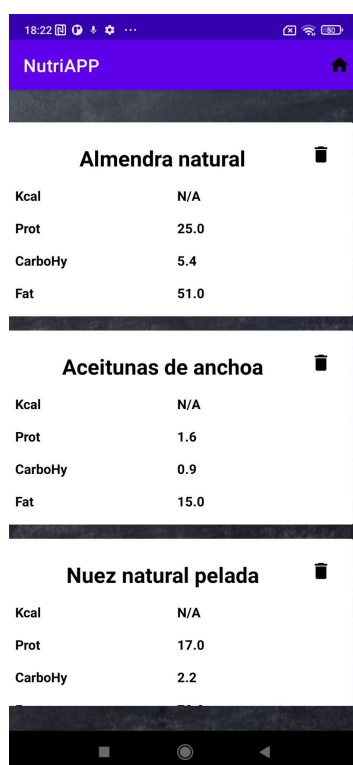


Figura 32.Llista menjars

O.- Ajuda

Finalment, tenim el botó d'ajuda que et dirigeix a la fotografia següent, on et mostra en diferents vinyetes, les diferents funcions que proporciona la nostra app i quina és la seva funcionalitat a través d'un breu text que hi ha a la part inferior, davall d'una fotografia representativa dels botons que hi han configurats i un breu títol per diferenciar-los.

Per altra part, com hem anat mencionant anteriorment tenim una icona a la part superior, en forma de casa, per poder tornar al Home de l'app.

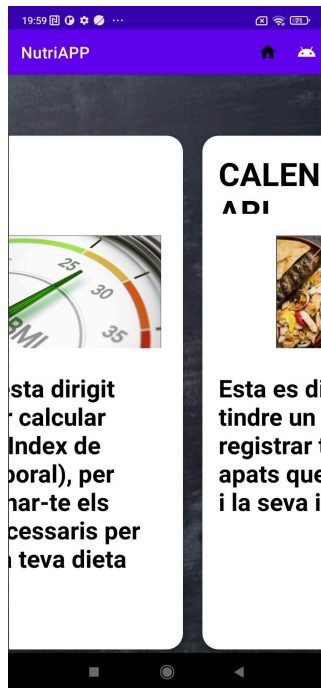


Figura 33. Tutorial

P.- Política de Privacitat

Finalment, tenim el botó d'Android que et dirigeix a la fotografia següent, on et mostra en diferents línies les diferents polítiques de privacitat que hem implementat en aquest projecte, les diferents funcions que proporciona la nostra app.



Figura 34. Política de privacitat

Q.- Xat

Finalment, tenim el botó d'Android que et dirigeix a la fotografia següent, on et mostra una pantalla amb el fons de l'app i a la part de baix està situat un text per escriure i un botó al seu costat per enviar les preguntes formulades.

A part a la pantalla, es van mostrant en forma de diàleg qui és la persona que parla si tu o la IA en cada moment i es van veient les converses que es fan entre si, així tenim en tot moment en detall el que has anat preguntant.

Aquestes decisions de disseny ajuden als usuaris de diferents maneres:

- **Quadre de Text:** Aquest quadre de text està dissenyat per ser fàcilment accessible i utilitzable, amb una mida adequada per a una entrada de text còmoda.
- **Botó d'Enviar:** Al costat del quadre de text, hi ha un botó d'enviar que permet als usuaris enviar les seves preguntes o comentaris a la IA. El botó està ben destacat, assegurant que els usuaris poden enviar els seus missatges sense cap dificultat.
- **Diàleg:** Les converses es mostren en forma de bombolles de diàleg, indicant clarament qui està parlant en cada moment. Les bombolles del text de l'usuari i les de la IA es diferencien visualment per posició, facilitant la lectura i la comprensió de la conversa.



Figura 35.Xat

5.3.- Disseny de la tecnologies utilitzades

En el desenvolupament de la nostra aplicació Android enfocada en el sector de l'alimentació, he utilitzat una combinació de tecnologies per garantir una experiència a l'usuari fluida i eficient. A continuació, explicaré les diferents tecnologies i els conceptes que he utilitzat.

Per a la realització del codi he utilitzat Kotlin com a llenguatge principal per al desenvolupament de l'aplicació.

Kotlin és un llenguatge modern, estretament integrat amb la plataforma Android, que ofereix una sintaxi clara i concisa. Els avantatges d'utilitzar Kotlin inclouen una millor gestió de la nul·labilitat, com per exemple, les variables en Kotlin que per defecte no poden contenir valors nuls, a menys que es marquin explícitament amb un `?`, o les extensions de funcions, que permeten afegir noves funcionalitats a les classes existents sense necessitat de modificar-les.

Per al disseny de les interfícies d'usuari, hem combinat l'ús de Jetpack Compose i els antics XMLs:

- Jetpack Compose: Simplifica el procés de disseny d'UI mitjançant un model declaratiu, on descrius l'UI en funció de l'estat de l'aplicació. Aquest enfocament permet una major flexibilitat i una millor integració amb altres components de l'arquitectura d'Android, és a dir, en lloc d'escriure molts codis complicats per definir com es veurà la teva aplicació, et permet descriure fàcilment l'aparença que desitges, com per exemple, com ara quins botons es mostren o com es disposen els elements en pantalla.
- XML: Es un tipus de codi que s'utilitza per descriure com ha de ser l'aspecte de les pantalles d'una aplicació d'Android, és a dir, son un conjunt d'instruccions senzilles que li diem al nostre dispositiu per saber on posar els botons, textos i altres elements visuals en la pantalla alhora de generar la nostra aplicació.

A part dins el projecte hem implementat LiveData i Coroutines, que son components clau per a la gestió de l'estat i la programació asíncrona a l'aplicació.

LiveData l'hem utilitzat per mantenir i observar dades dins del ViewModel. Quan les dades canvien, LiveData notifica automàticament les vistes associades, actualitzant la interfície d'usuari de manera reactiva i sense necessitat de codi extra per a la gestió del cicle de vida. Això és útil perquè així la nostra aplicació pot actualitzar-se automàticament amb la informació més recent, sense la necessitat que els nostres usuaris hagin de fer res.

En canvi les coroutines de Kotlin ens permeten escriure codi asíncron de manera més senzilla i eficient. S'han utilitzat coroutines per manejar operacions que requereixen temps, com l'accés a la base de dades així evitant bloquejar el fil principal de l'aplicació. Això assegura que la interfície d'usuari estigui operativa i proporciona una experiència d'usuari fluida.

En resum les coroutines actuen com a treballadors que gestionen tasques complicades de manera eficient, com per exemple, quan fem una crida a la base de dades, alhora de realitzar el escaneig i comprovar si ja existeix aquest aliment, les coroutines s'encarreguen d'aquestes feines en segon pla, sense afectar el rendiment general de la nostra aplicació.

5.4.- Disseny de la base de dades

Quan vaig començar el projecte, tenia moltes opcions de bases de dades per triar. Cada opció tenia els seus avantatges i inconvenients, i vaig passar molts dies investigant i analitzant quina seria la millor opció per a les necessitats del meu projecte.

La meua idea principal era emmagatzemar totes les dades en un sistema Room, el qual té una gran quantitat d'avantatges per què és una biblioteca de persistència de dades proporcionada per Google com a part de l'Android.

Finalment, vaig decidir utilitzar Room per al meu projecte. A continuació vaig estar fent una gran recerca, on vaig tenir en compte diversos avantatges que aquesta biblioteca ofereix i com podia resoldre alguns dels reptes habituals associats amb la gestió de bases de dades.

Primerament, vaig descobrir que room proporciona una capa d'abstracció sobre SQLite que simplifica l'accés a la base de dades[6]. En comptes de treballar directament amb consultes SQL complexes i susceptibles a errors, room permet definir consultes utilitzant anotacions dins de classe de Kotlin. Això redueix significativament la quantitat de codi que cal escriure i mantenir, facilitant al mateix temps la lectura i comprensió del codi.

Un altre dels avantatges destacades de Room és la integració amb les LiveData i les coroutines de Kotlin. LiveData és una classe de dades observables que té consciència del cicle de vida, el que significa que actualitza automàticament les interfícies d'usuari quan les dades canvien, evitant així fugues de memòria i errors comuns en la sincronització de dades amb la interfície d'usuari.

Però mentre duia a terme el projecte, quan vaig començar a necessitar més d'una base de dades, vaig començar a tenir problemes a l'hora de demanar les dades, ja que a l'estar totes les bases de dades actives, la nostra app no ho aguantava bé, cosa que em va fer decidir canviar d'opinió i finalment vaig decidir utilitzar el Room per guardar als usuaris únicament, ja que era l'entitat més important del projecte i la resta de base de dades les vaig crear amb SQLite Database.

SQLite té una gran varietat en la persistència de dades en aplicacions mòbils per diverses raons. A diferència de Room, que és una capa d'abstracció sobre SQLite, SQLite és una biblioteca de base de dades relacional lleugera que implementa un SQL autocontingut, sense servidor i sense configuració.

Una de els grans avantatges que em proporciona SQLite és la seva simplicitat i eficiència. És una de les bases de dades més utilitzades a causa de la seva facilitat d'integració i el seu petit pes.

Una de les característiques principals de SQLite és que tota la base de dades resideix en una única llibreria de codi, i les dades es guarden en un sol fitxer del sistema de fitxers. Això fa que la seva gestió sigui molt senzilla i que la seva integració sigui directa, sense necessitat d'una configuració de servidor o d'una instal·lació complicada.

5.4.1.- Arquitectura del Room

Aquesta biblioteca (Room) segueix els principis de l'arquitectura de components d'Android, la qual cosa ajuda a crear aplicacions més robustes, mantenibles i eficients.

Room es compon principalment de tres components:

- Entitats (Entities): Les entitats representen les taules de la base de dades. Cada entitat és una classe anotada amb `@Entity` i els seus camps es corresponen amb les columnes de la taula. On les seves variables es poden definir com a claus primàries, claus foranes, índexs i altres restriccions.
- DAOs (Data Access Objects): Els DAOs són interfaces que defineixen els mètodes per accedir a la base de dades. S'anoten amb `@Dao` i contenen mètodes per a operacions de consulta, inserció, actualització i esborrat. Room genera automàticament les implementacions d'aquests mètodes en temps de compilació, el que redueix la quantitat de codi.
- Database: La classe de la base de dades és una classe abstracta anotada amb `@Database` que extén `RoomDatabase`. Defineix la llista d'entitats associades i proporciona els DAOs corresponents. També gestiona la creació i migració de l'esquema de la base de dades.

5.4.1.1.- Avantatges del Room

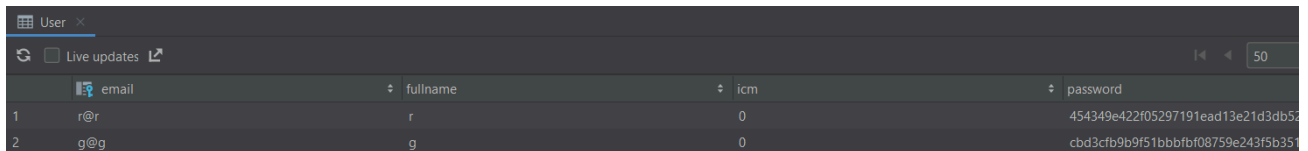
La gran varietat d'avantatges que ens aporta construir la nostra base de dades, amb la biblioteca són moltes[6], unes d'elles són les següents:

- Simplicitat i Reducció de Codi: Room automatitza gran part de la gestió de la base de dades, incloent-hi la generació de codi per a les operacions CRUD (Create, Read, Update, Delete). Això redueix dràsticament la quantitat de codi que he d'escriure i mantenir.
- Verificació en Temps de Compilació: Room verifica les consultes SQL en temps de compilació, la qual cosa ajuda a detectar errors abans que l'aplicació s'executi. Això millora la fiabilitat del codi i redueix el nombre de fallades en temps d'execució.
- Integració amb LiveData i Flow: Room es pot integrar fàcilment amb LiveData i Flow, el que em permet actualitzar la interfície d'usuari automàticament quan les dades canvien. Això facilita la implementació del patró MVVM (Model-View-ViewModel) i millora la reactivitat de l'aplicació.
- Suport per Consultes Complexes: Room suporta consultes SQL complexes, incloent-hi subconsultes i operacions de join (que són unions de taules), permetent-me així gestionar dades relacionades de manera eficient.

5.4.1.2.- Implementació del Room

En el meu projecte, he decidit utilitzar Room per gestionar la base de dades dels usuaris. Aquesta decisió es basa en la necessitat de tenir una gestió robusta, segura i eficient dels usuaris, que són una entitat clau del projecte.

La base de dades d'usuaris, anomenada User, emmagatzema informació crítica com el nom, l'email, la contrasenya i l'IMC (Índex de Massa Corporal). Aquesta informació és essencial per al funcionament de l'aplicació i requereix un maneig acurat i segur.



	email	fullname	icm	password
1	r@r	r	0	454349e422f05297191ead13e21d3db52
2	g@g	g	0	cbd3cfb9b9f51bbbfbf08759e243f5b351

Figura 36.Base de dades Room

5.4.2.- Arquitectura del SQLite

SQLite és una implementació compacta de SQL (que es un llenguatge per fer consultes a bases de dades) que s'emmagatzema com un fitxer únic en el sistema de fitxers. La seva arquitectura és basada en diversos components clau:

- Motor SQL: Interpreta i executa les consultes SQL. Proporciona suport per a la majoria de les funcions estàndard de SQL, com seleccions, insercions, actualitzacions, esborrats, transaccions i joins.
- Motor de Transaccions: Gestiona les transaccions de manera que garanteix l'atomicitat, consistència, aïllament i durabilitat (ACID). Això assegura que les operacions de la base de dades siguin completes i segures, fins i tot en cas d'errors o apagades sobtades.
- Emmagatzematge de Fitxers: Utilitza un fitxer únic per a emmagatzemar totes les dades de la base de dades, incloent les taules, índexs, triggers i vistes. Aquest fitxer pot ser copiat, mogut o compartit fàcilment, i és totalment independent del sistema operatiu.
- API C: SQLite proporciona una interfície de programació d'aplicacions (API) escrita en C que permet integrar la funcionalitat de la base de dades directament en les nostres aplicacions. Això permet una gestió eficient i directa de les dades.

5.4.2.1 Avantatges del SQLite

Hi ha nombrosos avantatges associats amb l'ús de la biblioteca de SQLite per construir la nostra base de dades. Aquests avantatges abasten una àmplia gamma d'àmbits.

Per exemple, SQLite, el sistema de bases de dades que fem servir, és excepcionalment compacte i no requereix cap configuració preliminar ni un servidor separat per al seu funcionament. Això fa que sigui especialment adequat per a aplicacions mòbils i incrustades.

En comparació amb altres sistemes de bases de dades relacionals, configurar i utilitzar SQLite és molt senzill, ja que no cal instal·lar ni administrar un servidor de bases de dades. A més, les bases de dades SQLite s'emmagatzemen com a fitxers individuals que es poden copiar, reubicar i compartir sense esforç entre diverses plataformes i sistemes operatius. Finalment, quan es tracta d'operacions habituals de bases de dades, especialment en entorns amb volums de dades moderats, SQLite ofereix un rendiment excepcional.

5.4.2.2.- Implementació del SQLite

En el meu projecte, he decidit utilitzar SQLite per a diverses bases de dades específiques a més de la base de dades de Room mencionada anteriorment per a la base de dades dels usuaris. D'aquesta manera puc aprofitar els avantatges de Room per gestionar les dades dels usuaris de manera robusta i eficient, mentre que SQLite es fa servir per a altres necessitats de dades més lleugeres i variades.

- **IMC per a Cada Usuari:** Una base de dades per emmagatzemar l'IMC de cada usuari al llarg del temps. Aquesta informació es pot utilitzar per seguir l'evolució de la salut i proporcionar recomanacions personalitzades.

	id	EMAIL	IMC
1	1	g@g	0.043478260869565216
2	2	g@g	0.04914933837429111
3	3	g@g	0.017857142857142856

Figura 37. Base de dades SQLite IMC

- **Notes com a Recordatori:** Una base de dades per emmagatzemar notes de recordatori. Les notes poden incloure informació com el títol, el contingut, i poden ser utilitzades per recordar a l'usuari tasques importants.

	id	email	title	content
1	1	g@g	hola	hola

Figura 38. Base de dades SQLite recordatoris

- **Notes de Cada Dia del Mes:** Una base de dades per emmagatzemar notes específiques per a cada dia del mes. Aquesta funcionalitat permet a l'usuari afegir i consultar notes diàries, ideal per a seguiment d'activitats o dietes.

	id	dia	mes	EMAIL	kcal
1	3	20	enero	g@g	hola
2	5	12	enero	g@g	bb
3	6	11	enero	g@g	hola
4	7	11	enero	g@g	proba1

Figura 39. Base de dades SQLite notes

- Menjar Escanejat amb el seu Nom i Nutrients: Una base de dades per emmagatzemar informació sobre el menjar que l'usuari escaneja, incloent el nom del menjar i els seus nutrients. Això permet a l'usuari mantenir un registre detallat de la seva ingesta alimentària.

	id	email	name	calories	protein	carbs	fats
1	8480000348654	g@g	Almendra natural	N/A	25.0	5.4	51.0
2	8402001001314	g@g	Aceitunas de anchoa	N/A	1.6	0.9	15.0

Figura 40. Base de dades SQLite menja

6.- Implementació

En aquesta secció es detalla la implementació del projecte. A causa de la complexitat de les funcionalitats que l'aplicació ofereix als usuaris, ha estat necessari desenvolupar una gran quantitat de codi.

Per tant, proporcionaré una visió general de les tecnologies i eines utilitzades en la implementació. Això inclou les biblioteques i altres recursos que s'han emprat per desenvolupar l'aplicació. Aquesta visió general permetrà entendre l'arquitectura i les decisions tècniques sense sobrecarregar la documentació amb detalls innecessaris.

6.1.- Estructura de la app

L'estructura i el disseny d'un projecte són factors crítics des del punt de vista dels desenvolupadors. Quan hi ha més d'una persona treballant en un projecte, una estructura inadequada pot complicar el procés de desenvolupament per als altres membres de l'equip.

Per al desenvolupament de l'aplicació he anat posant en diferents carpetes depenent que consisteix cada una de les parts del projecte, ja dins de la base que té predeterminats Android on distribueix de forma estructurada cada una de les parts.

Per a la part de les imatges que he afegit dins el projecte, van totes dirigides dins la carpeta ja existent de "drawable", on allí guardava totes les imatges i els vectors que anava creant i utilitzava dins de l'app

Per la part dels layouts, també tenen una carpeta determinada anomenada "layouts", on allí estan totes les pàgines del front end del projecte, allí és on es guarda tota la part visible de l'aplicació.

Dins de les carpetes creades també tenim la de values, on guarda en una part els colors que podem utilitzar a l'app, d'aquesta manera tots es carreguen a aquesta pàgina i dins del projecte

únicament cal anar cridant-los d'aquí, igual que els string (els noms o textos que apareixen a l'app)

Finalment, tenim la última carpeta on es guarda tota la programació del projecte i totes les interaccions que es duen a terme i totes les funcions que pot fer. Aquí dins aquesta carpeta he creat diferents carpetes per a poder diferenciar les diferents parts del projecte i poder tenir un ordre i poder veure amb més facilitat on està cada una de les parts generades.

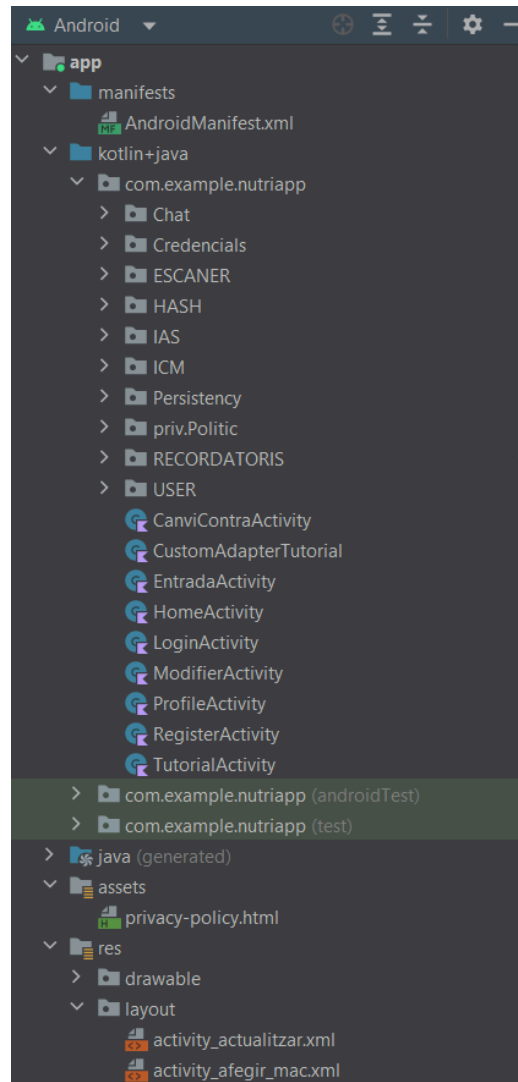


Figura 41.Estructura

Com podem observar a la fotografia anterior, tenim diferents carpetes creades dins la carpeta de l'app. On cada una emmagatzema els codis de la seva part corresponent:

- **USER:** Dintre la carpeta de user, estan les tres classes necessàries per a poder tindre el Room, la primera classe és User on estan els camps que emmagatzema, seguidament està la classe Dao i finalment el repository per utilitzar aquestes dades.

- IAs: Aquí està organitzat tota la part de les connexions amb la base de dades externa i tot el seu procés per emmagatzemar les dades que ens envien.
- Chat: Aquí està la implementació del xat amb la nostra IA i la seva connexió amb el servidor.
- Hash: Està carpeta és la encarregada de guardar el procés de hash de les contrasenyes dels usuaris.
- Persistency: Aquí dins únicament està generada la classe de la DataBase, qui és l'encarregada de generar i crear la nova base de dades on guarda el nou usuaris que es van creant durant el funcionament de l'app
- ICM: Aquí dins estan creades la base de dades de l'IMC, on registra tots els IMC que cada un dels usuaris va afegint a la llista, i la classe on crea els camps de la base de dades.
- Credencials: Aquí hi ha un seguit de classes, que estan destinades a verificar els errors que es poden generar alhora del registre o inici de sessió a l'aplicació, per altra part, les classes que tenen la funció d'interactuar amb el inici de sessió alhora que tingui una bona funcionalitat.
- Notes: Aquí dins estan guardades les classes de les notes que es guarden per emmagatzemar i crear la base de dades dels recordatoris, a part de les funcions d'afegir i actualitzar la llista de dades que hi han.
- Escaner: Dins la carpeta de escàner podem trobar les classes que tenen la funció de poder escanejar i guardar a la base de dades els nous aliments, a part de la base de dades que es crea i els seus camps determinats.
- Calendari: Aquí podem trobar totes les classes que generen el calendari i la selecció dels dies, a part del adapter que genera la llista a cada un dels dies que anem seleccionant, dins aquesta carpeta hi ha una subcarpeta anomenada Mac.
 - MAC: Dins aquesta subcarpeta, únicament tenim la classe que genera els camps de la base de dades i la creació de la base de dades de les notes de les notes per dia

6.2.- Lliberies emprades

Com a la gran varietat de programes, existeixen les lliberies que són diverses implementacions que afegim a l'app perquè ens proporcionin una gran varietat de noves funcionalitats, optimitzacions de codi i simplificacions.

Les llibreries són implementacions de codis que ajuden a fer aquestes implementacions que surten a la nostra app

Les llibreries que he anat afegint a l'app per poder realitzar altres funcions que no venen ja incorporades a Android són:

- Room-compiler: Necessària per a la generació automàtica de codi que Room utilitza per gestionar les bases de dades.

Aquesta llibreria va ser complicada de trobar ja que per la funcionalitat de l'escaneig de codi de barres, hi ha molt poques llibreries implementades on una d'elles és el Zxing, que és la que hem implementat al nostre projecte.

- Zxing-android-embedded: Biblioteca per a l'escaneig de codis de barres i codis QR. Simplifica la integració de la funcionalitat d'escaneig a l'aplicació.

Per a poder fer les connexions a través de les API cap a la base de dades externa he utilitzat les llibreries de:

- Retrofit: Biblioteca per fer totes les connexions amb Internet per la part de les API, que hem afegit a l'app.

A continuació es presenta una breu descripció de les biblioteques més rellevants que s'han implementat durant el projecte, que ja estan incorporades a Android a l'hora de generar un projecte:

Les primeres llibreries tenen la funcionalitat de millorar la compatibilitat del projecte en diferents versions d'Android, així tenim la possibilitat de poder arribar a més gent:

- Androidx.core.ktx: Aquesta biblioteca proporciona extensions de Kotlin per a la biblioteca de compatibilitat d'Android. Facilita la utilització de les API d'Android amb un codi més concís i idiomàtic de Kotlin.
- Androidx.appcompat: Proporciona compatibilitat amb les versions anteriors d'Android i permet utilitzar components d'UI moderns en dispositius antics.

Les següents llibreries ens ajuden en la implementació del projecte a l'hora de fer la navegació entre diferents pàgines:

- Androidx.lifecycle.livedata.ktx: Proporciona una manera de construir components que respectin el cicle de vida de les activitats i fragments, millorant la gestió de dades reactives a través de LiveData.
- Androidx.navigation.fragment.ktx: Aquesta biblioteca facilita la navegació entre fragments, simplificant la gestió de les transicions i mantenint una estructura clara de navegació dins l'aplicació.

- `Androidx.navigation.ui.ktx`: Complementa la biblioteca de navegació, ajudant a la integració amb components de la interfície d'usuari com ara el menú de navegació i la barra d'eines.

Finalment, tenim les llibreries que hem implementat per a la construcció de les bases de dades que hem anat generant al llarg del projecte:

- `Androidx.room.common`: Una biblioteca per gestionar la base de dades SQLite d'una manera més segura i eficient, utilitzant abstraccions i simplificant les operacions amb les bases de dades.
- `Androidx.room.ktx`: Ofereix extensions de Kotlin per Room, millorant la sintaxi i les capacitats idiomàtiques a l'hora de treballar amb bases de dades.

Aquestes biblioteques, més l'altra part de llibreries que no he mencionat han estat essencials per construir una aplicació robusta, eficient i fàcil de mantenir, proporcionant funcionalitats avançades i millorant el desenvolupament de l'aplicació.

6.3.- Servei de escanejament manual

En aquesta part s'explica com es fa la funcionalitat a l'hora de poder escanejar els diferents codis de barres que nosaltres com a usuaris vulguem agregar.

Abans de posar-me amb el funcionament de la càmera i la seva funcionalitat de poder escanejar els diferents codis de barres, vaig estar buscant les diferents llibreries que ajudaven amb la seva implantació. A partir d'aquí vaig trobar que únicament hi existien dos, la ja mencionada Zxing que és la que he utilitzat pel funcionament de l'app i ML Kit que té una gran complexitat a l'hora d'implementar-lo.

Gràcies a la llibreria de Zxing he pogut implementar de manera intuïtiva i sense tindre grans complexitats l'estructura de codi per poder obrir la càmera i poder escanejar tots els codis de barres.

Com podem observar a les fotografies a continuació aquesta ha sigut la implementació que he utilitzat per poder fer aquestes funcions i els permisos que he hagut de donar a l'app per poder permetre un bon funcionament del projecte.

```
<uses-feature
    android:name="android.hardware.camera"
    android:required="false" />

<uses-permission android:name="android.permission.CAMERA" />
```

Figura 42.Permisos càmera

6.4.- Servei de escanejament API

En aquesta part s'explica com es fa la funcionalitat alhora de poder escanejar els diferents codis de barres i poder fer peticions a una base de dades externa per poder saber el nom i els nutrients d'aquell codi de barres que nosaltres volem agregar a la nostra base de dades.

Per la part de la càmera, com he comentat anteriorment he utilitzat la llibreria del Zxing ja que amb la mateixa implementació podia agafar el codi de barres que necessitava.

A partir d'aquí he fet la connexió a través de l'api amb el seu URL corresponent i ha generat la connexió entre la nostra aplicació i la base de dades de food, a partir d'aquí he creat la funció per enviar el nostre codi de barres que hem escanejat i he creat una petició GET per rebre la informació que necessitem sobre aquell aliment i a partir d'aquí poder registrar-lo nosaltres en local a la nostra base de dades.

Per poder utilitzar aquestes funcions he hagut de donar-li permisos a l'app per poder-se connectar amb internet i així poder fer aquestes peticions dels codis de barres.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Figura 43.Permisos internet

6.5.- Seguretat als passwords

A l'app també li he implementat un seguit de funcions per poder protegir les nostres dades, sobre qualsevol atac que puguem rebre de l'exterior.

A l'hora de registrar la nostra contrasenya, he creat una funció per poder posar un hash a la contrasenya i d'aquesta manera registrar-la a la nostra base de dades, encriptada.

Així aconseguim tenir la nostra contrasenya en tot moment encriptada i posant una major dificultat a qualsevol persona que ens vulgui atacar o robar les nostres credencials.

Per poder fer aquesta encriptació he utilitzat la següent funció:

```
Albert *
fun hashPassword(password: String): String {
    //Utilitzem el SHA-256 COM A METODE DE ENCRIPACIO
    val bytes = MessageDigest.getInstance( algorithm: "SHA-256").digest(password.toByteArray())
    return bytes.joinToString( separator: "" ) { "%02x".format(it) }
}
```

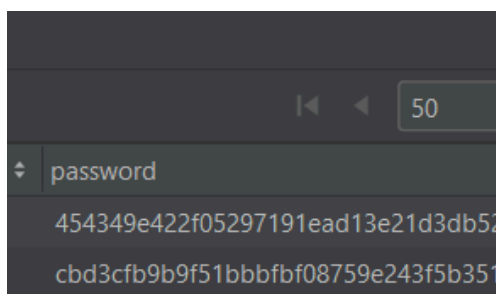


Figura 44 i 45.Hash

6.6.- Servidor

Aquesta implementació no és la part principal del projecte, sinó un pas addicional per fer una prova de conceptes. El nostre objectiu és avaluar la viabilitat de portar la nostra aplicació més enllà, oferint una interfície de xat que permeti als usuaris fer preguntes i resoldre dubtes de manera eficient e interactiva. Aquesta prova de concepte ens ajudarà a identificar millores potencials i a preparar-nos per una implementació més completa i robusta en el futur.

Seguidament us explicaré com he fet la creació d'un servidor en local per poder fer la instal·lació d'un xat, on es podrà comunicar amb la nostra app i així poder generar un xat dinàmic amb els usuaris per poder preguntar aquelles coses que necessita saber o aquells dubtes que tenen.

Mentre que, per altra banda, aixequem un servidor web on s'anirà introduint les preguntes que el nostre xat no sap respondre o no té una resposta molt coherent. Tot això serà essencial per al nostre correcte funcionament de la part de back-end (servidor) i el front-end (app).

- Servidor: Per la part del servidor, he utilitzat el Visual Studio amb la implementació de les llibreries de Python que he descarregat anteriorment per dur a terme aquest projecte.

A partir d'aquí hem generat el nostre projecte amb les instal·lacions corresponent per poder fer tota la part del servidor correctament, que són les llibreries següents:

- flask
- wikipedia
- nltk

A continuació hem fet la implementació del servidor, on hem generat de les dues rutes que utilitzarem per a la nostra pàgina web, una d'elles l'hem anomenat “/chat” i l'altra “/chat_history”, per poder dirigir les preguntes que ens envien els usuaris

Seguidament, hem implementat els ports i la ip per on es connectaran la nostra app i el servidor, per això hem utilitzat el port 5000 i la ip corresponent per al nostre ordinador.

Un cop estan fetes totes aquestes implementacions hem posat en funcionament el servidor per poder fer la connexió ja amb l'app

- APP: Un cop hem implementat tota la part del servidor i l'hem posat en funcionament hem configurat la part de la connexió amb la nostra app i el layout per als usuaris.

Per començar, hem donat els permisos adients per a poder fer la connexió amb API amb el servidor, per això necessitem els mateixos permisos mencionats anteriorment per la connexió amb la base de dades de menja.

A continuació, hem fet la implementació de la interfície de l'usuari per poder-se comunicar amb el nostre xat i així rebre les respostes adients a les seves qüestions, per això hem fet ús de la llibreria de retrofit que hem comentat anteriorment, per poder fer les connexions correctament i així a partir de la ip i el port que hem posat, podem fer la connexió adient entre les dues parts.

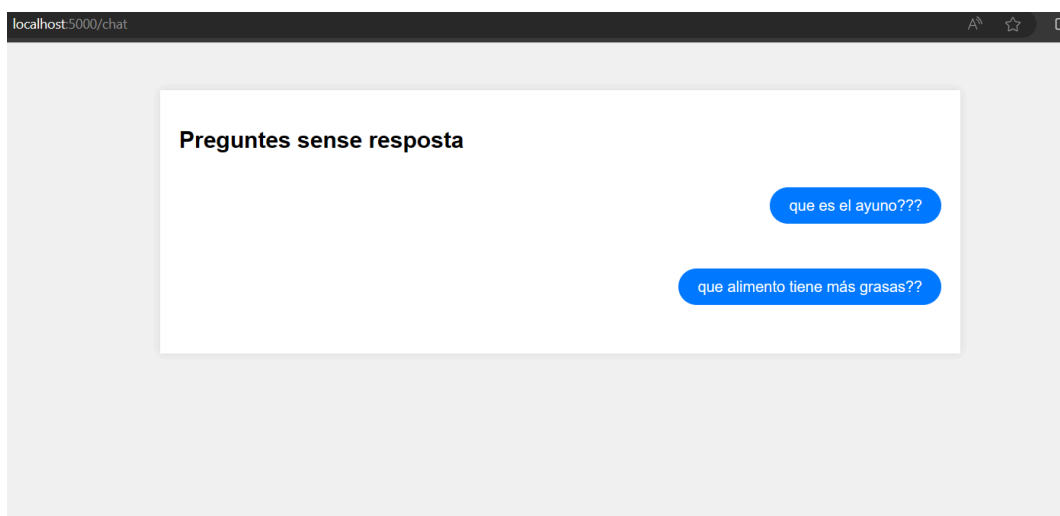
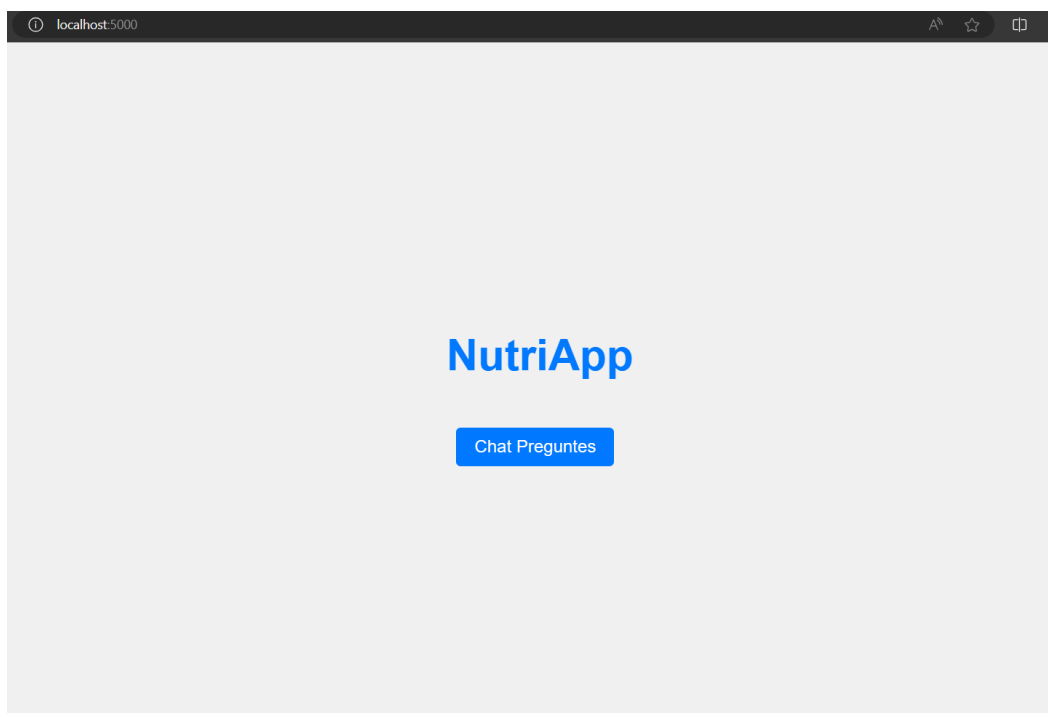


Figura 46 i 47. Servidor web

6.6.1.- Administrador de servidors

A part de les funcions que hem mencionat anteriorment, hem afegit un botó on recull de la nostra app els usuaris que estan registrats, i mostrant-los al nostre servidor, amb la diferència que únicament poden visualitzar aquests usuaris la persona que té les credencials com a administrador.

Per la part de la nostra app, hem generat una connexió amb el servidor al moment d'encendre l'aplicació i càrrega el layout de l'entrada.

En aquest moment fem una connexió igual com hem fet anteriorment amb el xat on connectem a través de la ip i el port la nostra app amb el servidor. Seguidament, fem una crida a la nostra base de dades on demanen tots els usuaris que tenim agregats, a partir d'aquí únicament agafem els emails dels usuaris ja que així al fer la petició no enviem les contrasenyes (així evitem que algú ens pugui entra a la base de dades i robar les credencials de la nostra sessió).

Per l'altra part tenim el nostre servidor, on rep la petició de la nostra aplicació i guarda en una llista tots els usuaris que rep.

Tots els usuaris que arriben es mostren a la nova pàgina que hem afegit amb el botó d'usuaris, amb l'excepció que per poder accedir dins d'aquesta pàgina hem implementat una finestra emergent que demana les credencials corresponent com a administrador per poder accedir-hi.

Per això hem creat unes credencials com administrador perquè únicament pugui accedir-hi l'administrador del servidor en aquesta ocasió sóc jo mateix, i així evitar que algú pugui entrar dins les dades personals dels nostres usuaris.

A l'entrar dins d'aquest endpoint, tenim diferents funcions que podem utilitzar com a administradors que som.

- Cerca: Aquí he afegit la funció per buscar a través del nom que nosaltres posem al buscador, així podem trobar ràpidament aquell usuari que nosaltres volem inspeccionar.
- Mostra tot: Al contrari del botó anterior que és per especificar una cerca únicament, aquí està la funció per poder veure tots els usuaris que hi han actualment creats.
- Ordenar: I finalment tenim el botó de poder ordenar alfabèticament els usuaris de la nostra app, amb la peculiaritat que podem decidir si ordenar-los ascendentment o descendent.

A continuació quan tenim els usuaris que busquem ens mostra una imatge predeterminada a cada un d'ells que he posat per defecte i amb la possibilitat d'entrar a veure el perfil de cada un d'ells personalment.

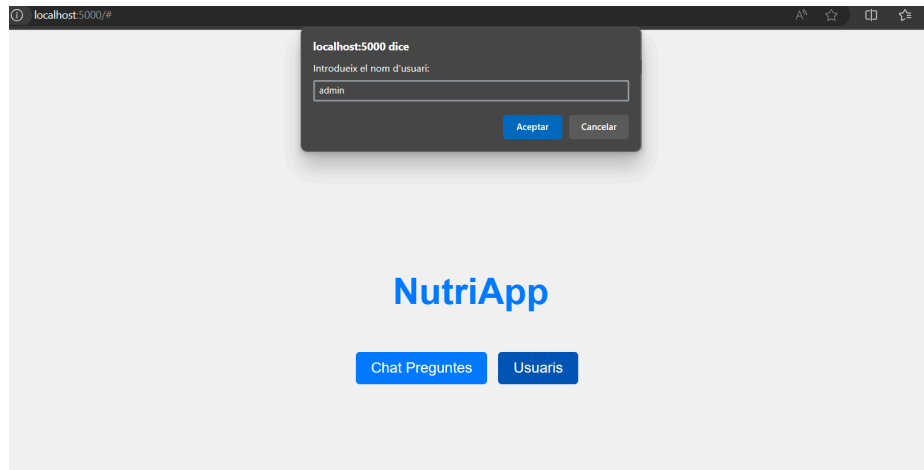


Figura 48.Administrador

Usuaris Registrats

Cerca usuaris... Cerca Mostrar tots

Ordena alfabèticament: Ascendent ▾ Ordena

Tornar a l'inici

Figura 49.Busqueda d'usuaris

Usuaris Registrats

Cerca usuaris... Cerca Mostrar tots

Ordena alfabèticament: Ascendent ▾ Ordena

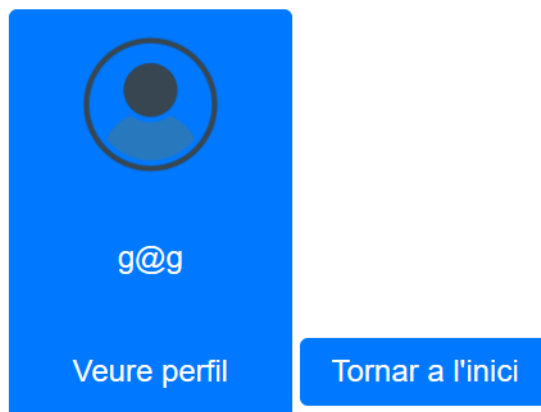


Figura 50.Perfils d'usuaris

6.6.2.- Perfeccionament de la IA del servidor

La implementació que he estat comentant abans, seria una implementació tota en local i el nostre objectiu per a l'app seria fer-la de manera global. Aquesta seria la forma que hauria seguit per poder fer la nova implementació de la IA. Tota aquesta nova implementació proporciona una gran quantitat de respostes molt més precises i rellevants per als nostres usuaris.

A causa de la gran dificultat i sobretot el factor temps per poder realitzar degudament aquesta implementació, no l'he pogut dur a terme com jo hagués volgut, no obstant aquests inconvenients, detallaré el procés que seguiria per poder desenvolupar correctament una IA en el món de l'alimentació i com buscaria les dades reals, per proporcionar informació més precisa.

Primerament, el més important és aconseguir les dades reals i fiables sobre la nutrició. Per això faria una gran recopilació d'informació a través de fonts, com estudis científics, bases de dades nutricionals oficials o publicacions mèdiques. A part d'informació important d'aliments, nutrients i recomanacions dietètiques i altres aspectes relacionats amb la salut alimentària.

A part per poder confirmar que la IA ens dona informació rellevant, seria crucial treballar amb nutricionistes professionals, ja que són experts en el tema i poden ajudar a validar totes les dades que hem anat recopilant i oferir-nos informació addicional.

Totes aquestes eines ens ajudarien a tenir millors coneixements per a la nostra IA a l'hora de donar respostes als nostres usuaris, per això introduiríem totes aquestes dades al nostre model de IA (Open IA) on a través de tècniques com el machine learning (tècniques d'aprenentatge automàtic) entrenaríem a la nostra IA de la següent manera que ara comentarem, per reconèixer patrons i proporcionar respostes molt més concretes als usuaris basats en les dades entregades anteriorment.

Per a l'entrenament de la IA, un cop les dades han estat revisades i perfeccionades pels nostres experts amb el món de l'alimentació, s'han d'estructurar en un format adequat per a l'entrenament. Un dels formats més comuns i compatibles amb les eines d'OpenAI és el JSONL.

JSONL és un format de fitxer on cada línia conté un objecte JSON separat. Aquest format és molt útil per a l'entrenament de models d'aprenentatge automàtic perquè permet processar grans conjunts de dades de manera eficient, amb la particularitat que cada una de les línies poden ser diferents.

Després de preparar el conjunt de dades, el següent pas és carregar aquestes dades al sistema d'OpenAI utilitzant la seva API de Fine-Tuning. Aquesta API permet als usuaris ajustar els models preentrenats d'OpenAI amb les seves pròpies dades, per això cal utilitzar les eines proporcionades per OpenAI per enviar els fitxers JSONL al seu sistema.

Un cop carregades, és necessari configurar els paràmetres d'entrenament per adaptar el procés a les necessitats específiques del projecte. Alguns dels paràmetres més importants inclouen el nombre de mida del batch[8].

Mida del batch: Es refereix a la quantitat de dades processades en cada pas d'entrenament. Un batch més petit pot fer que l'entrenament sigui més precís, però també pot augmentar el temps necessari per completar l'entrenament.

Un cop configurats aquests paràmetres, es pot iniciar l'entrenament del model. Durant aquest procés, el model aprèn a partir de les dades proporcionades, ajustant els seus paràmetres interns per millorar la precisió de les respostes.

Aquest procés d'aprenentatge implica una sèrie d'iteracions on el model fa prediccions, les compara amb les dades reals i ajusta els seus paràmetres per reduir l'error entre les prediccions i les respostes correctes.

Pel procés d'entrenament de models personalitzats amb OpenAI, tenen uns grans documents on especifiquen molt bé tota l'informació necessària per a tota aquesta implementació[8][9].

Finalment, amb el model d'IA entrenat, el pas següent és implementar-lo en l'aplicació.

Per això, utilitzarem la OpenIA que és un dels models de IA més implementats que hi ha actualment, ja que ofereix molts models de llenguatges avançats, com ara els actuals GPT-3 o 4, que són uns models de comprensió i generació de textos d'un alt nivell, cosa que ens ajudaria a poder desenvolupar a gran escala la nostra IA de NPL, millorant la qualitat de les respostes i oferint una experiència d'usuari superior.

Per a poder fer aquesta implementació necessitarem crear un compte a la OpenIA per a poder tenir accés a les seves claus d'API per poder connectar la nostra app amb la IA, així permetent-nos poder enviar i rebre les respostes que vulguin els nostres usuaris.

A partir d'aquí faríem una configuració semblant amb el flask, al Visual Studio on canviarem i integrarem la nostra nova API, configurant les noves rutes per enviar les preguntes dels usuaris al GPT i que en retorni les respostes. Flask és un framework lleuger que ens permetrà crear un servidor back-end eficient i gestionar les comunicacions entre la nostra aplicació i la IA.

Per a fer aquesta nova integració, haurem d'instal·lar les noves llibreries que necessitem com són el request per poder rebre les peticions a través de HTTP i amb la creació d'un entorn virtual per poder tenir més privacitat a les nostres claus d'API, L'entorn virtual ens ajudarà a gestionar les dependències del projecte de manera aïllada, la seguretat de les nostres credencials.

A partir d'aquí afegiríem les rutes per a poder connectar el servidor i la IA, amb els mètodes de l'API, on afegiria un mètode POST per poder rebre les preguntes dels usuaris i poder enviar-les a la IA i rebre la seva resposta i seguidament d'un GET per permetre'ns tenir el registre aquelles preguntes que no han tingut una resposta molt coherent de l'Open IA.

Aquesta estructura ens permetrà gestionar eficaçment les interaccions entre els usuaris i la nostra aplicació.

Finalment, farem la implementació de les connexions a dins la nostra aplicació, per això hem de tornar a fer bé la nostra configuració amb el servidor de flask que hem generat, per això

continuarem necessitant les llibreries que hem instal·lat ja al nostre projecte i configurarem la nostra classe per a distribuir de manera adient les preguntes cap al servidor i mostra les respostes que li retorna.

Aquesta transició d'una implementació en local a global ens permetria a la nostra aplicació fer un pas crucial per millorar l'experiència dels usuaris i així poder expandir l'accessibilitat a molta més població.

Crear una solució global amb OpenAI implica diversos passos crítics. En primer lloc, cal crear un compte a la plataforma d'OpenAI per obtenir les claus d'API necessàries per connectar-nos als seus serveis. Aquesta etapa implica un registre senzill, però vital, ja que aquestes claus d'API són l'accés clau per a la interacció amb els models d'IA.

Amb les claus d'API en mà, el següent pas seria configurar la infraestructura de la nostra aplicació per connectar-se amb els serveis d'OpenAI. Això implica la implementació de noves rutes i controladors al nostre back-end, que serà realitzat a través del framework Flask

Un aspecte crític d'aquesta implementació és la seguretat de les claus d'API OpenAI. Per garantir la confidencialitat i la seguretat de les dades, és crucial crear un entorn virtual per al nostre projecte. Això ens permetrà mantenir aïllades les dependències del projecte i protegir les nostres claus d'API de qualsevol accés no autoritzat.

Un cop configurada l'arquitectura del back-end, hem de gestionar les interaccions entre la nostra aplicació i els serveis d'IA d'OpenAI. Això inclou la de mètodes d'API que permetran enviar preguntes dels usuaris als models d'IA i rebre respostes pertinents.

Tot això seria la implementació que seguiria per a poder fer una implementació d'una IA amb una millora a la quantitat de respostes molt més rellevants i precises perquè els nostres usuaris. Aquesta millora ens permetrà oferir un servei de major qualitat assegurar que els nostres usuaris rebin respostes adequades a les seves necessitats, incrementant així la satisfacció i l'eficàcia de la nostra aplicació.

7.- Proves d'ús de l'app

És primordial la realització d'un bon procés de validació i avaluació en la realització d'un projecte software. En el projecte actual s'ha realitzat el joc de proves, avaluarem les funcions principals de l'aplicació, ja que la principal tasca és assegurar que l'usuari pugui fer servir totes les funcions implementades en l'aplicació sense cap problema, i que totes elles funcionin correctament i compleixin amb tots els requisits.

Així també vigilant tots aquells casos que poden causar algun problema o generar algun tipus d'error dins l'app

A part de la implementació de tots els casos que ens podem trobar, com des de la creació del mateix usuari o l'intent de guardar el mateix codi de barres a la nostra base de dades.

Totes aquestes avaluacions han sigut afegides a través de log, on nosaltres com a programadors podem veure a la consola de l'ordinador els missatges que nosaltres anem comentant, segons les funcions que fem a l'app.

7.1.- Registre de l'usuari

Començarem des de la creació d'un nou usuari, on podem observar, com hem anat mencionant anteriorment que tenim els tres camps que s'han d'emplenar obligatòriament per poder tenir un usuari a la nostra app.

Primer de tot abans de poder guardar el nostre usuari, es verifica que estiguin els tres camps omplerts, ja que si no et deixarà crear-lo, d'aquesta manera bloquejarà el botó de registrar i no et deixarà continuar avançant.

Una vegada ja complets els camps, podem prémer el botó de guardar on farà una verificació de si l'email que has introduït ja existeix o no. Si ja existeix mostrarà per pantalla un toast on ens dirà que aquest usuari no es pot registrar, ja que ja té un compte registrat amb aquest email.

Casos de Registre de l'usuari	Comprovació
Verificació camps relleats	Comprovació feta, ja que si no, no deixa pressionar el botó de guardar
Verificació si ja existeix el compte	Comprovació feta, perquè si no et surt un missatge d'usuari creat ja

7.2.- Login de l'usuari

A continuació anem a la part del inici de sessió, aquí és on és s'inicia la sessió dins la nostra app, sempre que estigui ja creada el compte de l'usuari.

Al començament es fa una verificació dels dos camps que demana, que són l'email i la contrasenya, d'on mira si estan els dos camps introduïts o no, així bloquejant el botó d'entrada fins que no estan els dos camps emplenats.

Seguidament, quan els dos camps estan introduïts, es fa una validació de l'usuari, mirant si ja està registrat aquest email o no. Per així poder donar accés a l'usuari o enviar un missatge avisant-lo que no existeix aquest usuari.

Casos de Login de l'usuari	Comprovació
Verificació camps relleats	Comprovació feta, ja que si no no deixa pressionar el botó d'entrada

Verificació si ja existeix el compte	Comprovació feta, perquè si no et surt un missatge d'usuari creat ja
--------------------------------------	--

7.3.- Canvi de contrasenya

Amb aquesta pàgina podrem canviar la contrasenya de la teva sessió, per si has tingut algun problema.

Per això surt una casella de verificació del primer email, aquí verifica si aquest email existeix o no, si no existeix et surt un missatge avisant-te que no estas registrat, i per tant, no et deixa fer res si no introdueixes bé l'email o tornes a la pàgina d'inici de sessió.

Si l'email ha sigut correcte llavors t'apareixen els dos camps per introduir la nova contrasenya dues vegades, aquí verifica si has emplenat els dos camps i si son igual, cosa que si no és així, i per tant no compleix i no et deixa prémer el botó de guardar.

Casos de canvi de contrasenya de l'usuari	Comprovació
Verificació que el email existeix	Comprovació feta, sinó envia un missatge d'usuari no registrat
Verificació camps rellenats	Comprovació feta, ja que si no no deixa pressionar el botó d'entrar
Verificació si coincideixen contrasenyes	Comprovació feta, perquè si no no deixa pressionar el botó d'entrar

7.4.- Home

Aquesta pàgina s'encarrega de poder trobar a totes les possibilitats que hi ha dins la nostra app.

Aquí és on es fa la distribució de tota l'app, per poder dirigir-se a totes les funcionalitats que hi han incloses, per aquesta raó és molt important els listeners i botons que estan configurats funcionin correctament.

Casos Home	Comprovació
Verificació que el email existeix	Comprovació feta, sinó envia un missatge d'usuari no registrat
Verificació que els listeners de les imatges funcionen correctament	Comprovació feta, ja que cada imatge et dirigeix correctament a cada una de les parts del projecte

Verificació que els botons funcionin correctament	Comprovació feta, perquè cada botó et dirigeix correctament a cada una de les parts del projecte.
---	---

7.5.- Càlcul del IMC

Aquesta secció s'encarrega de guardar tots els càlculs que vas realitzant durant l'ús de l'app

Cada vegada que l'usuari calcula el seu nou Índex de Massa Corporal, s'afegeix a una base de dades que et porta a un registre i té la funció de mostrar-les en un layout de forma cronològica per poder anar veient el seu progrés.

Per això hem verificat que tingui una bona càrrega de dades per a poder mostrar-les i que no es perdi cap càlcul, per això hem seguit aquestes verificacions.

Casos de agregar els nous IMC	Comprovació
Verificació que l'usuari existeix	Comprovació feta, sinó no et deixa entrar a la pàgina
Verificació camps rellenats	Comprovació feta, ja que si no no deixa pressionar el botó de calcular
Verificació si esta guardat internament	Comprovació feta, perquè a la consola ens mostra un missatge indicant-nos que està ben guardat
Verificació de les mostres guardades per poder visualitzar	Abans de carregar la pàgina es pot veure un registra a la consola conforme està guardat tot als nous camps
Verificació del us alhora de rebre les dades corresponents	Comprovació feta, si no rep bé les dades de la base tenim un excepció creada per notificar-ho.

7.6.- Verificació Recordatoris

Aquesta secció té la funció de guardar tots aquells recordatoris que l'usuari vulgui registrar per recordar-se més endavant.

A part té la funció de poder eliminar o editar tots aquells recordatoris introduïts la qual comporta poder trobar un gran seguit de verificacions per evitar qualsevol problema alhora quan un usuari faci ús d'ella.

Per això hem seguit uns passos determinants per poder veure que funciona correctament la nostra base de dades i els layouts relacionats a aquesta funcionalitat.

Casos de agregar els nous Registres	Comprovació
Verificació que hi ha un usuari actiu	Comprovació feta, sinó no et deixa entrar a la pàgina
Verificació càrrega de de la base de dades	Comprovació feta, ja que si no tenim un error que ens indica que tenim un error, tot això a la consola
Verificació si esta guardat internament el nou recordatori	Comprovació feta, perquè a la consola ens mostra un missatge indicant-nos que està ben guardat
Verificació de les mostres guardades per poder visualitzar	Al carregar la pàgina es poden veure un registra a la consola conforme està guardat tots els nous camps i es visualitzen correctament
Verificació alhora d'actualitzar les dades	Comprovació feta, perquè mostra un toast indicant que ha estat modificada correctament
Verificació alhora d'eliminar les dades	Comprovació feta, perquè mostra un toast indicant que ha estat eliminat correctament.

7.7.- Verificació Notes

Aquesta secció té la funció de guardar tots aquelles notes independents per a cada un dels dies del any on l'usuari vulgui registrar per recordar-se més endavant.

A part te la funcio de poder eliminar o editar tots aquells recordatoris que estan posats, cosa que comporta un gran seguit de verificacions per evitar qualsevol problema alhora quan un usuari faci ús d'ella, amb la peculiaritat de que cada nota va emmagatzemada amb el seu dia i mes corresponent.

Per això hem seguit un passos determinants per poder veure que funciona correctament la nostra base de dades i els layouts relacionats a esta funcionalitat.

Casos de agregar els nous Registres	Comprovació
Verificació que hi ha un usuari actiu	Comprovació feta, sinó no et deixa entrar a la pàgina

Verificació càrrega de de la base de dades	Comprovació feta, ja que sinó tenim un error que ens indica alguna cosa que no funciona, tot això a la consola
Verificació si esta guardat internament el nou recordatori	Comprovació feta, ja que a la consola ens mostra un missatge indicant-nos que esta ben guardat
Verificació del dia	Comprovació feta, ja que envia un toast al dia que entres
Verificació del mes	Comprovació feta, ja que envia un toast al mes que entres igual que amb el dia
Verificació de les mostres guardades per poder visualitzar	Al carregar la pàgina es pot veure un registra a la consola conforme esta guardat tot els nous camps i es visualitzen correctament
Verificació alhora d'actualitzar les dades	Comprovació feta, ja que mostra un toast indicant que ha estat modificada correctament
Verificació alhora d'eliminar les dades	Comprovació feta, ja que mostra un toast indicant que ha estat eliminat correctament

7.8.- Escaner Manual

Aquesta secció té la funció de poder escanejar els codis de barres dels aliments que nosaltres vulguem amb la diferència que els nutrients i el nom de l'aliment l'introduïm nosaltres manualment.

D'aquesta manera tenim la possibilitat de poder guardar aquelles receptes o aliments que nosaltres fem, d'una manera més ràpida i eficient a la nostra base de dades d'aliments que proporciona l'app.

Per a poder fer ús d'aquesta funcionalitat, hem fet uns seguits de verificacions per poder veure que realment tots els processos funcionen correctament i fan el que està implementat.

Casos de agregar els nous aliments manualment	Comprovació
Verificació que hi ha un usuari actiu	Comprovació feta, sino no et deixa entrar a la pàgina
Verificació de introducció de tots els	Si els noms i els nutrients no estan

elements (nutrients i nom)	introduïts correctament, no et deixa escanejar
Verificació camara	Es comprova internament que la càmera s'activi correctament sense cap problema a la consola
Verificació sollicitud camara	Es comprova inicialment els permisos de l'aplicació per l'ús de la càmera amb l'acceptació de l'usuari
Verificació si esta guardat internament el nou recordatori	Comprovació feta, ja que a la consola ens mostra un missatge indicant-nos que esta ben guardat
Comprovació en codis de barres	S'han provat diferents codis de barres per veure si agafa correctament el codi
Proves de llum	S'ha provat que la càmera agafi correctament els codis de barres amb diferents llums i angles

7.9.- Escaner API

Aquesta secció té la funció de poder escanejar els codis de barres dels aliments que nosaltres vulguem amb la diferència que la nostra app ens connecta amb una base de dades externa i ens introdueix ell directament el nom i nutrients del que hem escanejat.

D'aquesta manera tenim la possibilitat de poder guardar aquells aliments que escanegem sense la necessitat d'introduir cap dada d'una manera més ràpida i eficient a la nostra base de dades d'aliments que proporciona l'app.

Per a poder fer ús d'aquesta funcionalitat, hem fet uns seguits de verificacions per poder veure que realment tots els processos funcionen correctament i fan el que està implementat.

Casos de agregar els nous aliments manualment	Comprovació
Verificació que hi ha un usuari actiu	Comprovació feta, sino no et deixa entrar a la pàgina
Verificació connectivitat Base de dades	Es comprova inicialment si es connecta correctament a través de l'URL a la base de dades

Verificació mètodes API	Es comprova que el mètode que enviem funcioni correctament i que agafi l'aliment de manera correcta
Verificacions d'aliments que no existeixen	Aquells aliments que no estan a la base de dades d'Internet, et mostra un error i et notifica que no existeix aquest aliment
Verificació camara	Es comprova internament que la càmera s'activi correctament sense cap problema a la consola
Verificació sollicitud camara	Es comprova inicialment els permisos de l'aplicació per l'ús de la càmera amb l'acceptació de l'usuari
Verificació si esta guardat internament el nou recordatori	Comprovació feta, ja que a la consola ens mostra un missatge indicant-nos que esta ben guardat
Comprovació en codis de barres	S'han provat diferents codis de barres per veure si agafa correctament el codi
Proves de llum	S'ha provat que la càmera agafi correctament els codis de barres amb diferents llums i angles.

7.10.- Verificació del RecyclerView (Tutorial)

En aquesta pàgina s'encarrega de mostrar un recyclerView on ensenya en una diapositiva diferent cada una de les funcions que té la nostra app amb una explicació.

Aquí és on es fa un petit resum o com nosaltres l'hem anomenat tutorial, on et va mostrant cada una de les parts que té el nostre projecte i és important verificar que ensenya correctament aquestes vistes.

Casos Tutorial	Comprovació
Verificació que l'usuari existeix	Comprovació feta, sinó envia un missatge d'usuari no registrat
Verificació que cada diapositiva es mostra i tens la possibilitat de passar a la següent	Comprovació feta, ja que es pot passar fàcilment i de manera intuïtiva a cada una

	de les diapositives
Verificació que els textos i imatges del adapter	Comprovació feta, perquè totes les diapositives tenen bona estructuració i mostren correctament la informació.

7.11.- Xat

Aquesta secció té la funció de poder interactuar amb el nostre xat que hem implementat a un servidor web on a través de API ens podem comunicar.

D'aquesta manera tenim la possibilitat de poder tindre unes respostes més ràpides sobre aquelles preguntes que els usuaris de l'app estan interessats en saber.

Per a poder fer ús d'aquesta funcionalitat, hem fet uns seguits de verificacions per poder veure que realment tots els processos funcionen correctament i fan el que està implementat.

Casos de comunicació amb el chat a través del servidor	Comprovació
Verificació que hi ha un usuari actiu	Comprovació feta, sinó no et deixa entrar a la pàgina
Verificació de introducció de la pregunta	Si no introdueixes cap dubte no et deixa enviar res cap al xat
Verificació de connexió	Es fa una comprovació dins l'app a través de la consola per si està ben connectat l'app amb el servidor
Verificació dels missatges	Es verifica els missatges, depenent el que arriba amb un toast amb l'error sino et mostra la resposta correcta a la pantalla
Verificació dels mètodes	Es verifica a la consola del servidor que les peticions que arriben són correctes amb els codis corresponents "200"
Verificació funcionament del servidor	Si el servidor no es posa en funcionament es mostrarà a la consola del servidor
Verificació respostes chat	Si rep una pregunta que no sap la resposta envia una llista amb les dubtes que sap respondre
Verificacions de preguntes que no sap respondre el chat	Aquelles preguntes que no sap respondre el xat, es guarden i es posen al servidor per

	poder veure les i afegir-les més endavant
--	---

7.12.- Comprovació finals dels botons

Finalment i una part molt important a l'hora de comprovar si la nostra app funciona tot correctament és fer una revisió de totes les implementacions que has afegit al projecte, per poder tornar a la pàgina anterior, els botons de menú per poder tornar al layout del Home.

Tots aquests petits detalls és molt importants que estiguin controlats i verificats que funcionin correctament per un bon desenvolupament de la nostra app.

Casos Botons i Listeners	Comprovació
Verificació que hi ha un usuari actiu	Comprovació feta, sinó no et deixa entrar a la pàgina
Verificació que cada diapositiva es mostra tots els botons que estan implementats	Comprovació feta, ja que es veuen tots correctament
Verificació que tots els botons dirigeixen correctament a la pàgina corresponent	Comprovació feta, ja que tots tenen la seva funcionalitat ben implementada

7.13.- Servidor

Per altra part, tenim també el servidor, on hem creat una pàgina per als administradors de l'app, on podem veure els perfils de tots els usuaris que es van crear.

Per poder accedir a aquesta pàgina és necessària les credencials d'administrador ja que sinó no pots tenir accés a ella.

Per poder verificar tots aquests passos que hem realitzat hem seguit un seguit de proves per comprovar que tot funciona correctament.

Servidor	Comprovació
Verificació de connexió	Es fa una comprovació dins l'app a través de la consola per si està ben connectat l'app amb el servidor
Verificació de les credencials de administrador	Comprovació feta, ja que sinó no et deixa accedir a la pàgina d'usuaris
Verificació de usuaris creats	Està comprovat a través de la base de dades

	que agafa correctament tots els usuaris
Verificació de usuaris inexistents al buscador	Si es busca un usuari que no existeix, ja està implementat que no faci res la pàgina
Verificació de l'ordre	Està comprovat que l'ordre és el correcte a l'ordenar els usuaris
Verificació de que mostri tots els usuaris creats	El botó ensenya tots els usuaris que es poden observar
Comprovació que es pot veure el perfil de cada usuari	Cada usuari en fer-li clic et dirigeix al seu perfil correctament.

8.- Conclusions

Després de tota la implementació d'aquest projecte es crea una aplicació nativa per a la plataforma d'Android, que ens permet als usuaris registrar-nos i tindre un servei d'alimentació a mà durant el nostre dia a dia. On l'usuari disposa d'una aplicació mòbil que li podrà permetre realitzar diferents accions com podrien ser calcular el seu IMC, o tindre un bloc de recordatoris o de notes sobre les macros diaris, la possibilitat de poder escanejar els aliments que vulguis i guardar-los a una base de dades, la possibilitat de poder interactuar amb el nostre xat per preguntar aquells dubtes que et sorgeixin.

Amb aquesta aplicació també s'està implementant una part per als administradors, cosa que permet tenir un seguiment dels usuaris que es van creant.

Considero que la implementació d'aquesta aplicació pot ajudar molt a totes aquelles persones amb ganes de tenir uns hàbits alimentaris més saludables o aconseguir algun objectiu que necessitin un recompte de les menjades.

Al principi d'aquest projecte la idea principal únicament era la construcció d'un programa on poder guardar tots els aliments que anem consumint al dia a dia amb els seus nutrients corresponents.

Però després de pensar bastant la idea i de diverses interaccions i la necessitat personal d'afegir altres ampliacions, vaig anar millorant el disseny i implementant noves incorporacions a l'aplicació. La maduració del projecte ha sigut reiterativa, des del disseny dels primers layouts amb la seva lògica poc estructurada fins a la implementació d'un seguit de consultes i funcions complexes en base de dades i servidors externs.

Prèviament, aquest treball no havia abordat grans projectes de forma independent, ja que durant els cursos que he anat realitzant durant els meus estudis a la universitat sempre hem treballat en equips on sempre es dividien més les tasques a fer.

Per altra banda, tenint en compte que el treball en equip és un valor molt important dins la categoria d'Enginyer i de qualsevol altra evidentment, volia desafiar-me a mi mateix i dur a terme un gran projecte individual i així poder veure tots els coneixements que he anat adquirint cursant el grau d'enginyeria de Sistemes i Serveis de Telecomunicacions, i especialitzant-me en la part telemàtica, i així poder comprovar si tinc suficients coneixements i habilitats, anant més enllà de la programació i assumint el rol general de tot el treball.

Durant el desenvolupament d'aquest projecte, he anat adquirint nous coneixements de programació, ja que hi ha hagut parts del treball, que em sortien errors inesperats, i he sigut capaç de poder solucionar-los ràpidament, tot i que també he de dir que alguns altres errors m'ha costat una mica més esbrinar-los, però els he pogut arreglar dedicant una mica més de temps, tot i això, sempre he trobat la solució d'aquests petits problemes que m'han sorgit i, per tant, personalment he de dir-vos que m'he sentit motivat.

Al final del projecte, totes aquestes dificultats que m'havien anat sorgint, han desaparegut, ja que els he resolt i he pogut fer una implementació corresponent als objectius finals que tenia marcats, per tant, he aconseguit tenir uns millors coneixements i aprenentatges de les aplicacions Android i la creació i connexió a través d'API dels servidors web.

He de dir-vos que tal com anava realitzant el projecte m'anaven sorgint moltes idees noves, fins i tot, algunes que es podrien implementar en futures millores de l'app, com per exemple el fet de poder donar permisos a l'administrador, perquè pugui eliminar i editar els usuaris que hi ha a la nostra app. En un futur m'agradaria afegir una pàgina web més dedicada a l'explicació de la nostra app i així poder obrir-nos a la possibilitat que ens conegui molta més gent.

La idea és que puguem aconseguir que tothom que vulgui s'animi a poder seguir uns hàbits saludables i de manera fàcil i ràpida, sense la necessitat de tenir coneixements molt alts sobre el tema en qüestió, i per tant, aquesta eina els hi sigui de gran ajuda.

9.- Annexes

Per a la instal·lació del nostre projecte, és necessari seguir uns passos concrets per assegurar que tot funcioni correctament, tant a la part d'Android com a la part del servidor.

Instal·lació de Android

Per la part d'Android, el procés és força senzill, ja que totes les llibreries i implementacions ja estan incloses en l'aplicació.

Els passos que s'han de seguir són els següents:

Importar el Projecte: Cal importar el projecte Android al vostre entorn de desenvolupament.

Verificar Dependències: Tot i que totes les llibreries ja estan incloses, és recomanable revisar les dependències al fitxer `build.gradle` per assegurar-se que tot està correctament configurat, així comparant les llibreries mencionades anteriorment al apartat (6.2)

Instal·lació de Visual Studio

La part del servidor requereix alguns passos addicionals per instal·lar les llibreries necessàries i configurar correctament les connexions.

Els passos que s'han de seguir són els següents:

Importar el Projecte del Servidor: Cal importar el projecte del servidor al vostre entorn de desenvolupament (com podria ser Visual Studio, que és el que hem implementat nosaltres).

Si prefereixes pots crear un entorn virtual per generar el servidor del projecte, per la seva creació cal aïllar les dependències del projecte.

Instal·lar les Llibreries Necessàries: Un cop activat el projecte dins o a nou entorn virtual, instal·leu les llibreries necessàries, que són les següents

Les comandes són iguals per les tres llibreries que s'implementen:

- `Pip install flask`
- `Pip install ntlk`
- `Pip install wikipedia`

Finalment, per les dues parts del projecte, és molt important obtenir la IP del Vostre Ordinador, per poder configurar la IP del vostre ordinador correctament dins les dos parts del projecte així assegurant la correcta connexió entre les dues interfícies (l'aplicació Android i el servidor)

Referències

Webgrafies:

[1] Native, Hybrid or Web

<https://copperdigital.com/blog/which-is-the-best-app-development-approach-native-hybrid-or-pwas/>

[2] Android Arquitectura

<https://crhystamil.medium.com/android-arquitectura-de-android-f6618e6ae969>

[3] Planificació Monday

[monday - Home](#)

[4] Zxing

[Zxing - Leer QR y códigos de barras en Kotlin - Curso Kotlin Para ANDROID](#)

[5] Retrofit

[Retrofit \(gerardfp.github.io\)](#)

[6] Room

[Room, la librería de Base de datos de Android - DevExpert](#)

[7] Open IA

[OpenAI](#)

[8] Open IA API

[Fine-tuning - OpenAI API](#)

[9] Open IA API

[API Reference - OpenAI API](#)

Fotografies:

- Figura 1.** Planificació del Projecte
- Figura 2.** Principi del projecte
- Figura 3.** Implementació del projecte
- Figura 4.** Implementació de les proves
- Figura 5.** Documentació
- Figura 6.** Resolució Final
- Figura 7.** Aplicacions Natives
- Figura 8.** Aplicacions web
- Figura 9.** Aplicacions híbrides
- Figura 10.** Aplicacions
- Figura 11.** Capes de sistema Android
- Figura 12.** Capes de sistema iOS
- Figura 13.** Entrada
- Figura 14 i 15.** Login i register
- Figura 16 i 17.** Canvi de contrasenya
- Figura 18.** Home
- Figura 19.** Perfil
- Figura 20.** Modificacions del perfil
- Figura 21 i 22.** Recordatoris i Afegir recordatoris
- Figura 23.** Calendari
- Figura 24 i 25.** Notes i Afegir notes
- Figura 26.** IMC càlcul
- Figura 27.** IMC
- Figura 28.** IMC solucions
- Figura 28 i 29.** Elecció i permisos camara
- Figura 30 i 31.** Escaner manual i API
- Figura 32.** Llista menjars
- Figura 33.** Tutorial
- Figura 34.** Política de privacitat
- Figura 35.** Xat
- Figura 36.** Base de dades Room
- Figura 37.** Base de dades SQLite IMC
- Figura 38.** Base de dades SQLite recordatoris
- Figura 39.** Base de dades SQLite notes
- Figura 40.** Base de dades SQLite menja
- Figura 41.** Estructura
- Figura 42.** Permisos càmera
- Figura 43.** Permisos internet
- Figura 44 i 45.** Hash
- Figura 46 i 47.** Servidor web
- Figura 48.** Administrador
- Figura 49.** Busqueda d'usuaris
- Figura 50.** Perfils d'usuaris

Diagrams:

Diagrama 1.Estructura App

Diagrama 2.Entitats