

Gerard Serra Rodríguez

Desenvolupament d'una aplicació iOS per el món del fitness

“FitYou”

TREBALL DE FI DE GRAU

dirigit per Tomàs González Sánchez

Grau d'Enginyeria en Sistemes i Serveis de Telecomunicacions



**Tarragona
2023**

Resum.

Darrerament hi ha més gent que comença a cuidar més intensament la seva salut i el seu aspecte físic, a realitzar alguna mena d'esport o exercici físic de manera regular. Comença a ser habitual que molta gent vagi al gimnàs o d'altra comencin a desenvolupar alguna activitat relacionada amb el món del "fitness" com pot ser running, CrossFit, etc. Així doncs, el desenvolupament d'una aplicació enfocada per un món com aquest pot esdevenir una eina fonamental en la rutina diària de qualsevol persona aficionada al fitness, per tal de poder tenir una planificació perfecta de tots els seus entrenaments. Aquest projecte consisteix en la disseny i la creació d'una aplicació per dispositius amb un sistema operatiu iOS. L'objectiu principal es dotar a l'usuari de mitjans a través de l'aplicació desenvolupada per poder emmagatzemar, afegir o eliminar totes les seves rutines d'esport. Tanmateix l'usuari podrà rebre exercicis proposats per la mateixa aplicació sense necessitat de tenir rutines d'exercici pròpies. Els exercicis comptaran amb informació i explicació de l'objectiu de l'exercici per guiar a l'usuari i ajudar a triar el que realment busca o necessita. També aquesta aplicació compta amb un espai on l'usuari podrà aportar dades fisiològiques, entre d'altres el pes o l'alçada, per poder calcular certs valors molt coneguts a l'entorn fitness. L'aplicació també comptarà amb un espai on trobar informació nutricional. D'aquesta manera l'aplicació pot ser incorporada a la vida diària d'una persona esportista, per tal que aquesta tingui un seguiment en tots els seus processos diaris, un control detallat i exhaustiu de les seves sessions d'esport, i tenir una eina a la seva mà que l'acompanyi en tots els progressos en el món del fitness, a part de comunicar-se amb altres usuaris registrats a la mateixa aplicació per tal de poder compartir qualsevol informació que considerin rellevant dins d'un xat global. L'aplicació també incorpora un mapa on els usuaris tindran a la seva disposició els gimnasos que tenen a propers i tindran a la seva disposició plans, relacionats amb entrenaments o nutrició, per poder comprar si l'usuari veu que s'ajusta a les seves necessitats.

Resumen.

Últimamente, hay más gente que empieza a cuidar con más rigurosidad su salud y su aspecto físico, y a realizar algún tipo de deporte o ejercicio físico de forma regular. Empieza a ser habitual que mucha gente vaya al gimnasio o incluso empiecen a desarrollar alguna actividad relacionada con el mundo del "fitness" como puede ser running, CrossFit, etc. Así pues, el desarrollo de una aplicación enfocada a una sociedad como esta puede convertirse en una herramienta fundamental en la rutina diaria de cualquier persona aficionada al fitness, y conseguir con ello una planificación perfecta de todos sus entrenamientos. Este proyecto consiste en el diseño y la creación de una aplicación para dispositivos móviles con un sistema operativo iOS. El objetivo principal es dotar al usuario de medios a través de los cuales la aplicación desarrollada pueda almacenar, añadir o eliminar todas sus rutinas de deporte. Asimismo, el usuario podrá recibir ejercicios propuestos por la aplicación móvil, sin necesidad de tener rutinas de ejercicio propias. Los ejercicios contarán con información y explicación complementaria para guiar al usuario y ayudar en la elección de los recursos que se necesitan. Esta aplicación también cuenta con un espacio en el que el usuario podrá aportar datos fisiológicos, entre otros el peso o la altura, para poder calcular ciertos valores muy conocidos en el entorno fitness. La aplicación también contará con un espacio donde se dispondrá de información nutricional. De esta forma, la aplicación puede ser incorporada a la vida diaria de una persona deportista, para que esta tenga un seguimiento de todos sus procesos diarios, un control detallado y exhaustivo de sus sesiones de deporte, y tener una herramienta a su disposición que le acompañe en todos sus progresos en el mundo del fitness, así como, comunicarse con otros usuarios registrados en la misma aplicación para poder

compartir cualquier información que consideren relevante dentro de un chat global. La aplicación también incorpora un mapa donde los usuarios tendrán a su disposición los gimnasios mas cercanos y podrán optar a planes relacionados con entrenamientos o nutrición que podrán adquirir en caso de estar interesados.

Abstract.

Nowadays, more and more people are starting to take more rigorous care of their health and physical appearance, and to do some kind of sport or physical exercise on a regular basis. It is becoming common for many people to go to the gym or even begin to develop some activity related to the world of fitness, such as running, CrossFit, etc. Therefore, the development of an application focused on a society like this can become a fundamental tool in the daily routine of any fitness enthusiast, and thus achieve a perfect planning of all their workouts. This project consists of the design and creation of an application for mobile devices with an iOS operating system. The main objective is to provide the user with the means through which the developed application can store, add or delete all their sport routines. In addition, the user will be able to receive exercises proposed by the mobile application, without the need to have their own exercise routines. The exercises will have complementary information and explanations to guide the user and help in the choice of the resources needed. This application also has a space where the user can provide physiological data, including weight and height, in order to calculate certain well-known values in the fitness environment. The application will also have a space where nutritional information will be available. In this way, the application can be incorporated into the daily life of a sportsperson, so that they can keep track of all their daily processes, a detailed and exhaustive control of their sports sessions, and have a tool at their disposal to accompany them in all their progress in the world of fitness, as well as communicate with other users registered in the same application to be able to share any information they consider relevant within a global chat. The application also incorporates a map where users will have at their disposal the nearest gyms and will be able to opt for plans related to training or nutrition that they can purchase if they are interested.

Índex

1	Introducció	1
1.1	Descripció.....	1
1.2	Motivació.....	2
1.3	Objectius.....	2
1.4	Planificació	3
2	Justificació tecnologies emprades.....	6
2.1	Tipus d'aplicacions mòbils.....	6
2.1.1	<i>Aplicacions natives</i>	7
2.1.2	<i>Aplicacions web</i>	7
2.1.3	<i>Aplicacions híbrides</i>	8
2.1.4	<i>Elecció</i>	9
2.2	Sistema operatiu.....	9
2.2.1	<i>iOS</i>	10
2.2.1.1	<i>Arquitectura iOS</i>	10
2.2.1.2	<i>Avantatges i desavantatges</i>	11
2.2.2	<i>Android</i>	11
2.2.2.1	<i>Arquitectura Android</i>	12
2.2.2.2	<i>Avantatges i desavantatges</i>	13
2.2.3	<i>iOS VS Android</i>	14
2.2.4	<i>Elecció</i>	14
3	Requisits.....	15
3.1	Requisits funcionals.....	15
3.2	Requisits no funcionals.....	16
3.3	Casos d'ús.....	17
3.3.1	<i>Actors</i>	17
3.3.2	<i>Diagrama dels casos d'ús</i>	17
3.3.3	<i>Especificació textual dels casos d'ús</i>	18
4	Entitats del projecte.....	25
4.1	Diagrama de classes d'entitats sense operacions.....	25
5	Disseny	26
5.1	Disseny de l'arquitectura Software.....	26
5.2	Disseny de la interfície	28
5.2.1	<i>SplashScreen</i>	31
5.2.2	<i>Login, Register, Forgot Password i UserData</i>	31
5.2.3	<i>Profile</i>	33
5.2.4	<i>Home</i>	33
5.2.5	<i>Gym</i>	34
5.2.5.1	<i>Exercises</i>	35
5.2.5.2	<i>Routines</i>	35
5.2.5.3	<i>Favorites</i>	36
5.2.6	<i>Crossfit</i>	37

5.2.7	<i>Diets, Healthy Food & Recipes</i>	37
5.2.8	<i>Global Chat</i>	38
5.2.8	<i>Health</i>	39
5.2.9	<i>Map</i>	39
5.2.10	<i>Settings</i>	40
5.3	Disseny de la base de dades	41
5.3.1	<i>Avantatges de bases de dades NoSQL</i>	41
5.3.2	<i>Avantatges de bases de dades en temps real</i>	42
5.3.3	<i>Disseny de l'arquitectura software de la base de dades</i>	43
6	Implementació	45
6.1	Llibreries utilitzades.....	45
6.2	Estructura del projecte.....	46
6.3	Firestore.....	48
6.4	Autenticació d'usuaris.....	49
6.4.1	<i>Recuperació de contrasenya</i>	50
6.4.2	<i>Eliminació d'usuari</i>	52
6.5	Servei de localització.....	54
6.5.1	<i>Agrupació de marcadors de gimnasos al mapa</i>	54
6.6	Servei de pagament.....	55
6.7	Servei de missatgeria.....	57
6.8	Estadístiques codi projecte.....	58
7	Avaluació	59
7.1	Crear new User.....	59
7.1.1	<i>Visualització perfil</i>	60
7.2	Verificació visualització principal aplicació.....	61
7.2.1	<i>Esports</i>	61
7.2.1.1	<i>Visualització exercicis</i>	61
7.2.1.2	<i>Visualització rutines</i>	62
7.2.1.3	<i>Verificació funcionament preferits</i>	63
7.2.2	<i>Nutrició</i>	63
7.2.2.1	<i>Visualització dietes</i>	63
7.2.2.2	<i>Visualització receptes</i>	65
7.2.2.3	<i>Visualització Healthy food</i>	65
7.2.3	<i>Visualització Health View</i>	66
7.3	Map View.....	67
7.4	Global Chat App.....	68
7.5	Verificació Plans.....	69
7.6	Validació del funcionament "Settings".....	71
7.6.1	<i>Verificació funcionament Log Out</i>	71
7.6.2	<i>Verificació eliminar usuari</i>	72
8	Conclusions	74
9	Referències	76

1 Introducció

En els darrers anys i mentrestant cursava els meus estudis del grau d'Enginyeria en Sistemes i Serveis de Telecomunicacions he començat un projecte personal que té com a principal motivació el desenvolupament d'una aplicació fitness per un producte de futur i que he anomenat "FitYou". Aquest projecte de final de carrera té com objectiu la creació de l'aplicació per iPhone que permeti portar a terme el projecte "FitYou".

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

1.1 Descripció

FitYou és una aplicació mòbil dissenyada exclusivament pel sistema operatiu iOS, que l'objectiu principal de l'aplicació és arribar a ser un monitor fitness amb informació esportiva i nutricional per l'usuari, que tingui tota la informació possible al seu abast. En primer lloc, aquesta aplicació, és una aplicació que està enfocada a les persones que practiquen esport amb certa freqüència i sobretot a la gent que realitza esport al món del gimnàs, CrossFit, etc.

En segon lloc, gràcies a aquesta aplicació, els seus usuaris podran desenvolupar unes certes tasques com poden ser: afegir rutines d'entrenaments personalitzades per poder-les visualitzar ràpidament amb l'aplicació, tenir un historial de totes les vegades que l'usuari ha calculat el seu índex de massa corporal depenent del pes i de l'altura de l'usuari, visualitzar dietes i poder veure així tota una varietat de informació nutricional relacionada amb el món del fitness, visualitzar dietes, menjars saludables, etc. L'usuari pot tenir una petita planificació del que ha de realitzar ja que l'aplicació li proporciona al usuari tota mena d'informació que ell/a necessita.

Entre altres coses, l'aplicació FitYou té com objectiu final facilitar tota informació nutricional, esportiva ja que deixa a la disposició dels usuaris certes plantilles o rutines d'entrenaments, exercicis concrets depenent del múscul a entrenar, rutines suggerides que l'usuari pot decidir si realitzar o no en els seus entrenaments, i obtenir diferents plans nutricionals o esportius o ambdós inclosos depenent de l'objectiu que tingui l'usuari amb el seu pla fitness. Per tal de motivar als usuaris de l'aplicació, aquests podran comunicar-se amb altres usuaris de l'aplicació mitjançant un xat global i també tindran la possibilitat de realitzar certes compres dins de la mateixa aplicació depenent de les seves necessitats.

Tenir en compte, que una aplicació de fitness com és FitYou, és una aplicació dissenyada i desenvolupada per a què el mateix usuari pugui tenir un control o seguiment de tots els seus entrenaments, objectius, reptes, ingesta dels seus aliments i que tingui constància de quin patró d'entrenament està seguint. I sobretot tenint com a punt central que l'usuari pugui centrar-se en la seva pròpia salut i creixement personal dins del món de l'esport.

1.2 Motivació

L'objectiu principal a l'hora d'escollir aquest projecte va ser dissenyar una aplicació que pogués ser suport del desenvolupament i seguiment de qualsevol esport incloent-hi els ítems bàsics de qualitat.

Actualment hi ha molta gent que està incorporant-se al món del fitness, molta gent està començant a cuidar-se cada cop més i comencen a tenir més cura dels aliments que mengem. La barrera d'entrada quan ens plantegem 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 com el del fitness i tenen dubtes sobre quins plans d'entrenament seguir, quins consells alimentaris tenir en compte, i sobretot un dels problemes més grans és la mancança d'informació 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 del Fitness pugui trobar una guia i aquesta li permeti tenir a l'abast tot el que sigui necessari per poder tenir un seguiment, un control, i un pla detallat per no trobar-se perdut i sense saber què fer.

No obstant això, l'aplicació desenvolupada té com objectiu ajudar a tothom, ja sigui per una persona que just acaba de començar en un món com és el fitness, com per una persona que ja fa temps que esta immensa en aquest món i requereix unes necessitats totalment diferents.

1.3 Objectius

Més enllà dels objectius merament de l'aplicació, el meu primer l'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 iOS en assignatures com "Programació Avançada en Dispositius Mòbils", en la que vam realitzar el disseny i el desenvolupament d'una aplicació per poder realitzar el lloguer de patinets elèctrics, juntament amb l'ajuda del professor Tomàs González Sánchez, que afegir i agrair-li personalment l'acceptació d'un projecte com aquest i ser el meu tutor de treball de final de grau.

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 iOS, és a dir, dispositius iPhone i iPad.

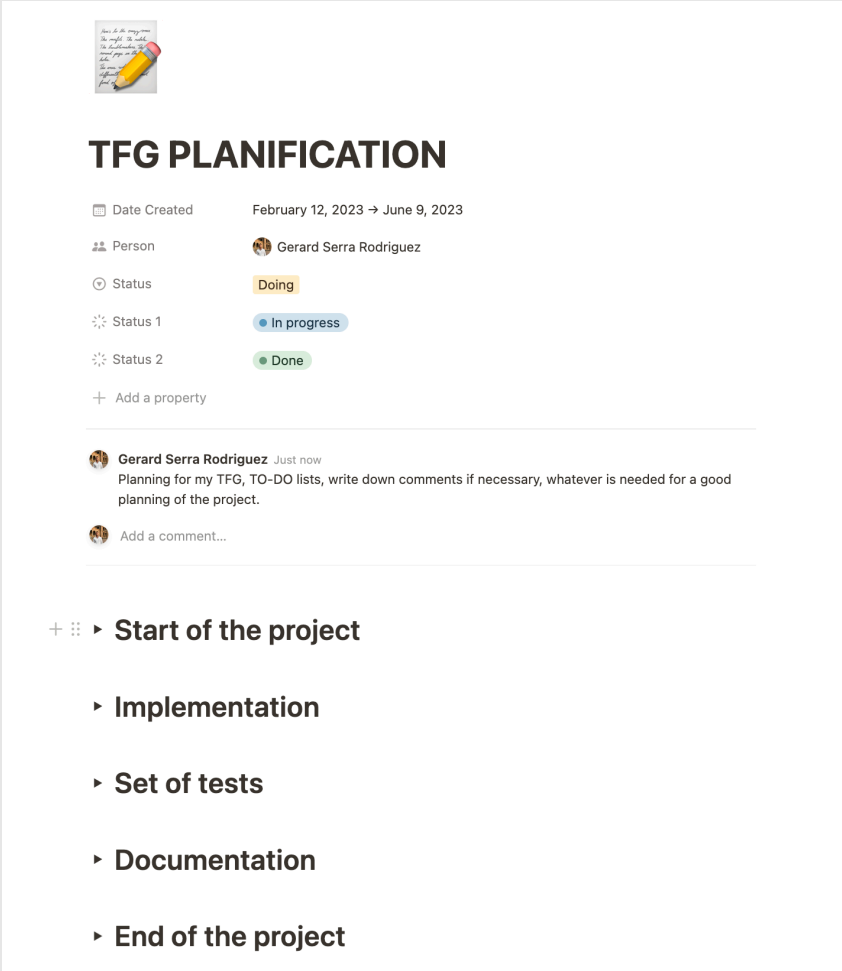
Per altra banda, crear un projecte, únicament i exclusivament per un sistema operatiu, en aquest cas iOS, és un dels altres objectius a tenir en compte, ja que personalment utilitzo aquest programari i puc dur a terme el joc de proves en el meu mateix dispositiu. Finalment, realitzar un projecte d'una aplicació real i amb un objectiu concret em permet llençar l'aplicació al mercat d'Apple, com per exemple "Apple Store", podent convertir la resolució del meu problema o dels meus interessos en un producte exclusiu i que pugui donar-me ingressos prestant un servei a gent de l'àmbit fitness.

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 projecte per corregir possibles desviacions temporal i que aquestes no afectin a la data de lliurament o acabament. Per la planificació d'aquest treball utilitzaré "Notion" [1] ja que ofereix plantilles de gestió de projectes bastant fàcils d'utilitzar, així com altres funcions essencials per la gestió d'aquests.

El projecte està dividit en 5 parts diferents tal com es pot apreciar en la *figura 1*:

- Introducció.
- Implementació.
- Proves.
- Documentació.
- Fi del projecte



The image shows a Notion page titled "TFG PLANIFICATION". At the top left is a small icon of a notepad and pencil. Below the title, there are several metadata fields: "Date Created" (February 12, 2023 → June 9, 2023), "Person" (Gerard Serra Rodriguez), "Status" (Doing), "Status 1" (In progress), and "Status 2" (Done). Below these fields is a section for comments, with a comment from Gerard Serra Rodriguez stating: "Planning for my TFG, TO-DO lists, write down comments if necessary, whatever is needed for a good planning of the project." Below the comments is a list of project phases, each preceded by a right-pointing arrow: "Start of the project", "Implementation", "Set of tests", "Documentation", and "End of the project".

Figura 1. Planificació del Projecte

▼ Start of the project

- ✓ Introduction
 - ✓ Submit idea to tutor
 - ✓ Meeting with the tutor to discuss the topic
- ✓ Preparing Software
 - ✓ Xcode
 - ✓ Emulator
 - ✓ Github repository
- ✓ Software Design
 - ✓ Functional requirements
 - ✓ Non-functional requirements
 - ✓ Cases of use
- ✓ Entities
 - ✓ Class diagram without operations
 - ✓ UI/UX Design
 - ✓ BBDD Design

Figura 2. Planificació d'Inici del projecte

▼ Implementation

- Create Xcode-project
- Connect project with firebase
- Connect to GitHub repository
- User authentication
- ▼ Views
 - Splash
 - Login & Register
 - Reset Password
 - User Data View
 - Profile View
 - Home View
 - Sports View
 - Nutrition View
 - Exercise & Routine
 - Food, Diets & Recipes
 - Map View
 - Global Chat View
 - Plans View
 - Settings View
 - Health View
- Logical functioning of the app

Figura 3. Planificació Implementació

▼ Set of tests

- Test all functionalities

Figura 4. Planificació Joc de proves

▼ Documentation

- Comments in the code
- Write all documentation
- General review of all documentation
- Assessment of all documentation with the tutor

Figura 5. Planificació Documentació

▼ End of the project

- Project completed and marked as good

Figura 6. Planificació Final de Projecte

TFG TIMELINE						
Timeline view +						
Aa Name	Assign	Blocked by	Date	Status	continue	+ ...
START OF THE PROJECT	Gerard Serra Rodriguez		February 12, 2023 → Februar	Completed	IMPLEMENTATION	
IMPLEMENTATION	Gerard Serra Rodriguez	START OF THE PROJECT	March 1, 2023 → April 30, 20	Completed	SET OF TESTS	
SET OF TESTS	Gerard Serra Rodriguez	IMPLEMENTATION	May 3, 2023 → May 7, 2023	Completed	DOCUMENTATION	
DOCUMENTATION	Gerard Serra Rodriguez	SET OF TESTS	May 10, 2023 → June 1, 2023	Completed	END OF THE PROJECT	
END OF THE PROJECT	Gerard Serra Rodriguez	DOCUMENTATION	June 3, 2023 → June 9, 2023	Completed		

+ New

COUNT 5

Figura 7. Timeline

2 Justificació tecnologies emprades

En aquest apartat es realitza un petit anàlisi de les diferents tecnologies que s'han emprat per poder desenvolupar i dissenyar una aplicació mòbil, així com, la motivació que ha portat a utilitzar una llibreria, framework o tecnologia en concret i no una altra.

Tanmateix, es realitza un revisió dels diferents sistemes operatius per a dispositius mòbils i les motivacions que m'han portat a triar el sistema operatiu iOS i no d'altres.

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 mòbils que són desenvolupades específicament per a una plataforma o sistema operatiu en particular, com ara iOS, Android, etc. Aquestes aplicacions estan escrites en llenguatges de programació que són compatibles amb el sistema operatiu de destinació, i això permet als desenvolupadors accedir a les funcions i característiques del dispositiu o sistema operatiu de manera més directa.

Les aplicacions natives també tenen un SDK (Software Development Kit) propi de la plataforma inclòs una Developer API. Això significa que els desenvolupadors poden crear aplicacions que estiguin totalment integrades amb el sistema operatiu i altres aplicacions en el dispositiu. Això pot resultar en una experiència d'usuari més fluida i personalitzada. No obstant això, el desenvolupament d'aplicacions natives pot ser més complex i requerir un temps i recursos de desenvolupament més elevats en comparació amb altres formes de desenvolupament d'aplicacions com les aplicacions híbrides o les aplicacions web progressives.

Les avantatges que podem trobar en aplicacions natives poden ser:

- Millor rendiment i velocitat.
- Millor disseny d'interfície d'usuari.
- Accés a les funcions del dispositiu.
- Disponibilitat en les botigues d'aplicacions.
- 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.

2.1.2 Aplicacions web

Les aplicacions web, es poden programar independentment del sistema operatiu en el que s'utilitzi l'aplicació, és a dir, això ho diferencia de les aplicacions natives. Són aplicacions basades en web que no requereixen que l'usuari descarregui o instal·li res al seu dispositiu, aquestes son accessibles des de qualsevol lloc on l'usuari pugui connectar-se a Internet i no estan limitades com he comentat anteriorment, per la plataforma o el sistema operatiu que utilitzi l'usuari.

A diferència de les aplicacions natives, que estan escrites en llenguatges de programació natius 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, son més fàcils i menys costoses de dissenyar i desenvolupar perquè no es necessita desenvolupadors que coneguin la complexitat del sistema operatiu al que van destinades.

Les avantatges que podem trobar en una aplicació web:

- Accessibilitat.
- Compatibilitat..
- Actualitzacions fàcils
- Desenvolupament més fàcil i menys costos.
- Actualització instantània del contingut.
- No necessita aprovació externa per publicar-se.

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.
- Access molt limitat a las característiques hardware del dispositiu.
- Experiència del usuari pot disminuir, ja que no sempre estan perfectament adaptades a les diferents mesures.
- Les aplicacions web, a la pràctica, s'executen utilitzant el navegador web present al dispositiu mòbil i a la pràctica augmenta la complexitat en el desenvolupament perquè hi ha un gran nombre de navegadors web i versions al mercat.

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 son creades mitjançant un marc de treball híbrid, que permet 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.
- Accés a les funcions del dispositiu.
- Funcionalitat multi plataforma.

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

- Rendiment més lent.
- Dependència del marc de treball.
- Limitacions de personalització.

- Actualitzacions de l'API de desenvolupament (SDK). Les actualitzacions del framework utilitzat no són molt ràpides i depenen del proveïdor del framework per utilitzar les darreres versions o capacitats del SDK natiu.
- Dependent de la connexió a Internet..
- La gran majoria d'aplicacions híbrides tenen les mateixes limitacions que les aplicacions desenvolupades amb tecnologies web.

2.1.4 Elecció

Un cop valorades les avantatges i desavantatges que proporcionen cadascuna de les possibilitats, he decidit emprar en aquest projecte el SDK proporcionat per Apple per la creació d'aplicacions mòbils. És a dir, la creació d'una aplicació nativa ja que el que realment busco en el desenvolupament i disseny de l'aplicació, és proporcionar una experiència d'usuari optimitzada i personalitzada, tal com s'ha descrit prèviament en els avantatges i els desavantatges. El recurs web [2] fa un bon anàlisi del desenvolupament d'aplicacions natives i m'ha ajudat a triar aquesta opció en el procés de desenvolupament.

Altrament, el fet de desenvolupar una aplicació nativa, em dona accés més directe a les funcions de la plataforma, el que significa es que com a desenvolupador puc oferir una interfície d'usuari i una experiència de navegació més fluides. Així com proporcionar una millora del rendiment del dispositiu en l'execució de la meva aplicació ja que està optimitzada per una plataforma específica.

2.2 Sistema operatiu

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

El SO és responsable de les següents tasques:

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

Existeixen diversos sistemes operatius creats exclusivament per mòbils que estan dissenyats per optimitzar els seu recursos. Es van crear exclusivament per dispositius mòbils donat que aquests tenen característiques hardware més limitades que els ordinadors desktop o portàtils. Actualment, el sistema operatiu per dispositius mòbils més emprat és Android OS, conegut per la majoria d'usuaris com Android. En seccions posteriors es farà una descripció acurada del sistema operatiu Android.

En aquesta secció de la documentació, es farà una descripció dels sistemes operatius: Android i iOS, i les motivacions que poden portar a l'elecció d'un SO o un altre per ser

utilitzat en la creació d'una aplicació per dispositius mòbils. I les motivacions que m'han portat a triar el sistema operatiu iOS.

2.2.1 iOS

iOS és un sistema operatiu mòbil dissenyat i utilitzat per Apple pels seus dispositius mòbils, com poden ser l'iPhone, l'iPad, i l'iPod Touch. Aquest sistema operatiu és conegut per la seva interfície d'usuari elegant i intuïtiva, la seva seguretat i privacitat avançada, i la seva integració amb altres dispositius i serveis d'Apple.

Aquest sistema operatiu, és el que se suposa el màxim competidor pel sistema operatiu *Android*. iOS es troba present únicament en els dispositius Apple, ja que són els únics que disposen de llicència poder-lo utilitzar i aquest fet que limita notablement la seva quota de mercat.

2.2.1.1 Arquitectura iOS

En aquest sistema operatiu, iOS, l'arquitectura que trobem, és una combinació de software i hardware dissenyats per Apple, per tal de poder oferir un rendiment òptim als seus dispositius mòbils, els quals hem anomenat prèviament.

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

A nivell de software, iOS està basat en el kernel de macOS, però ha estat optimitzat per funcionar en dispositius mòbils. El kernel és el component central del SO que s'encarrega del control dels recursos de hardware i la comunicació dels diferents components del software entre si.

L'arquitectura d'iOS es pot entendre com una arquitectura de capes, és a dir, 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 Developers [3].

Les capes de l'arquitectura, com podem veure a la *figura 8*, són:

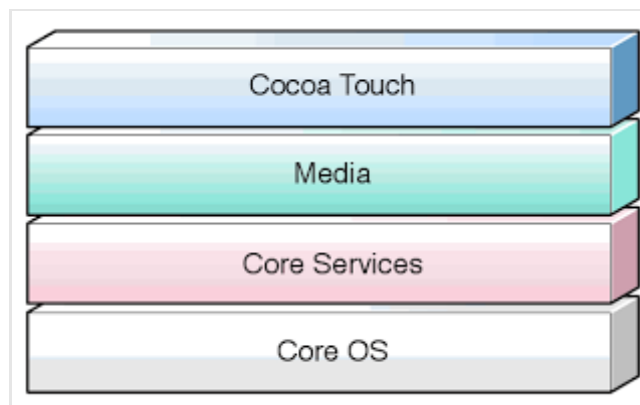


Figura 8. Arquitectura iOS

- Cocoa Touch: Aquesta capa proporciona els components d'interfície d'usuari per a les aplicacions iOS i també inclou els frameworks que s'utilitzen per les funcions comunes de les aplicacions, com ara l'accés als contactes, càmera, etc.
- Media: Aquesta capa proporciona els frameworks per a la gestió del contingut multimèdia, com ara àudio, vídeo i 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 OS: Aquesta capa proporciona els components del sistema operatiu a nivell més baix, com ara el Kernel del SO.

2.2.1.2 Avantatges i desavantatges

iOS té una sèrie d'avantatges i desavantatges que cal considerar a l'hora de triar-lo pel desenvolupar aplicacions per a certs dispositius.

Avantatges:

- Interfície d'usuari atractiva i intuïtiva.
- Mercat d'usuaris.
- Plataforma segura.
- Bona integració amb hardware.
- Qualitat d'aplicacions.

Desavantatges:

- Preu elevat per desenvolupar aplicacions iOS.
- Únic entorn disponible per desenvolupar aplicacions amb aquest llenguatge, Xcode.
- Dependència de les eines d'Apple.
- Limitacions de personalització
- Control exhaustiu de permisos.

2.2.2 Android

Android és un sistema operatiu mòbil desenvolupat per Google i dissenyat per a dispositius mòbils. El sistema operatiu Android és de codi obert, és a dir, que els desenvolupadors poden modificar i personalitzar el codi font segons les seves necessitats.

Tal com s'explica a la documentació oficial d'Android [4], el sistema operatiu es centra en proporcionar una experiència d'usuari de primera classe, amb funcions com ara una interacció intuïtiva, una personalització flexible i una gestió de la vida de la bateria intel·ligent. Android té una gran diversitat de dispositius i una gran quantitat d'aplicacions disponibles a través de la botiga oficial, Google Play Store. No obstant, la fragmentació del sistema operatiu pot arribar a ser un repte per als desenvolupadors d'aplicacions.

2.2.2.1 Arquitectura Android

Android es basa en una arquitectura de capes segons indica la documentació oficial de l'arquitectura d'Android [5], amb diferents components que treballen junts per així poder proporcionar una experiència d'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 poden ser actualitzats de forma independent, sense afectar la resta del sistema. Això permet que els fabricants de dispositius puguin personalitzar les seves implementacions d'Android i proporcionar funcions úniques als seus dispositius.

Les capes de l'arquitectura, com podem veure a la *figura 9*, són:

- Aplicació: aquesta capa és la que es troba a la part superior de l'arquitectura i representa les aplicacions que es veuen en la pantalla del dispositiu.
- Frameworks d'aplicacions: aquesta capa proporciona un conjunt de components i serveis que les aplicacions poden utilitzar per accedir a les funcions del sistema. Aquesta capa inclou el gestor d'activitats, que gestiona el cicle de vida de les activitats de l'aplicació, i el gestor de serveis, que gestiona els serveis que s'executen en segon pla.
- Biblioteques de sistemes: aquesta capa inclou diferents biblioteques que proporcionen funcions com ara gràfics 2D i 3D, bases de dades, comunicació de xarxa, entre d'altres.
- Android Runtime: aquesta capa inclou l'entorn d'execució d'aplicacions Android (ART), que és responsable d'executar el codi de l'aplicació.
- Kernel de Linux: aquesta capa és la capa més baixa de l'arquitectura d'Android i proporciona l'accés als recursos del maquinari del dispositiu, com ara la CPU, la memòria i els dispositius d'entrada/sortida. Android utilitza el nucli de Linux com a sistema operatiu subministrador.

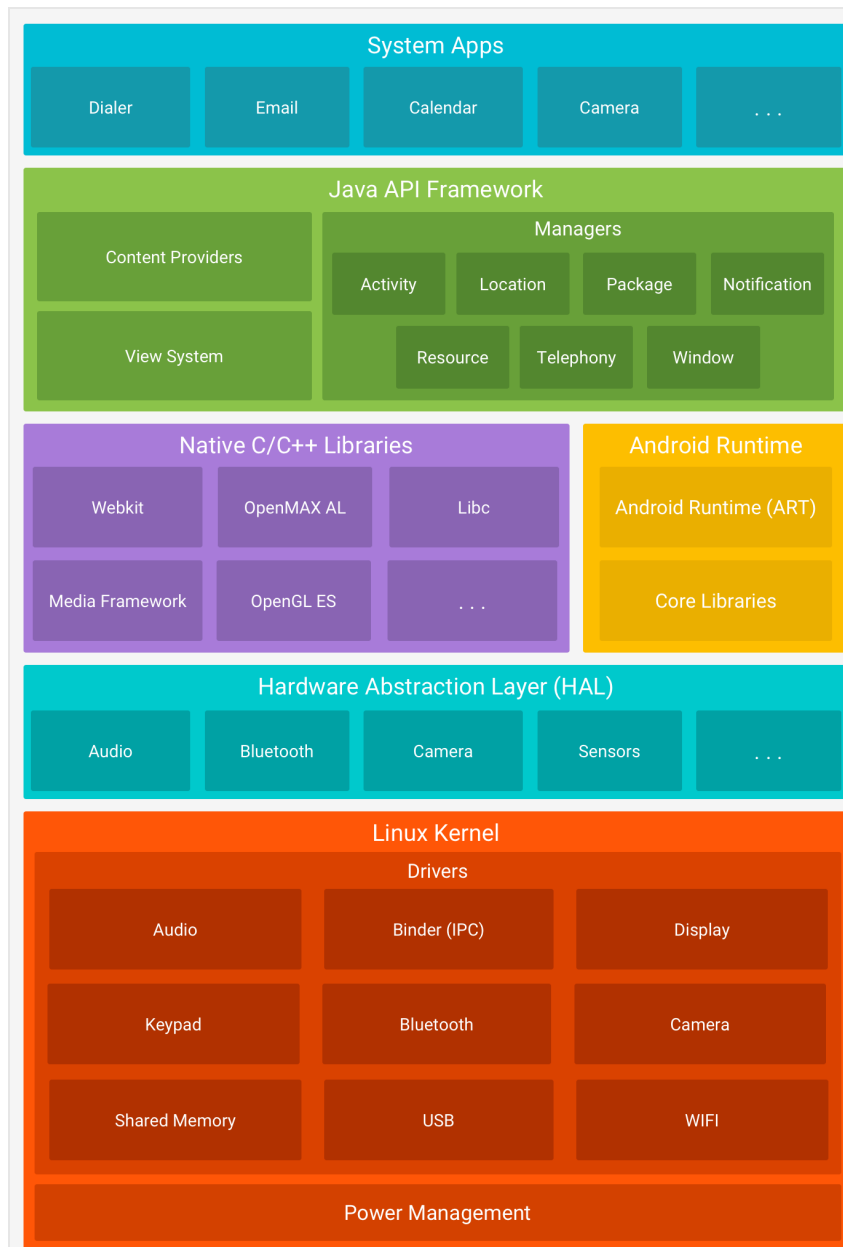


Figura 9. Arquitectura Android

2.2.2.2 Avantatges i desavantatges

Android, com iOS també té una sèrie d'avantatges i desavantatges que cal considerar a l'hora de ficar-se a desenvolupar aplicacions per a certs dispositius.

Els avantatges:

- Diversitat de dispositius.
- Personalització.
- Integració amb Google.
- Software lliure i multi-plataforma.
- Gestió de multi-tasca.
- Gran nombre d'aplicacions.

- Interfície intuïtiva.

Els desavantatges del sistema operatiu Android són:

- La gran varietat de versions obliga als desenvolupadors a estar constantment revisant l'aplicació i el seu funcionament.
- La gestió de multi-tasca comentada prèviament com a benefici, també pot ser considerat com una desavantatge, ja que al tenir varies aplicacions en segon pla fa que es rellenteixi el dispositiu, aleshores augmenta el consum de bateria.

2.2.3 *iOS VS Android*

Com a grans sistemes operatius per a dispositius mòbils, iOS i Android tenen molts punts forts i febles que els diferencien, així que a continuació, realitzaré una petita comparació entre aquests dos sistemes operatius en diferents aspectes que he trobat remarcables/importants.

- Facilitat d'ús: iOS és conegut per la seva interfície d'usuari intuïtiva i fàcil d'utilitzar amb icones clares i senzilles, i un disseny coherent en tot el sistema operatiu. Android, per la seva banda, ofereix més personalització i flexibilitat a l'hora de que l'usuari realitzi els canvis que li vinguin de gust, però això pot fer que la seva interfície sigui més complicada per als usuaris menys experimentats.
- Selecció d'aplicacions: Apple té un control més gran sobre la botiga d'aplicacions, l'App Store, i, per tant, pot garantir una major qualitat. No obstant això, també significa que hi ha menys opcions disponibles en comparació amb la tenda d'aplicacions d'Android, que té una selecció més gran d'aplicacions i jocs, però també pot presentar això molts més riscos de seguretat per l'usuari i per l'aplicació.
- Integració de hardware: Apple fabrica tant el hardware com el software dels seus dispositius, el que els permet proporcionar una integració més estreta i optimitzar millor el rendiment. En el cas d'Android, l'ecosistema és més obert i pot haver-hi més variabilitat en el rendiment de diferents dispositius.
- Preu: Els dispositius iOS tendeixen a ser més cars que els dispositius Android de gamma mitjana i baixa, cosa que pot ser un factor important per als consumidors que busquen una opció més assequible.

En conclusió, iOS i Android són dos sistemes operatius diferents que ofereixen avantatges i desavantatges segons les necessitats i preferències de l'usuari.

2.2.4 *Elecció*

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

Una de les motivacions principals d'aquesta aplicació es permetre l'ús personal de l'aplicació i poder treure el màxim benefici possible econòmicament, per el que em sembla més lògic el desenvolupament i el disseny de l'aplicació per el sistema operatiu iOS.

Un altre motiu de pes per el desenvolupament d'aquest projecte dirigit al sistema operatiu iOS, és que desenvolupar l'aplicació accessible per aquest SO és necessari la disposició de un terminal Mac (Ordinadors de la marca Apple) i de un iPhone (terminal mòbil de la marca Apple), que personalment dispo de aquests dos dispositius. A més, el sistema operatiu iOS ofereix una gran quantitat de recursos i eines per als desenvolupadors, incloent una extensa documentació i suport tècnic, així com una gran varietat de biblioteques i frameworks que simplifiquen el procés de desenvolupament.

3 Requisites

A continuació exposaré els elements bàsics dels requisits funcionals i no funcionals de l'aplicació, i tot seguit, en el punt nº 4 d'aquesta documentació realitzaré l'anàlisi de les entitats del projecte..

La realització d'aquests dos punts m'ajudaran a especificar amb molt més detall les taques realitzar a l'hora del desenvolupament i disseny del software. I l'anàlisi de casos d'ús em permetrà identificar els diferents comportaments de l'aplicació a desenvolupar, i em m'ajudarà especificar les accions que permetran que els usuaris interaccionin amb l'aplicació desenvolupada.

3.1 Requisites funcionals

A continuació realitzaré l'anàlisi permenoritzat dels requisits funcionals.

A l'inici del projecte es van enumerar els requisits que ha de tenir l'aplicació, és a dir, els objectius funcionals i que es considera important que han de ser presents a l'aplicació desenvolupada. La taula següent mostra quins són aquests requisits i quines serien les parts més importants d'aquest projecte, agafant com a referència [5] que descriu els requisits funciona i no funcionals.

Requisit Funcional	Descripció Requisit Funcional
RF1	Permetre al usuari identificar-se / autenticar-se mitjançant 2 mètodes: correu/contrasenya, Google. Són els serveis més populars a les aplicacions per poder entrar al sistema.
RF2	Al registrar-se per primera vegada, l'aplicació permet al usuari omplir les dades sobre el seu perfil.
RF3	Editar les dades personals.

RF4	Eliminar permanentment el seu compte de l'aplicació.
RF5	Permetre a l'usuari escollir entre diferents esports proporcionats per l'aplicació, així se li mostrarà un contingut o un altre.
RF6	Permetre a l'usuari la visualització d'exercicis, amb el títol, la descripció i un vídeo corresponent a la realització d'aquell exercici seleccionat en concret.
RF7	Permetre a l'usuari, la selecció de rutines, exercicis, etc. Com a preferides, així es guardaran en un apartat diferent perquè l'usuari les tingui accessibles més ràpidament.
RF8	L'usuari tindrà accés a un xat global de l'aplicació per poder comunicar-se amb els altres usuaris registrats en l'aplicació.
RF9	L'usuari serà capaç de calcular-se el BMI (Body Mass Index), afegint el seu propi pes i alçada.
RF10	L'usuari tindrà accessible uns consells alimentaris
RF11	L'usuari podrà reinicialitzar la contrasenya en cas que se n'hagi oblidat.
RF12	L'usuari tindrà certes accessibilitats disponibles al seu abast.
RF13	L'usuari tindrà a la seva disposició un mapa amb la seva ubicació marcada i tots els gimnasos disponibles al seu voltant tots ells identificats.
RF14	L'usuari tindrà a la seva disposició una pantalla amb tres plans diferents per poder realitzar compres
RF15	L'usuari tindrà a la seva disposició un historial amb els diferents BMI calculats al cap del temps
RF16	L'usuari podrà tancar la sessió de l'aplicació

3.2 Requisits no funcionals

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

Aquests requisits són els que s'originen 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 [5].

Requisit No Funcional	Descripció Requisit No Funcional
RNF1	El disseny de l'aplicació és intuïtiu, fàcil de navegar per tota l'aplicació.
RNF2	L'aplicació ha de tenir persistència de dades.
RNF3	Algunes parts del sistema: creació de rutines, visualització de vídeos explicatius, missatges, etc. Ha d'anar en temps real.
RNF4	Utilització de productes de Firebase, Firestore (base de dades), Firebase Authentication, etc.
RNF5	Possibilitat que en un futur, pugui haver-hi la implementació d'una aplicació web.

3.3 Casos d'ús

En aquest apartat de la documentació, parlaré i descriure els diferents casos d'ús, els actors que intervenen en l'execució de l'aplicació.

3.3.1 Actors

L'únic actor que actua en l'execució d'aquesta aplicació es "*User*".

3.3.2 Diagrama dels casos d'ús

En aquest apartat mostraré el diagrama dels diferents casos d'ús d'aquest "*User*" que he dut a terme gràficament.

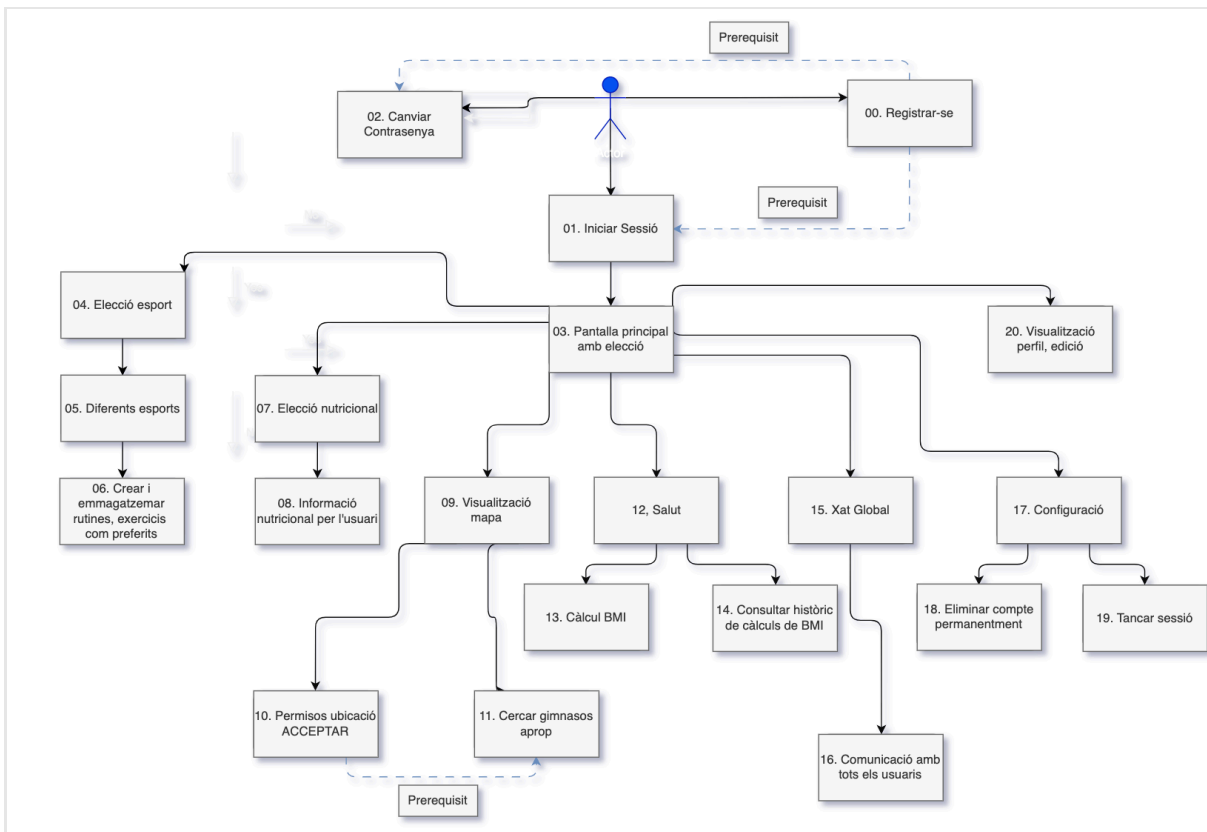


Diagrama 1. Casos d'ús.

3.3.3 Especificació textual dels casos d'ús

Cas d'ús 00. Registrar-se

Resum de la funcionalitat: L'usuari es pot registrar a l'aplicació.

Paràmetres d'entrada: correu electrònic, contrasenya.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: Cap.

Postcondició: L'usuari es crea un compte nou a l'aplicació.

Procés normal principal:

1. L'usuari decideix quin correu i quina contrasenya utilitzar per registrar-se.
2. L'aplicació connecta amb Firebase Firestore Database.
3. El Firebase crea un nou usuari amb aquest correu i aquest contrasenya i se li assigna un identificador d'usuari automàtic (uid).
4. Es redirigeix a l'usuari a una pantalla per que pugui indicar el gènere, l'edat i el nickname que utilitzarà a l'aplicació.
5. Firebase rep les dades introduïdes per l'usuari i les emmagatzema a la base de dades amb l'identificador del usuari creat.
6. El sistema mostra la pantalla principal perquè l'usuari pugui començar a fer-ne ús.

Alternatives de procés i excepcions:

- 1a. Ja existeix un compte amb aquest correu electrònic
- 1b. El sistema li indica al usuari que ja hi ha un compte registrat.

Cas d'ús 01. Iniciar Sessió

Resum de la funcionalitat: L'usuari pot iniciar sessió al sistema o amb el compte que ha creat prèviament o amb la plataforma de Google.

Paràmetres d'entrada: correu electrònic, contrasenya.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: Si l'usuari vol iniciar sessió mitjançant Google, ha de disposar d'un compte en aquesta plataforma o haver-se registrat prèviament amb usuari i contrasenya.

Postcondició: cap.

Procés normal principal:

1. L'usuari decideix quin mètode utilitzar per iniciar Sessió.
2. L'aplicació connecta amb Firebase Auth Service.
3. El Firebase retorna al sistema l'identificador del usuari (uid).
4. El sistema verifica si amb aquest identificador, existeix un usuari.
5. Es processa l'autenticació d'usuari i es dirigeix a la pantalla principal de l'aplicació.
6. El sistema mostra la pantalla principal perquè l'usuari pugui començar a fer-ne ús.

Alternatives de procés i excepcions:

4a. No existeix un compte amb aquest correu electrònic

4a1. El sistema li indica al usuari que no hi ha cap compte creat amb aquestes credencials, i l'usuari s'ha de dirigir a la pantalla de registre. (Cas d'ús 00).

Cas d'ús 02. Canviar contrasenya

Resum de la funcionalitat: L'usuari pot canviar la seva contrasenya, en cas de que se'n hagi oblidat de la seva.

Paràmetres d'entrada: correu electrònic

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: Cap.

Postcondició: La contrasenya del usuari queda canviada.

Procés normal principal:

1. L'usuari escriu el correu amb el que va registrar-se.
2. L'aplicació connecta amb Firebase Firestore Database.
3. El Firebase agafa el correu d'aquest usuari i envia un correu electrònic per la regeneració de la contrasenya de l'usuari.
4. L'usuari s'ha de dirigir al seu correu electrònic.
5. Es redirigeix a l'usuari a una pàgina web per què pugui indicar la nova contrasenya.
6. Firebase rep les dades introduïdes per l'usuari i les emmagatzema a la base de dades amb l'identificador del usuari corresponent.
7. La pàgina web mostra com la contrasenya ha sigut canviada correctament.
8. L'usuari ja pot tornar a l'aplicació i tornar a iniciar sessió amb la nova contrasenya.

Alternatives de procés i excepcions:

- 1a. No existeix cap compte amb aquest correu
- 1b. El sistema li indica al usuari que no hi ha cap compte amb aquest correu electrònic i no es realitza cap procés.

Cas d'ús 03. Pantalla principal amb elecció

Resum de la funcionalitat: L'usuari es troba a la pantalla principal i pot escollir que fer dins de l'aplicació i cap a on dirigir-se.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: cap.

Postcondició: cap.

Procés normal principal:

1. L'usuari decideix quina ruta agafar dins de la pantalla principal i quines accions realitzar.

Alternatives de procés i excepcions: cap.

Cas d'ús 04. Elecció esport

Resum de la funcionalitat: L'usuari pot escollir quin esport vol visualitzar.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: cap.

Postcondició: cap.

Procés normal principal:

1. L'usuari decideix quina ruta agafar i quin esport escollir.

Alternatives de procés i excepcions: cap.

Cas d'ús 05. Diferents Esports

Resum de la funcionalitat: L'usuari té a la vista tots els esports de l'aplicació.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: cap.

Postcondició: Elecció d'una vista o una altra.

Procés normal principal:

1. L'usuari decideix quina ruta agafar i quin esport escollir.

Alternatives de procés i excepcions: cap.

Cas d'ús 06. Crear i emmagatzemar rutines, exercicis, com a preferits

Resum de la funcionalitat: L'usuari guardar exercicis i rutines com a preferides i també té la possibilitat d'afegir rutines personalitzades a l'aplicació.

Paràmetres d'entrada: Rutina nova.

Paràmetres de sortida: Rutina nova, exercicis i rutines marcades com a preferides.

Actors: Usuari.

Precondició: cap.

Postcondició: Crearà una llista d'exercicis i rutines preferides, i crearà rutines seves personalitzades.

Procés normal principal:

1. L'usuari pot escollir quin exercici o rutina vol marcar com a preferida i pot decidir si crear una rutina o no.
2. Aquells exercicis i rutines que marqui com a preferides es guardaran a la col·lecció de Firebase de l'usuari per què les tingui disponibles a la vista de preferits.
3. Aquelles rutines creades per ell mateix s'emmagatzemaran a Firebase i l'usuari les podrà visualitzar a l'apartat de les seves rutines

Alternatives de procés i excepcions: cap.

Cas d'ús 07. Elecció nutricional

Resum de la funcionalitat: L'usuari pot escollir quina informació nutricional vol realitzar.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: cap.

Postcondició: cap.

Procés normal principal:

1. L'usuari pot escollir quina informació nutricional vol realitzar.

Alternatives de procés i excepcions: cap.

Cas d'ús 08. Informació nutricional per l'usuari

Resum de la funcionalitat: L'usuari té a l'abast tota la informació nutricional que ell hagi escollit visualitzar al cas d'ús 007.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: cap.

Postcondició: cap.

Procés normal principal:

1. L'usuari visualitza la informació escollida prèviament.

Alternatives de procés i excepcions: cap.

Cas d'ús 09. Visualització de mapa

Resum de la funcionalitat: Mostra un mapa interactiu amb els pins de gimnasos propers a la ubicació de l'usuari.

Paràmetres d'entrada: Permís d'ubicació. Cas d'ús 10.

Paràmetres de sortida: Mapa amb els pins de gimnasos.

Actors: Usuari.

Precondició: L'usuari ha atorgat permís d'ubicació (cas d'ús 10).

Postcondició: : L'usuari pot veure un mapa amb els pins dels gimnasos propers a la seva ubicació (cas d'ús 11)

Procés normal principal:

1. L'usuari selecciona la vista del mapa.
2. Li salta una alerta conforme ha d'autoritzar a l'aplicació per tal de poder accedir a la seva ubicació
3. Un cop l'usuari ha acceptat, es mostra la seva ubicació al mapa amb els pins dels gimnasos

Alternatives de procés i excepcions: cap.

Cas d'ús 12. Apartat de salut

Resum de la funcionalitat: L'usuari pot escollir si entrar a la vista de salut.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: cap.

Postcondició: cap.

Procés normal principal:

1. L'usuari tria la vista de la salut.

Alternatives de procés i excepcions: cap.

Cas d'ús 13. Càlcul del BMI

Resum de la funcionalitat: Permet a l'usuari calcular el seu índex de massa corporal (IMC) introduint l'alçada i el pes.

Paràmetres d'entrada: Alçada i pes de l'usuari.

Paràmetres de sortida: Valor del IMC calculat.

Actors: Usuari.

Precondició: Cap.

Postcondició: L'usuari obté el valor del seu IMC calculat.

Procés normal principal:

1. L'usuari ha d'introduir les seves dades als camps corresponents
2. Un cop l'usuari ha introduït les dades ha de polsar el botó de calcular
3. Se li mostra al usuari el resultat amb la informació corresponent

4. Aquelles dades passen a ser emmagatzemades al històric de Firebase d'aquell usuari

Alternatives de procés i excepcions: cap.

Cas d'ús 14. Consultar històric de càlculs de BMI

Resum de la funcionalitat: Permet a l'usuari veure l'historial de tots els càlculs realitzats del seu índex de massa corporal (IMC).

Paràmetres d'entrada: Cap.

Paràmetres de sortida: Llista d'historial de càlculs del IMC.

Actors: Usuari.

Precondició: L'usuari ha realitzat càlculs del seu IMC en el passat.

Postcondició: L'usuari pot veure una llista amb l'historial dels càlculs del seu IMC.

Procés normal principal:

1. L'usuari ha de dirigir-se a la pantalla del històric
2. Un cop l'usuari s'ha dirigit a la pantalla corresponent, observa una llista amb tots els càlculs previs
3. Si l'usuari clica sobre un d'ells, s'obrirà una vista detall
4. Mentre s'està obrint la vista detall, s'estan recuperant les dades de la col·lecció de Firebase per poder ser mostrades al usuari.

Alternatives de procés i excepcions: cap.

Cas d'ús 15. Xat global

Resum de la funcionalitat: L'usuari pot escollir si entrar a la vista de la conversació global de l'aplicació.

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: cap.

Postcondició: cap.

Procés normal principal:

1. L'usuari tria la vista de la conversació.

Alternatives de procés i excepcions: cap.

Cas d'ús 16. Comunicació amb tots els usuaris

Resum de la funcionalitat: Permet a l'usuari comunicar-se amb altres usuaris mitjançant un xat global de l'aplicació.

Paràmetres d'entrada: Missatge a enviar.

Paràmetres de sortida: Cap.

Actors: Usuari.

Precondició: L'usuari ha iniciat sessió i té accés al xat global.

Postcondició: L'usuari pot enviar i rebre missatges en el xat global de l'aplicació.

Procés normal principal:

1. L'usuari t'escriu el missatge a enviar
2. Es valida que no es un missatge buit
3. L'usuari polsa el botó d'enviar per que el missatge sigui processat
4. S'emmagatzema el missatge a Firebase
5. Es mostra el missatge a la conversació global en temps real

Alternatives de procés i excepcions: cap.

Cas d'ús 18. Eliminar compte permanentment

Resum de la funcionalitat: Permet a l'usuari eliminar permanentment el seu compte de l'aplicació.

Paràmetres d'entrada: Cap.

Paràmetres de sortida: Cap.

Actors: Usuari.

Precondició: L'usuari ha iniciat sessió.

Postcondició: El compte de l'usuari s'ha eliminat permanentment de l'aplicació.

Procés normal principal:

1. L'usuari es dirigeix a la pantalla de configuració de l'aplicació
2. Clica sobre el botó d'eliminar el compte
3. Salta una alarma per tal de que l'usuari n'estigui segur i acabi acceptant
4. Si l'usuari accepta, es passa la informació a Firebase
5. Firebase s'encarrega d'eliminar tots els registres d'aquest usuari amb el corresponent ID
6. L'usuari es dirigit de nou a la pantalla principal de l'aplicació, la d'iniciar sessió.

Alternatives de procés i excepcions:

3.1. Que l'usuari cancel·li l'avís

4.1. No s'elimina el compte i no es segueix amb el procés

Cas d'ús 19. Tancar Sessió

Resum de la funcionalitat: Permet a l'usuari tancar la sessió actual i sortir de l'aplicació.

Paràmetres d'entrada: Cap.

Paràmetres de sortida: Cap.

Actors: Usuari.

Precondició: L'usuari ha iniciat sessió.

Postcondició: L'usuari ha tancat la sessió i ha sortit de l'aplicació.

Procés normal principal:

1. L'usuari es dirigeix a la pantalla de configuració de l'aplicació
2. Clica sobre el botó de tancar sessió
3. Es passa la consulta a Firebase
4. Firebase s'encarrega de tancar la sessió d'aquest usuari per tal de que s'hagi de tornar a autenticar.
5. L'usuari es dirigit a la pantalla d'iniciar sessió de nou

Alternatives de procés i excepcions: cap.

Cas d'ús 20. Perfil usuari i edició

Resum de la funcionalitat: L'usuari pot escollir si entrar a la visualització del seu perfil..

Paràmetres d'entrada: cap.

Paràmetres de sortida: cap.

Actors: Usuari.

Precondició: cap.

Postcondició: cap.

Procés normal principal:

1. L'usuari tria la vista del perfil.
2. Es dirigeix a la vista del perfil.
3. Firebase connecta amb Firestore i mostra les dades recollides del usuari als camps corresponents

Alternatives de procés i excepcions:

- 3a. Que no es puguin llegir les dades correctament i l'aplicació no mostri res.
- 3b. Firebase mostra un missatge per consola.

4 Entitats del projecte

Un cop acabada l'explicació dels casos d'ús de l'aplicació, en aquest punt m'agradaria parlar sobre les entitats del projecte en qüestió.

Les entitats són les classes principals que representen els objectes importants en el projecte, perquè com he comentat, utilitzem una arquitectura de MVC (Model Vista Controlador), aleshores són les peces clau del nostre model de dades i són essencials per al funcionament d'aquest.

En la carpeta de "Entities", tenim totes les classes que representen les diferents entitats de la nostra aplicació. Aquestes classes defineixen les propietats i els comportaments específics de cada entitat i són utilitzades per interactuar amb les dades del sistema.

La organització adequada de les entitats en la carpeta "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.

4.1 Diagrama de classes d'entitats sense operacions

El diagrama que podem observar a continuació, és el diagrama de classes sense operacions del nostre projecte. Ens permet visualitzar les relacions existents entre les classes que hi han.

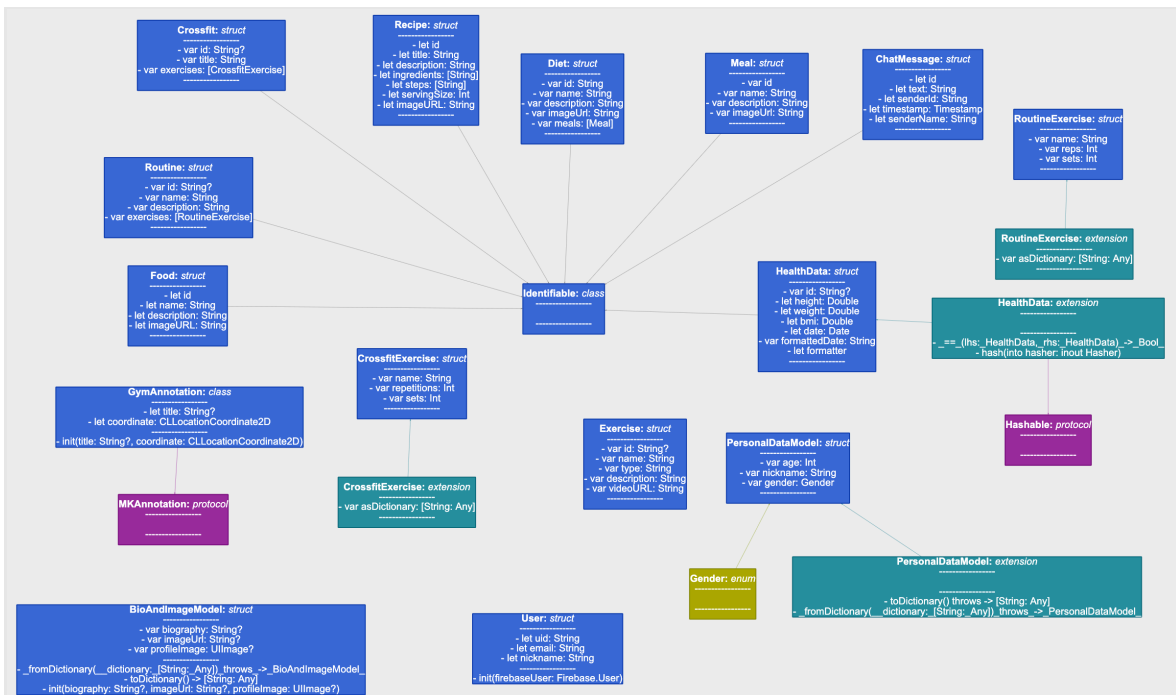


Diagrama 2. Diagrama de classes d'entitats sense operacions

Podem observar les associacions, herències i altres tipus de relacions entre les classes. A més, podem veure les classes que implementen el protocol Identifiable. Aquest protocol ens proporciona una manera de distingir i identificar les instàncies de les classes. Les classes que implementen aquest protocol tenen una propietat d'identificador únic que ens permet accedir-hi i manipular-les de manera eficient, així també cada objecte de la classe té un document únic (UUID) a la nostra base de dades de Firebase.

5 Disseny

Un cop acabat les fases de recollides de requisits, diagrames de seqüències i diagrama de les diferents classes podem començar amb el disseny de l'aplicació. El disseny de l'aplicació es basarà en tres components principals:

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

5.1 Disseny de l'arquitectura Software

Totes les aplicacions necessiten una arquitectura software completament descrita a l'inici del desenvolupament. Actualment, el desenvolupament d'aplicacions mòbils s'ha convertit en una activitat essencial a la indústria de la tecnologia. La demanda creixent d'aplicacions mòbils ha portat a la necessitat de produir aplicacions que siguin d'alta qualitat, escalables i fàcils de mantenir. Per aconseguir aquests objectius, cal una arquitectura de programari ben dissenyada i planificada.

L'arquitectura d'una aplicació mòbil és el conjunt de decisions sobre com organitzar un programa. Això inclou elements i interfícies estructurals, relacions entre elements seleccionats i estil general del programa. En resum, l'arquitectura d'una aplicació mòbil defineix l'estructura i el comportament del sistema.

Una bona arquitectura significa simplicitat i eficiència. Una arquitectura ben dissenyada permet als desenvolupadors 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, entre els quals hi ha:

- *Eficiència*: Una aplicació amb una arquitectura ben dissenyada pot fer tasques i funcions en qualsevol condició, és a dir, el sistema és productiu i fiable.
- *Flexibilitat*: La solució escollida és fàcil de canviar i hi ha menys errors. Això permet canviar un element sense que això sigui fatal per a altres components.
- *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-hi canvis més eficientment.
- *Reutilització del codi*: Un bon disseny de software promou la reutilització del codi, el que significa que s'han creat mòduls que es poden utilitzar en diferents parts del software. Això ajuda a estalviar temps i diners en la creació de codi nou.
- *Extensibilitat*: Es poden afegir tantes funcions com calgui a l'aplicació.
- *Escalabilitat*: Un bon disseny de software és escalable, el que significa que es pot adaptar i ampliar per satisfer les necessitats en el futur. Això és important per a les empreses que volen créixer i ampliar el seu negoci.
- *Testabilitat*: L'aplicació es pot provar fàcilment, cosa que significa que el nombre d'errors disminueix i la seva confiança augmenta. Els elements o mòduls de l'aplicació són independents.
- *Comprensió*: El codi ha de ser entès per tantes persones com sigui possible si en el futur treballaran més persones a l'aplicació.
- *Facilita la col·laboració*: Un bon disseny de software facilita la col·laboració entre els membres de l'equip de desenvolupament. Un disseny ben estructurat i documentat ajuda a comunicar clarament les idees i les expectatives, així com a establir un conjunt de normes i pràctiques que tots els membres de l'equip poden seguir. Això ajuda a coordinar millor les tasques, a evitar errors i a assegurar que tot l'equip treballi en la mateixa direcció.

Actualment, hi ha molts patrons diferents que s'apliquen a l'arquitectura d'una aplicació mòbil. Un és el patró Model-Vista-Controlador (MVC), que divideix l'arquitectura en tres parts: Model, Vista i Controlador. El MVC es pot implementar de dues maneres: controlador supervisor i visualització passiva.

Vaig decidir que el disseny de software utilitzat per crear l'aplicació i l'arquitectura utilitzada seria el Model-Vista-Controlador (MVC) i al disseny vaig seguir l'article del portal web SwiftBeta [6] per definir l'arquitectura software següent:

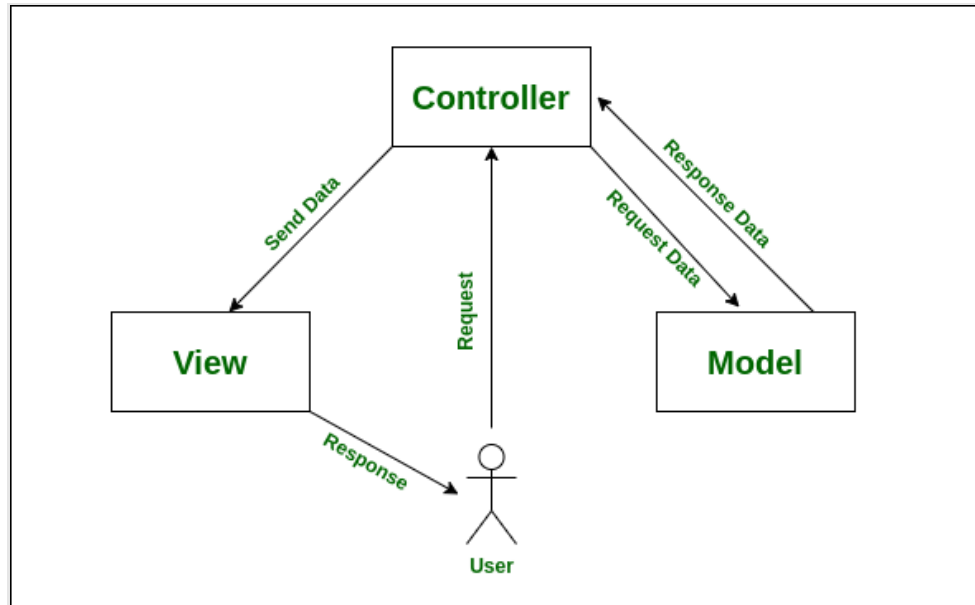


Diagrama 3. MVC

Si observem el *diagrama 3*, veurem que hi ha 3 responsabilitats clarament separades i connectades:

- *Model*: dades que s'utilitzen per obtenir la informació que es vol representar en les vistes de la nostra aplicació.
- *View*: són una representació visual del model que volem mostrar.
- *Controller*: és el mediador, és el que connecta la View amb el Model. Quan rebem una acció de la View, aquesta acció la rep el Controller. Quan aquest element rep l'acció, el Controller s'encarrega de controlar i decidir quina lògica executar.

El controlador crea la vista i el model, i a la vegada fa de mediador entre aquestes dues capes. Mai el model es comunica amb la vista directament (i viceversa), tot passa pel View Controller. Per això el controlador és la peça més important i també és la menys reutilitzable.

5.2 Disseny de la interfície

Dissenyar una interfície d'aplicació eficaç és crucial per proporcionar una experiència d'usuari satisfactòria i fer que l'aplicació sigui fàcil d'utilitzar i entenedora per als usuaris.

Els punts més importants a l'hora de dissenyar una interfície d'aplicació són:

- *Intuitivitat:* La interfície d'usuari ha de ser intuïtiva perquè els usuaris puguin fer ús de l'aplicació sense necessitat de consultar manuals o guies. Això significa que el disseny ha de ser senzill i coherent.
- *Consistència:* L'aplicació ha de tenir una aparença i comportament consistent en tots els seus elements per evitar la confusió dels usuaris i facilitar la seva navegació per l'aplicació.
- *Accessibilitat:* L'aplicació ha de ser accessible per a tots els usuaris, independentment de les seves habilitats o limitacions. Això significa que el disseny ha de ser accessible per a persones amb discapacitats visuals, auditives o motrius.
- *Estètica:* L'aparença visual de l'aplicació també és important. Els usuaris són més propensos a fer servir una aplicació que sigui atractiva i estèticament agradable.
- *Funcionalitat:* L'aplicació ha de ser funcional i complir les expectatives dels usuaris. Això significa que ha de ser capaç d'executar les tasques per les quals ha estat dissenyada de manera eficient.

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 de l'usuari. Això també pot portar a una millor retenció d'usuaris i a una millor imatge de marca per a l'aplicació.

Altres punts importants a considerar en el disseny d'interfícies d'aplicacions són:

- *Priorització del contingut:* L'aplicació ha de mostrar el contingut més rellevant i rellevant de manera clara i destacada per als usuaris.
- *Facilitat de navegació:* L'aplicació ha de proporcionar una navegació clara i senzilla per als usuaris perquè puguin accedir fàcilment a totes les funcions de l'aplicació.
- *Orientació:* L'aplicació ha de guiar als usuaris a través de la seva experiència i proporcionar-los informació útil perquè puguin completar les tasques que estan intentant realitzar.
- *Flexibilitat:* L'aplicació ha de ser flexible per adaptar-se a diferents dispositius i pantalles, així com a les preferències de l'usuari.
- *Actualitzacions:* El disseny de la interfície ha de ser flexible per adaptar-se a les actualitzacions i canvis futurs de l'aplicació perquè no sigui necessari canviar tot el disseny cada vegada que es duu a terme una actualització.

De manera general, el disseny d'interfícies d'aplicacions eficaces pot millorar significativament la satisfacció de l'usuari i la seva experiència amb l'aplicació, així com la seva percepció de la marca i la seva lleialtat a l'aplicació, ja que aquestes dues últimes coses són les més importants per el propietari de l'aplicació (jo mateix en aquest cas).

Tot seguit detallaré algunes de les regles més importants que he fet servir per dur a terme la interfície de la meva aplicació:

1. Mantenir la coherència: Totes les pantalles de l'aplicació han de mantenir la mateixa estructura visual per ajudar als usuaris a sentir-se còmodes i conèixer com funciona l'aplicació.
2. Prioritzar la simplicitat: Les aplicacions senzilles són més fàcils d'utilitzar i entenedores per als usuaris, així que sempre s'ha de buscar la simplicitat en la seva estructura i disseny.
3. Utilitzar icones i índexs clars: Les icones i els índexs han de ser clars i fàcils d'entendre, i els seus significats han de ser evidents per als usuaris.
4. Utilitzar el color de forma efectiva: El color pot ser utilitzat per destacar elements importants, crear jerarquia visual i guiar als usuaris a través de l'aplicació.
5. Utilitzar fonts fàcils de llegir: Les fonts han de ser fàcils de llegir, i ha d'evitar-se el text petit o confús.
6. Provar i iterar: El disseny de la interfície ha de ser provat amb usuaris reals per assegurar que funcioni i per identificar possibles problemes o oportunitats de millora.
7. Adaptar-se a diferents dispositius: L'aplicació ha de ser dissenyada per adaptar-se a diferents dispositius i resolucions de pantalla.
8. No utilitzar excessivament animacions o efectes visuals: Les animacions i efectes visuals poden ser distractors i disminuir la usabilitat de l'aplicació, així que s'han d'utilitzar amb moderació.
9. Utilitzar una jerarquia visual clara: Els elements més importants de l'aplicació han de ser destacats visualment per als usuaris, per ajudar-los a comprendre la seva estructura i utilitat.

Seguint aquestes regles de disseny es com he dut a terme el meu projecte, creant així una interfície d'aplicació efectiva que proporcioni als usuaris una experiència d'usuari excel·lent.

Quan vaig decidir que volia fer una aplicació de fitness com a projecte de treball final de grau vaig definir mentalment un primer dibuix d'aplicació amb unes funcionalitats bàsiques de tal manera que el desenvolupament i el treball continu de definició em portes a esbrinar quines altres necessitats podria tenir el futur usuari d'una aplicació d'aquesta tipologia. Tant bon punt va ser així i l'aplicació la vaig anar farcint d'altres finalitats i utilitats que estic segur que donaran respostes als possibles usuaris d'aquesta eina.

5.2.1 SplashScreen

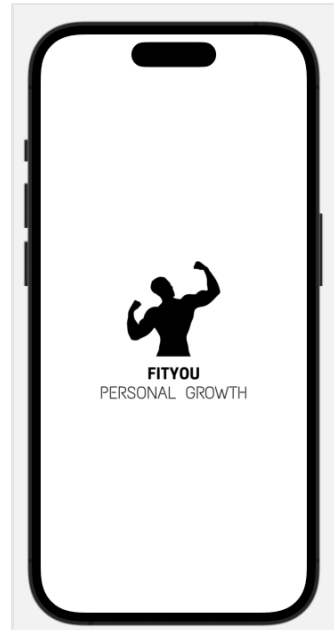


Figura 10. Pantalla Splash Screen

A la figura 10 podem veure la pantalla SplashScreen, es la pantalla que apareix durant uns segons quan s'inicia l'aplicació, abans que l'usuari pugui interactuar amb ella. Aquesta pantalla podem observar com ens mostra el logotip de l'aplicació (amb una petita animació que no es pot apreciar en una imatge), mentre que l'aplicació es carrega en segon pla. L'objectiu del SplashScreen és proporcionar una sensació de professionalitat i marca a l'aplicació, així com mantenir l'usuari informat que l'aplicació s'està iniciant i que no hi ha cap problema de connexió o de càrrega de dades.

5.2.2 Login, Register, Forgot Password i UserData

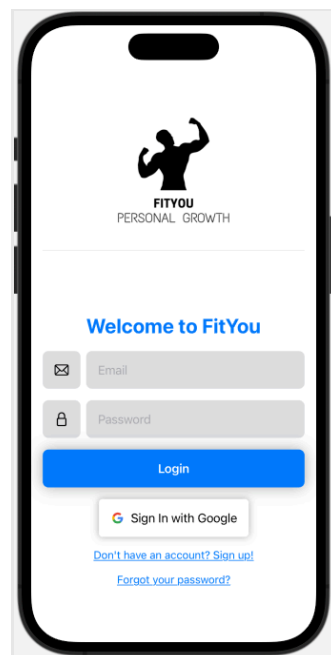


Figura 11. Inici aplicació

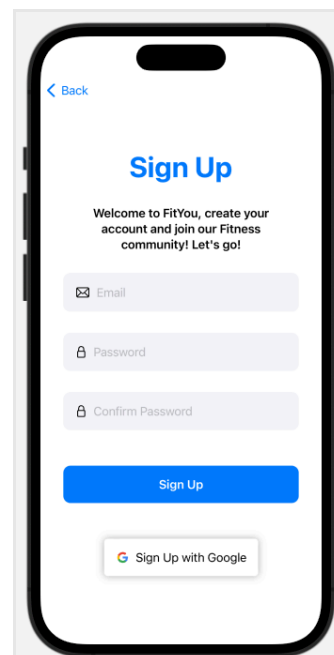


Figura 12. Pantalla Sign up

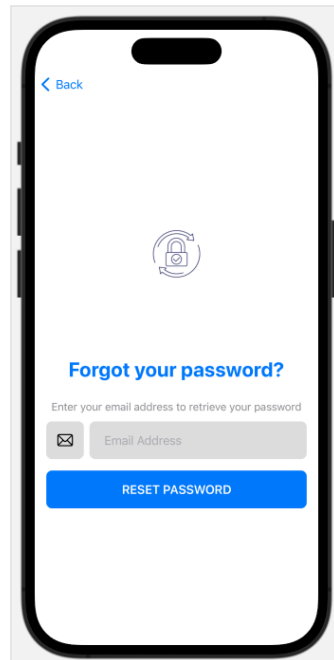


Figura 13. Pantalla canvi contrasenya

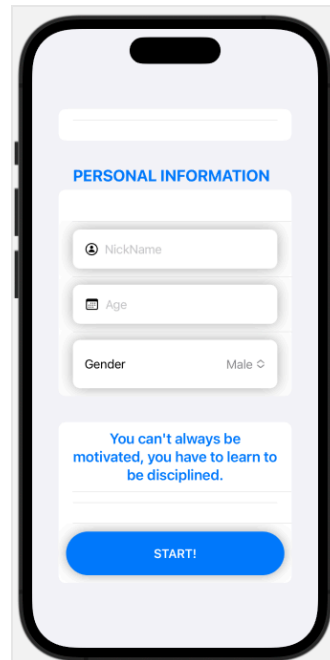


Figura 14. Pantalla informació personal

A la *figura 11* es pot veure la pantalla inicial de l'aplicació, la pantalla d'iniciar sessió. A la part superior es pot veure el logotip de la pròpia aplicació. Com he comentat en punts anteriors, concretament als requisits funcionals, l'usuari es pot autenticar mitjançant email/contrasenya o bé per Google, per això podem observar que hi ha dos camps d'entrada per les credencials corresponents i just baix del botó de "Login" tenim el botó encarregat per iniciar sessió amb Google, el qual ens dirigeix a la pantalla de Google per escollir el nostre compte. Es pot apreciar en la mateixa pantalla dos línies de text subratllades i de color blau, com el seu nom indica cada una ens redirigeix en una pantalla en concret.

A la *figura 12* podem veure la pantalla de registre de l'aplicació per aquests usuaris que són nous i no tenen cap compte per poder-ne fer-ne ús. Podem apreciar també un botó per donar-nos d'alta mitjançant Google que funciona exactament igual que el de la pantalla d'iniciar sessió. En aquesta pantalla hem pogut accedir mitjançant el primer link que teníem a la pantalla de "Login", tal i com indica el seu text.

A la *figura 13* podem apreciar la pantalla per canviar la nostra contrasenya en el cas de que ens haguem oblidat d'ella. El funcionament d'aquesta pantalla es molt senzill, quan l'usuari proporciona el seu correu electrònic (ha d'estar registrat si o si), gràcies a la utilització de l'API de Firebase, li arriba un correu electrònic al usuari personalitzat (per mi) amb un link per poder-ne posar una de nova.

A la *figura 14* es pot veure la pantalla de dades de l'usuari, però aquesta pantalla només es mostra quan l'usuari es registra per primera vegada. És a dir, quan l'usuari està en la pantalla de registre i segueix endavant, passa a la pantalla de la figura 12 per així proporcionar el nickname, l'edat i el seu gènere.

5.2.3 Profile

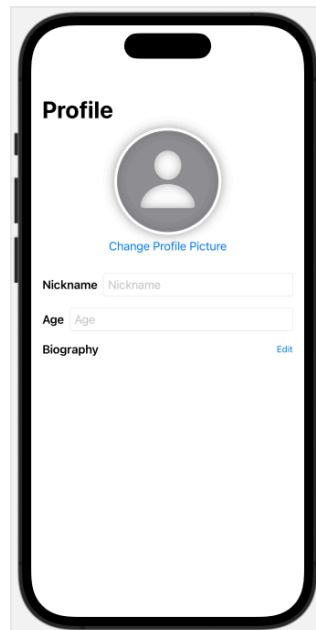


Figura 15. Pantalla perfil

A la *figura 15* podem observar la vista de perfil de l'usuari, es mostraran els camps de Nickname i l'edat de l'usuari en concret, i es pot observar com hi ha un camp anomenat "Biography", el qual l'usuari podrà editar per posar qualsevol text. Veiem que també disposa d'una imatge i un botó per poder editar i canviar-la en el cas que l'usuari ho vulgui.

5.2.4 Home

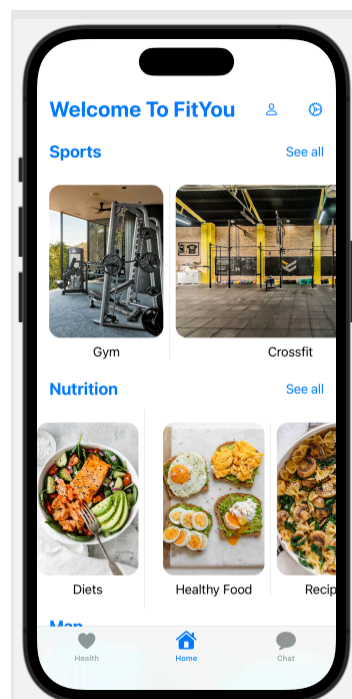


Figura 16a. Pantalla Principal

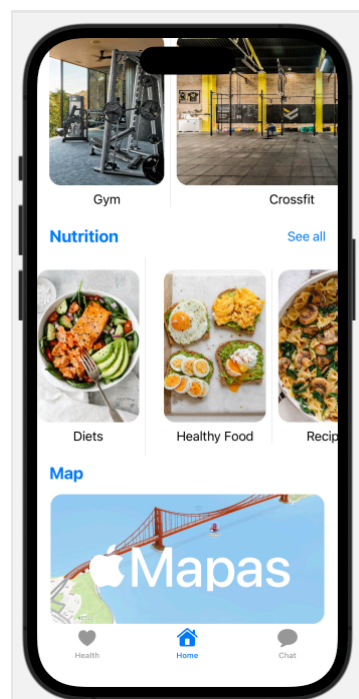


Figura 16b. Pantalla principal

A la *figura 16a* i *figura 16b* podem apreciar la pantalla principal que visualitzarà l'usuari al 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, és a dir, ens serveix de menú de navegació. Es pot observar clarament com a la part inferior, disposa d'una barra de navegació. També tenim disponible a dalt a la dreta dos botons en forma de persona i d'engrenatge, que són els encarregats de dirigir cap a la vista de perfil o de configuració.

Podem apreciar com en les seccions tant d'esports com de nutrició tenim un petit text a la dreta del títol que posa “*See all*”, es un botó que ens serveix per redirigir-nos a les pantalles de cada secció per poder veure així tots els diferents departaments que hi ha a les seccions, l'escollida en concret.

5.2.5 Gym

En aquesta secció del gimnàs que podem observar a la *figura 17* trobarem tres pantalles diferents que us explicaré a continuació. En primer lloc, tindrem la pantalla d'exercicis, on podem trobar una gran varietat d'exercicis per a diferents parts del cos i amb diferents nivells de dificultat. Així l'usuari podrà triar els exercicis que millor s'adaptin a les seves necessitats i objectius de fitness.

En segon lloc, tindrem la pantalla de rutines, on l'usuari podrà crear i guardar les seves pròpies rutines de forma personalitzada, apart de poder visualitzar les que vinguin per defecte.

Finalment, tindrem la pantalla de preferides de cadascuna, on l'usuari podrà guardar les seves rutines i exercicis preferits per a un accés ràpid i fàcil en qualsevol moment. Això li permetrà estar més organitzat i motivat per seguir amb la seva rutina de fitness. Amb aquestes tres pantalles, l'usuari tindrà tot el que necessita, i és d'aquesta manera que s'organitza la secció de gimnàs.

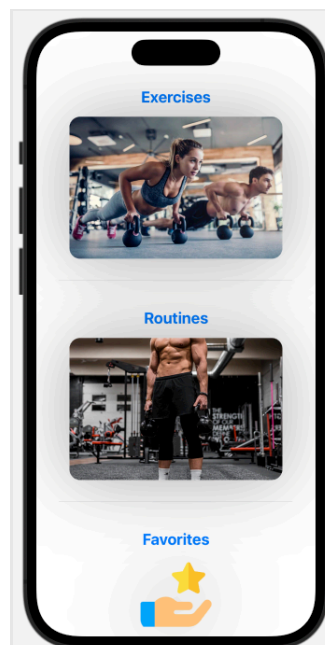


Figura 17. Pantalla Gym

5.2.5.1 Exercises

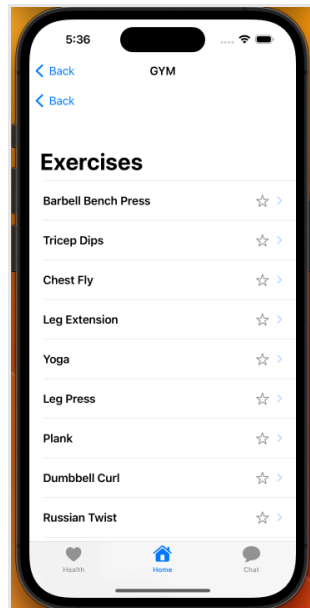


Figura 18. Pantalla llista exercicis

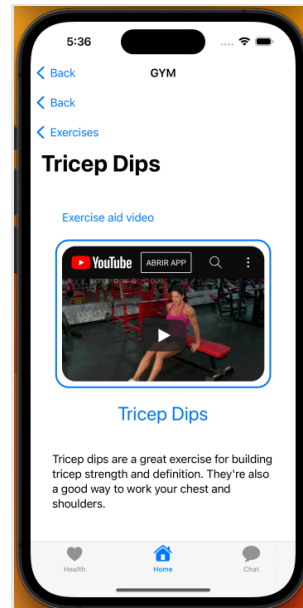


Figura 19. Pantalla detall exercici

A la figura 18, tenim la vista principal de la llista d'exercicis proposats per la mateixa aplicació, podem observar que se'ns mostra el títol de l'exercici i a la dreta del tot trobem el símbol d'una estrella, que fa referència als exercicis preferits de l'usuari, aleshores quan aquest la cliqui, aquesta es posarà en groc i s'emmagatzemarà a la vista de preferits.

A la figura 19, podem observar la vista detallada de l'exercici, la qual ens mostra el títol de l'exercici, una breu descripció del mateix i al centre de la vista podem observar com tenim un vídeo extret de Youtube, el qual ens mostrarà la realització correcta de l'exercici.

5.2.5.2 Routines

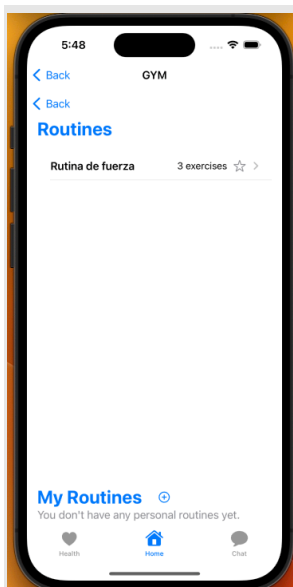


Figura 20. Llista rutines.

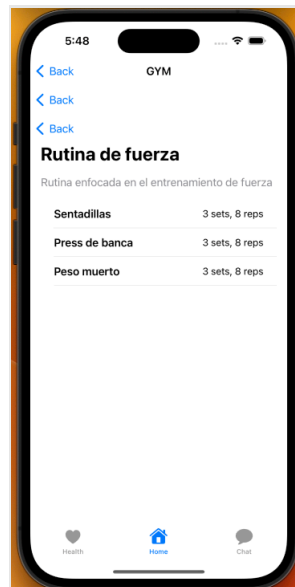


Figura 21. Vista detall rutina

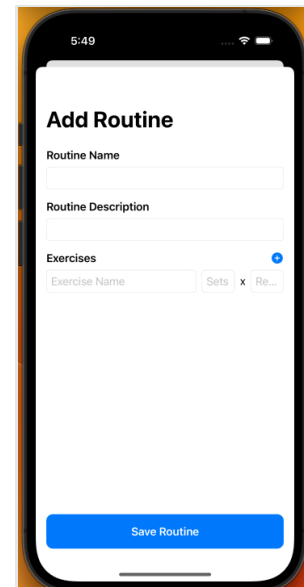


Figura 22. Pantalla adició rutina

A la *figura 20* de la nostra aplicació, podem veure la llista de rutines disponibles, en aquest cas només n'observem una perquè encara hem de carregar la base de dades de les rutines. En aquesta vista, també hi ha un botó de preferits que funciona exactament igual que en la llista d'exercicis. A sota de la llista de rutines, també trobem una secció on es mostren les rutines guardades de l'usuari. Aquesta secció inclou un botó per afegir noves rutines a la llista.

A la *figura 21*, podem veure la vista de detall de la rutina seleccionada. Aquesta vista inclou el títol de la rutina, la descripció detallada i la llista d'exercicis que s'han de realitzar en aquesta rutina.

Finalment, a la *figura 22* es mostra la vista que s'obre quan l'usuari vol crear una nova rutina. En aquesta vista, l'usuari pot afegir un títol, una descripció i seleccionar els exercicis que desitgi incloure en la nova rutina amb les respectives sèries i repeticions.

En resum, amb aquestes tres vistes diferents, els usuaris podran navegar per les rutines disponibles, afegir-ne de noves, i guardar les seves rutines preferides per a un accés ràpid i fàcil.

5.2.5.3 Favorites

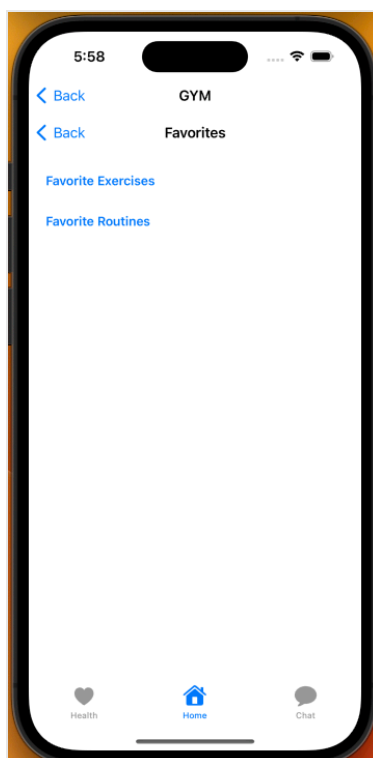


Figura 23. Llista preferits (buida)

A la *figura 23*, podem apreciar la vista d'exercicis i rutines preferides de l'usuari, es mostrarien tal com hem apreciat les llistes de rutines i exercicis, aquest cas amb l'estrella de color groc simbolitzant que és preferida i també en clicar al damunt ens redirigiria a la pantalla de detall.

5.2.6 Crossfit

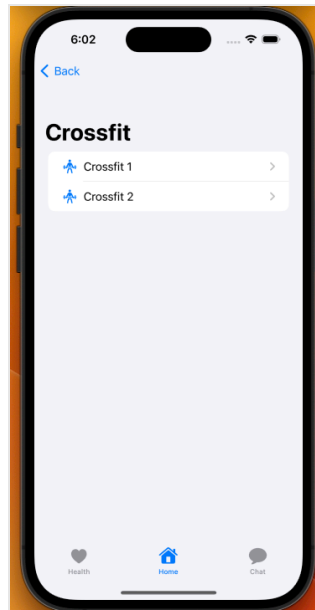


Figura 24. Llista Crossfit

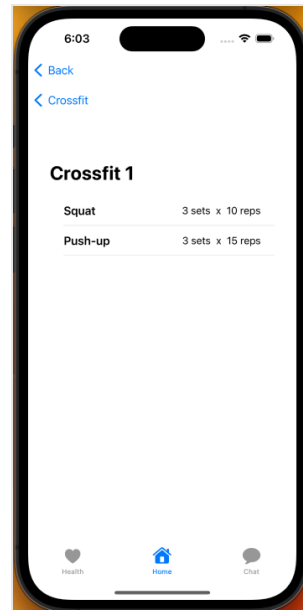


Figura 25. Vista detall crossfit

A les figures 24 i 25 podem observar les pantalles d'esport que tenen a veure amb la selecció del Crossfit, com podem observar primer es tracta d'una llista que mostra tots els diferents entrenaments o rutines d'aquest esport i a continuació la vista detallada d'aquesta amb tots els detalls que corresponguin.

5.2.7 Diets, Healthy Food & Recipes

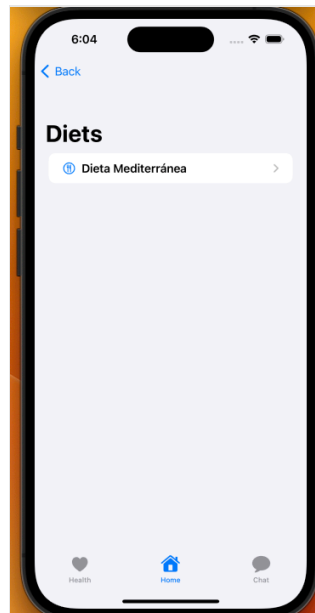


Figura 26. Llista dietes



Figura 27. Vista detall dieta

A les figures 26 i 27 podem trobar dins de l'apartat de nutrició, doncs l'apartat de les dietes, on tornem a observar que la dinàmica de les vistes es sempre igual, una llista principal amb una vista detall que ens dona tota la informació necessària.

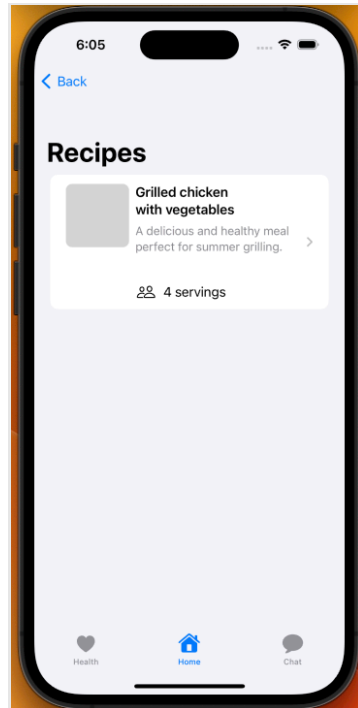


Figura 28. Llista receptes

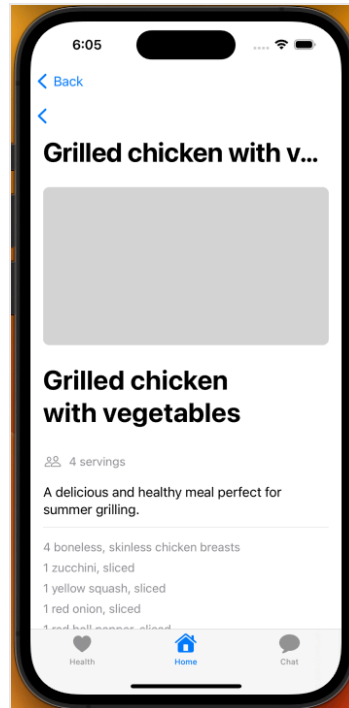


Figura 29. Vista detall recepta

En les *figures 28 i 29* podem observar les vistes de la secció de les receptes, podem veure que en la llista ja se'ns mostra una mica més d'informació, tenim una petita imatge del plat a realitzar, la quantitat de persones per la qual es la recepta i el títol de la mateixa. En la vista detall ja tenim la imatge en gran, el títol, els ingredients amb un text més gris i a baix de la pantalla les instruccions i els passos per poder realitzar la recepta.

5.2.8 Global Chat

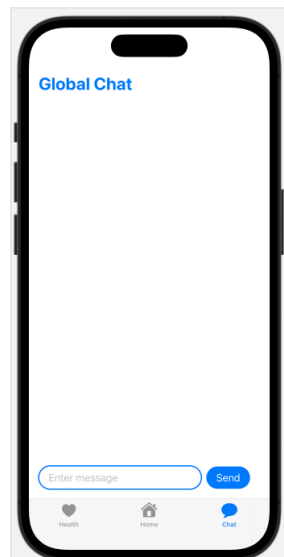


Figura 30. Vista global xat

A la *figura 30*, apreciem la vista del xat global de l'aplicació, podem observar que es una estètica senzilla, que segueix amb la mateixa línia de color de l'aplicació, el blau. Podem

observar el títol de la pantalla a dalt a l'esquerra, el requadre on l'usuari escriurà el text a enviar esta emmarcat en blau, i el botó d'enviar el missatge al costat a la dreta.

5.2.8 Health

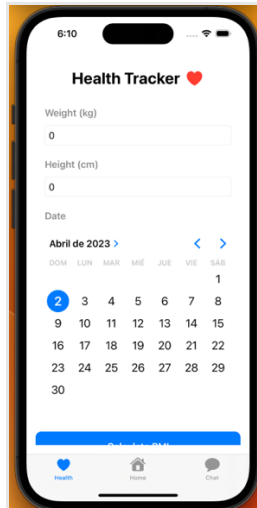


Figura 31. Vista Salut

A la figura 31 podem veure la pantalla que te a veure amb la salut del usuari, és una pantalla simple, on l'usuari podrà calcular el seu IBM, incorporant el seu pes i la seva alçada, marcant el dia del mes en el que es troba per saber-ho, i el botó mes a baix encarregat de realitzar l'operació, que el resultat surt al fons de la pantalla amb una petita descripció, per saber si l'usuari es troba dins d'un bon rang amb les dades que ell mateix ha introduït.

En aquest cas, l'usuari quan enviï un missatge, es mostrarà el nickname del usuari que hagi enviat el missatge, i es col·locarà a dalt de tot de la pantalla, és a dir, els missatges aniran ordenats de manera descendent, el mes nou sempre es trobarà l'últim missatge. Si l'usuari estigues interessat en llegir algun missatge antic hauria de lliscar tota la pantalla cap amunt per poder recuperar tots els missatges antics. (Funcionament igual als xats de la majoria d'aplicacions)

5.2.9 Map

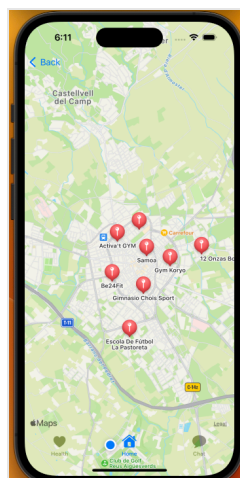


Figura 32. Vista mapa amb ubicacions

A la figura 32 se'ns mostra la pantalla de mapa de l'aplicació, en aquesta pantalla l'usuari serà capaç de poder identificar i visualitzar tots els gimnasos que tingui a prop seu, ja que la seva pròpia ubicació surt marcada en blau al mapa (funcionament exactament igual del mapa de Google maps)

Abans de que l'usuari pugui visualitzar els diferents pins que senyalen els gimnasos a prop seu, haurà de permetre que l'aplicació pugui recollir la seva ubicació i utilitzar-la en aquesta pantalla, és a dir, li sortirà un avís al usuari quan entri dins d'aquesta pantalla, es necessiten permisos per poder dur a terme les diferents localitzacions i la utilització del mapa.

5.2.10 Settings

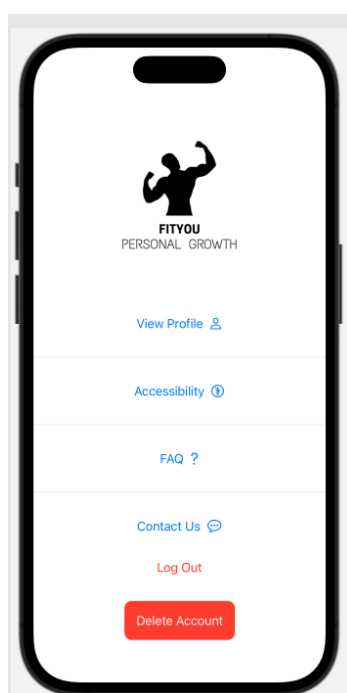


Figura 33. Pantalla configuració

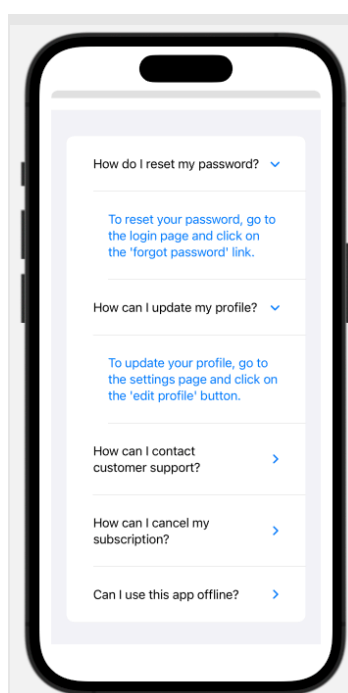


Figura 34. Pantalla FAQs

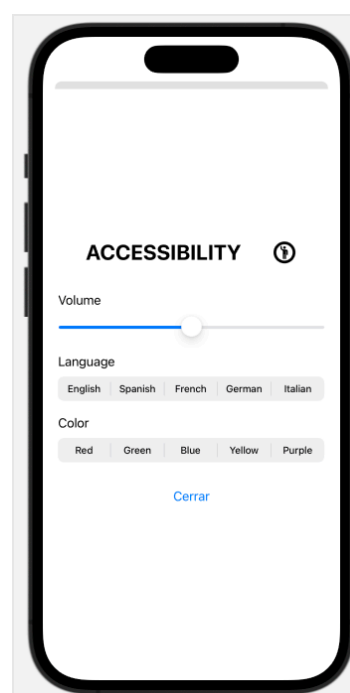


Figura 35. Pantalla accessibilitat

A la figura 33 podem observar el menú principal de la pantalla de configuració de l'aplicació. Es pot observar que es tracta d'una pantalla simple, la icona de la mateixa aplicació de capçalera. Aleshores tenim totes les diferents pantalles que ens proporciona la mateixa pantalla de configuració com pot ser la del perfil d'usuari, prèviament explicada a la figura 13, també disposem d'una pantalla de preguntes freqüents, per si l'usuari té algun dubte sobre l'aplicació. Podem extreure'n també la pantalla d'accessibilitat, per contactar amb nosaltres, el botó de tancar sessió i per acabar, l'usuari a la pantalla de configuració disposa d'un botó per eliminar permanentment la seva conta, aquest botó ja es pot apreciar que està més remarcat per donar-li més importància.

A la figura 34 veiem la pantalla de les preguntes freqüents, veiem que aquesta pantalla només tracta de certes preguntes amb la seva pròpia resposta, la qual està amagada, per poder-la visualitzar, l'usuari haurà de fer ús de la fletxa que hi ha a la dreta de cada pregunta per que se li obri el desplegable amb la resposta a aquella pregunta.

A la *figura 35* veiem la pantalla de accessibilitat, amb unes petites opcions d'usuari, per canviar el color, l'idioma i el volum de l'aplicació, que més que res són de decoració, ja que no està implementada la lògica.

5.3 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.

Finalment, vaig decidir utilitzar la Firebase. Firebase és un ecosistema d'aplicacions basades en el núvol i que inclou una Firebase API per a la persistència de dades i com a "BackEnd" per a l'aplicació destinada al cloud. Firebase és un servei web que inclou la gestió d'una base de dades en temps real basada en el núvol, que ofereix una gran escalabilitat, una gestió eficient de la informació i una alta disponibilitat. Això vol dir que es pot adaptar fàcilment a les necessitats del meu projecte, independentment de si augmenta el volum de dades o el nombre d'usuaris.

A més, Firebase ofereix una gran varietat de funcionalitats, com ara l'autenticació d'usuaris, l'emmagatzematge de fitxers, l'enviament de notificacions i l'analítica de dades. Això fa que sigui una opció molt completa i útil per a la meva aplicació, i em dona una gran flexibilitat per a futurs desenvolupaments.

En resum, la decisió de triar Firebase com a base de dades i "BackEnd" per al meu projecte va ser una decisió molt encertada. La seva escalabilitat, gestió eficient de la informació, alta disponibilitat i funcionalitats avançades, la fan una opció completa i útil per a les necessitats del meu projecte.

5.3.1 Avantatges de bases de dades NoSQL

Les bases de dades NoSQL, com Firebase, presenten diversos avantatges en comparació amb les bases de dades relacionals tradicionals, conegudes com a SQL. A continuació, enumeraré alguns dels avantatges més destacats de les bases de dades NoSQL, amb l'ajuda de la pàgina GeeksForGeeks [7] hem pogut extreure:

- *Escalabilitat:* Les bases de dades NoSQL són altament escalables, cosa que significa que es poden adaptar fàcilment a l'increment de la càrrega de treball i la quantitat de dades. Això és possible gràcies a la seva estructura distribuïda, que permet afegir nodes a mesura que es necessitin.
- *Flexibilitat:* A diferència de les bases de dades relacionals, les bases de dades NoSQL no utilitzen una estructura de dades fixa, sinó que permeten la creació de models de dades flexibles que s'adapten a les necessitats dels desenvolupadors. Això permet als desenvolupadors modificar el model de dades sense haver de canviar l'estructura de la base de dades.
- *Alta disponibilitat:* Les bases de dades NoSQL estan dissenyades per ser altament disponibles, amb un temps de parada mínim en cas de fallada del sistema. Això es deu a la seva estructura distribuïda, que permet que les dades estiguin replicades en

diversos nodes, de manera que si un falla, els altres nodes poden continuar processant dades.

- *Rendiment:* Les bases de dades NoSQL solen tenir un rendiment millor que les bases de dades relacionals, especialment quan es tracta de processar grans volums de dades en temps real. Això es deu a la seva estructura de dades optimitzada per a consultes i operacions de lectura i escriptura en temps real.
- *Cost:* Les bases de dades NoSQL són més econòmiques que les bases de dades relacionals, especialment per a les petites i mitjanes empreses, ja que no requereixen llicències costoses ni la necessitat d'invertir en infraestructures de servidors dedicats.

En resum, les bases de dades NoSQL com Firebase presenten una sèrie d'avantatges que les fan una opció atractiva per a molts desenvolupadors, especialment en aplicacions web i mòbils que necessiten escalabilitat, flexibilitat, alta disponibilitat, rendiment i costos reduïts.

5.3.2 Avantatges de bases de dades en temps real

Les bases de dades en temps real, com Firestore de Firebase, que és la que he utilitzat en el meu projecte presenten nombrosos avantatges que les fan una opció ideal per a aplicacions que requereixen dades en temps real.

Per aquest motiu he escollit Firestore, la base de dades en temps real de Firebase, és una elecció lògica i convenient. Firestore té molts avantatges que fan que sigui una opció atractiva per als desenvolupadors, entre les quals es destaquen les següents:

- *Dades actualitzades en temps real:* Firestore actualitza les dades en temps real, la qual cosa significa que els usuaris veuen canvis immediats en l'aplicació sense necessitat de refrescar la pàgina. Això proporciona una millor experiència de l'usuari i augmenta la interactivitat de l'aplicació.
- *Escalabilitat:* Firestore és altament escalable, cosa que significa que es pot adaptar a l'increment de la càrrega de treball i la quantitat de dades. Això és possible gràcies a la seva estructura distribuïda, que permet afegir nodes a mesura que es necessitin.
- *Integritat de les dades:* Firestore manté la integritat de les dades, evitant la duplicació i la inconsistència. A més, proporciona una gestió de transaccions segura per assegurar que les operacions de lectura i escriptura siguin atòmiques i coherent.
- *Fàcil d'usar:* Firestore és fàcil d'integrar amb altres serveis de Firebase i altres tecnologies. La seva API és intuïtiva i fàcil d'usar, cosa que facilita la implementació per als desenvolupadors.
- *Disponibilitat:* Firestore és altament disponible, amb un temps de parada mínim en cas de fallada del sistema. Això es deu a la seva estructura distribuïda, que permet que les dades estiguin replicades en diversos nodes, de manera que si un falla, els altres nodes poden continuar processant dades.

Tanmateix, significa que l'aplicació que he realitzat amb la utilització de Firestore podrà oferir una millor experiència d'usuari, ja que les dades es mostren de

manera immediata, sense la necessitat d'esperar a la recàrrega de la pàgina. A més, Firestore ofereix un ampli ventall de funcions per a la gestió de dades en temps real, com ara la sincronització de dades entre dispositius i la integració amb altres serveis de Firebase.

5.3.3 Disseny de l'arquitectura software de la base de dades

Com s'ha descrit anteriorment, he emprat Firebase Firestore com a base de dades del meu projecte. Totes les dades es guarden en forma de documents que estan en formats JSON.

JSON (JavaScript Object Notation) és un format molt lleuger que ens permet realitzar l'intercanvi de dades entre aplicacions, destaca molt per la capacitat de representar dades estructurades.

Primer de tot, comentar que Firestore organitza totes les dades en col·leccions i documents. Una col·lecció és un grup de documents que poden tenir un tema o un conjunt de característiques comunes. Cada document és una unitat d'emmagatzematge que conté les dades en format JSON, com he comentat prèviament, així Firestore ja adopta una estructura jeràrquica. Les col·leccions, tenen diversos documents, aleshores cada document té un identificador únic dins de la col·lecció (ID), a més cada document està format per un conjunt de camps amb els seus respectius valors.

Un cop feta la descripció de les funcionalitats i característiques de Firestore, així com s'organitza, el meu projecte té les següents col·leccions per poder emmagatzemar les dades:

- *Users*

Per guardar les dades dels usuaris, existeix la col·lecció “users”, on podem trobar tots els usuaris de la nostra aplicació amb el seu propi identificador (generat automàticament). I dins de cada usuari podem trobar els diferents camps, com poden ser l'edat, el gènere i el nom d'usuari.

Afegir que dins de cada col·lecció de Firestore és possible crear altres col·leccions, això ens permet organitzar les dades de manera lògica i estructurada. Cada col·lecció, inclosa la principal, conté els seus propis documents que contenen els seus camps i valors associats.

Per això dins de la col·lecció “users” podem trobar varies subcol·leccions que totes elles depenen de cada usuari, cada id únic., com poden ser:

- *Health*

Aquesta subcol·lecció, manté un històric de totes les vegades que l'usuari ha calculat el seu índex de massa corporal i les ha guardades a la base de dades, aleshores els camps que podem trobar dins de cada document d'aquesta col·lecció són, l'alçada, el pes, la data en que s'ha calculat i el respectiu resultat del bmi.

- *PersonalRoutines*

En aquesta subcol·lecció s'emmagatzemen totes les rutines personals de cada usuari,

I el funcionament segueix sent el mateix que totes les col·leccions, documents amb un identificador únic i cada document amb els seus camps i valors associats.

- *Plans*

Aquesta subcol·lecció conté documents que representen els plans o paquets als quals un usuari s'ha subscrit o ha comprat.

- *Profile*

Aquesta subcol·lecció conté un únic document que emmagatzema les dades de la biografia del perfil de l'usuari. També emmagatzemo una imatge de perfil en aquest document, ja sigui emmagatzemant una URL d'una imatge o utilitzant Firebase Storage per emmagatzemar la imatge en si i guardar l'enllaç de descàrrega.

- *TermsAndConditions*

Aquesta subcol·lecció conté un únic document que guarda la informació sobre els termes i condicions que l'usuari ha acceptat per poder registrar-se en l'aplicació.

Després de totes aquestes col·leccions que trobem internament dins de la col·lecció “users”, tornem amb les col·leccions principals del projecte.

- *Routines*

En aquesta col·lecció s'emmagatzemen les rutines d'entrenament. Cada document conté informació com ara el nom de la rutina, la descripció, els exercicis que la componen, el nombre de repeticions o sèries, els intervals de descans, entre altres detalls rellevants.

- *Recipes*

En aquesta col·lecció es guarden les receptes de menjars. Cada document conté informació sobre els ingredients necessaris, les instruccions de preparació, les calories o macros, les restriccions dietètiques i altres detalls relacionats amb la recepta.

- *Messages*

Aquesta col·lecció guarda els missatges del xat general de l'aplicació. Cada document és un missatge, i aquest conté informació com ara el remitent, el destinatari, el contingut del missatge, la data i altres metadades relacionades amb els missatges.

- *Foods*

En aquesta col·lecció s'emmagatzemen les dades sobre els aliments.

- *Exercises*

Aquesta col·lecció conté informació sobre exercicis específics. Cada document inclou el nom de l'exercici, la descripció, i un vídeo explicatiu de la realització d'aquest.

- *Diets*

En aquesta col·lecció guardo informació sobre les dietes o plans d'alimentació. Cada document conté detalls sobre el tipus de dieta, els aliments recomanats o restringits, les rutes de menjar i altres detalls relacionats amb la dieta.

- *Crossfit*

Aquesta col·lecció emmagatzema les dades relacionades amb l'entrenament de CrossFit. Funciona de la mateixa manera que la de rutines, perquè al fi i a la cap fan el mateix funcionament.

6 Implementació

En aquesta secció s'explicarà la implementació del projecte d'una manera més detallada. Donada la complexitat de les funcionalitats que l'aplicació ofereix als usuaris, s'ha requerit desenvolupar una gran quantitat de codi. No obstant això, considero que seria excessiu i redundant copiar totes les parts del codi i comentar-les en la documentació. Els comentaris podrien arribar a ocupar diverses pàgines, fent que la documentació esdevingui innecessàriament extensa.

Tanmateix, us proporcionaré una visió general de les tecnologies i eines emprades en la implementació. Això inclourà les biblioteques i altres recursos que he estat utilitzant per desenvolupar l'aplicació.

6.1 Llibreries utilitzades

Com en molts projectes de desenvolupament d'aplicacions per a la plataforma iOS amb Xcode, aquest projecte també fa ús de llibreries que permeten implementar diverses funcionalitats, optimitzar el codi i simplificar el procés de desenvolupament.

Les llibreries són eines valuoses que proporcionen conjunts de codi i funcionalitats addicionals que es poden integrar fàcilment en l'aplicació.

Les dependències i llibreries que es fan servir en aquest projecte específic basat en Firebase, Firestore, Google Sign-In, Stripe i MapKit són les següents:

- **FirebaseAnalytics:** Aquesta és un framework de Firebase que permet la recopilació i l'anàlisi de dades d'ús de l'aplicació. Proporciona eines per comprendre el comportament dels usuaris, l'eficàcia de les funcionalitats i altres mètriques clau.
- **FirebaseAuth:** És una llibreria de Firebase que proporciona serveis d'autenticació d'usuaris. Amb aquesta llibreria, puc gestionar el registre i l'inici de sessió dels usuaris mitjançant diverses opcions d'autenticació com ara adreces de correu electrònic, números de telèfon o autenticació amb xarxes socials com Google, que és la utilitzada en el projecte.
- **GoogleSignIn:** És una llibreria de Google que s'integra amb Firebase per permetre l'autenticació dels usuaris mitjançant el seu compte de Google. Proporciona una interfície senzilla perquè els usuaris puguin iniciar sessió amb el seu compte de Google i obtenir les seves dades d'autenticació.

- **GoogleSignInSwift:** Aquesta és un framework específic per al desenvolupament en Swift que proporciona una integració més senzilla amb Google Sign-In. Simplifica la configuració i l'ús de l'autenticació de Google a l'aplicació.
- **Stripe:** Aquesta és una llibreria de pagaments en línia que permet acceptar pagaments amb targeta de crèdit i altres mètodes de pagament a través de l'aplicació. S'integra amb Firebase per facilitar la gestió de pagaments i transaccions.
- **MapKit:** És una llibreria proporcionada per Apple que permet mostrar mapes interactius a l'aplicació. Amb MapKit, pots mostrar ubicacions, calcular rutes, afegir anotacions i interactuar amb el mapa en general.
- **CoreLocation:** És una llibreria proporcionada per Apple que permet accedir als serveis de localització del dispositiu. Aquesta llibreria ofereix funcionalitats per obtenir la ubicació actual del dispositiu, monitoritzant canvis d'ubicació, calcular distàncies entre punts i proporcionar informació detallada sobre els llocs geogràfics.

La utilització d'aquestes llibreries i dependències ha estalviat temps i esforç en el desenvolupament, ja que han proporcionat funcionalitats predefinides.

6.2 Estructura del projecte

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.

De manera predeterminada, els projectes d'aplicacions per iOS en Xcode segueixen una estructura específica per organitzar el codi i els recursos. Aquesta estructura m'ha ajudat bastant a mantenir l'ordre i el seguiment del projecte.

Les classes de codi font, com he comentat al principi del treball, les he escrites amb llenguatge Swift i es col·loquen dins de la carpeta general del projecte. En canvi, els recursos de l'aplicació, com ara imatges, fitxers de configuració o fitxers de localització estan ubicats a la carpeta "Assets".

En resum, he seguit una estructura adequada i ben dissenyada per poder mantenir l'ordre en el desenvolupament del projecte i poder tenir una bona organització del codi i dels recursos utilitzats, com es pot observar a la *figura 36* aquesta ha sigut l'estructura i l'organització del meu treball.

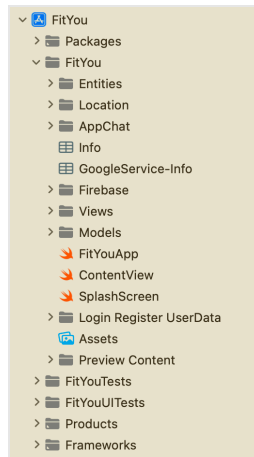


Figura 36. Estructura del projecte

En la *figura 36* es pot observar l'estructura del meu projecte, la qual he organitzat de manera lògica per facilitar el desenvolupament de l'aplicació. Algunes de les carpetes o packages més importants són:

- **Packages:** Aquesta carpeta conté les llibreries externes i les dependències del projecte que s'utilitzen per proporcionar funcionalitats addicionals.

Observem que la carpeta de Packages es troba fora de la carpeta principal del projecte, aleshores totes les següents carpetes ja es troben dins de la carpeta o package general de l'aplicació que s'anomena FitYou.

- **Entities:** Aquesta carpeta engloba totes les classes d'entitats del projecte.
- **Location:** Aquesta carpeta conté els fitxers relacionats amb la localització, i està separada de la resta de carpetes per la complexitat que té el codi de la localització i poder-ho treballar de manera més independent.
- **AppChat:** En un projecte que inclou una funcionalitat de xat, aquesta carpeta o package pot contenir tot el codi relacionat amb aquesta característica, ja que sol ser una part complexa i separada de l'aplicació principal. Exactament per el mateix motiu que la localització, és un codi ja molt més complex i facilita molt el temps poder-ho tenir en una carpeta apart.
- **Firebase:** En aquesta carpeta trobem els fitxers de configuració i altres fitxers relacionats amb l'ús de Firebase.
- **Views:** Aquesta carpeta conté tots els fitxers relacionats amb les vistes de l'aplicació. Aquesta estructura permet organitzar i localitzar fàcilment les diferents vistes de l'aplicació.
- **Models:** Aquesta carpeta conté els controladors que es comuniquen amb les vistes de l'aplicació. És una part crucial del projecte, ja que conté la lògica principal de l'aplicació.

- Carpeta de registre i inici de sessió: En alguns casos, les funcionalitats de registre i inici de sessió poden ser prou complexes i necessiten una carpeta dedicada per al seu codi, per això el fet de també tenir una carpeta a banda.
- Carpeta d'assets: Aquesta carpeta conté tots els recursos de l'aplicació, com ara imatges, fonts o fitxers de configuració.

Aquesta estructura de carpetes o packages permet una organització clara i lògica dels fitxers del Projecte i tenir una certa facilitat a l'hora de seguir amb el desenvolupament del projecte i tenir-ho tot ben guardat i organitzat.

6.3 Firebase

Per utilitzar el framework de Firebase al meu projecte, he utilitzat les llibreries de Firebase per iOS. Per accedir a les funcionalitats de Firebase, he importat les llibreries necessàries a les classes corresponents i això ho he fet mitjançant de les línies de codi que es mostren al principi de cada classe, com per exemple es pot apreciar a la *figura 37*.

```
import Foundation
import FirebaseAuth
import Firebase
import Combine
import GoogleSignIn
```

Figura 37. Llibreries Firebase (exemple)

He creat una classe anomenada “FirebaseManager”, que és la que actua com a intermediària, és a dir un wrapper, entre el projecte i les llibreries de Firebase. Aquesta classe conté mètodes per poder gestionar l'autenticació d'usuaris, crear i eliminar comptes, també funcions específiques per l'autenticació amb Google.

Aleshores, per poder utilitzar “FirebaseManager” a través de les diferents classes del projecte, he creat una classe anomenada “AuthenticationManager”, que aquesta classe és un “ObservableObject”, la qual cosa significa que emet notifikacions a través del protocol “objectWillChange”, i es la classe que pot interactuar amb la classe “FirebaseManager” i manté un estat d'autenticació. Per tant, les vistes del projecte poden accedir en aquesta classe, cada cop que les dades canvïn, ja que així ens permet controlar les dades d'autenticació de l'usuari i actualitzar les vistes corresponents quan hi hagin canvis per tal de poder actuar a temps real.

Una altra característica important de la implementació de Firebase és l'ús de *closures* i *callbacks* per gestionar les respostes asíncrones. En els mètodes de “FirebaseManager”, com ara “signin” o “createUser”, s'ha utilitzat el paràmetre *completion* de tipus *closure* per aconseguir això. Aquesta tècnica permet que el codi dins de la *closure* s'executi quan es completa la funció de Firebase, proporcionant així els resultats o errors corresponents.

A més, s'ha utilitzat la classe Auth de Firebase per poder utilitzar l'autenticació d'usuaris, aleshores, mitjançant aquesta classe, s'han utilitzat molts mètodes com poden ser “signin(withEmail: Password: completion:)” per poder accedir amb correu electrònic i

contrasenya, o altres com poden ser “createUser(withEmail: Password: completion:), que com s’aprecia, es per crear un nou usuari.

Per a l'autenticació amb Google, s'ha utilitzat la classe OAuthProvider de Firebase. A través d'aquesta classe, s'ha obtingut el proveïdor d'autenticació de Google i s'ha generat el credencial corresponent per iniciar sessió amb Google. El mètode signIn(with:completion:) de la classe Auth s'ha utilitzat per iniciar sessió amb el credencial de Google.

Per concloure, la utilització del framework Firebase en el meu projecte ha permès una gran integració eficient i robusta tant de seguretat com de facilitat (a l'hora de realitzar autenticacions, bases de dades en temps real, etc) al meu projecte.

6.4 Autenticació d'usuaris

En aquest projecte necessitem conèixer la identitat d'un usuari per poder guardar i emmagatzemar les seves dades en el núvol de forma segura per que no hi hagi cap error o cap problema. Firebase Authentication proporciona serveis de backend, un SDK fàcil i senzill d'utilitzar i biblioteques d'interfície d'usuari predefinides per poder autenticar els usuaris a l'aplicació.

Perquè un usuari pugui accedir a l'aplicació, primerament hem de recopilar les seves credencials d'autenticació. Aquestes credencials poden ser l'adreça de correu electrònic i la contrasenya d'usuari, o bé un símbol OAuth proporcionat per un proveïdor d'identitat, en aquest cas, en aquest projecte, utilitzem Google com a exemple de proveïdor. Un cop obtenim aquestes credencials, les enviem al SDK de Firebase Authentication. Després, els nostres serveis de backend verifiquen aquestes credencials i proporcionen una resposta al client. Un cop es verifiquen aquestes credencials de l'usuari, el sistema Firebase Auth retorna un identificador d'usuari únic (UID), que ens permet treballar amb les dades de l'usuari mitjançant aquest identificador únic. Com s'ha mostrat en el disseny de les bases de dades, totes les dades d'usuari es guarden associades a aquests identificadors.

Si accedim a la consola de desenvolupador de Firebase i anem a l'apartat d'autenticació, podem visualitzar tots els usuaris registrats en el sistema, tal com es mostra a la *figura 38*. A la llista, podem veure els identificadors únics d'usuari (UID), les dates de creació d'usuari i d'inici de sessió. Però el que realment ens importa es el UID a la nostra base de dades ja que el necessitem per poder treballar amb les dades específiques de cada usuari.



Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
adeu@gmail.com	📧	13 abr 2023	13 abr 2023	4umuy8kQ9o0DLQ6efALiziuvMEb2
hola@gmail.com	📧	9 abr 2023	4 may 2023	ALRh4vg95AbckLHZJ3wxRBQEge...
abril.martinlago@gmail.com	📧	9 abr 2023	9 abr 2023	P7pG0GGcYCMDI5d0J9zxe0FbHp...
hola2@gmail.com	📧	8 abr 2023	8 abr 2023	0iBr6bSFMcPeSiPpILwwK88s0J02
ramonserracruz@gmail.com	📧	20 mar 2023	20 mar 2023	nGKLAniUAzWcPTUrOqqNybN0...
gerard.serra@estudiants.ur...	📧	16 mar 2023	6 abr 2023	UKA3cHol9gYIRiyPOYH5ca0tjD3

Figura 38. Pantalla usuaris autenticats Firebase

L'autenticació d'usuari és un aspecte clau en el nostre projecte, i hi ha dos funcionalitats rellevants en aquest procés. Primerament, quan l'usuari ja està autenticat a l'aplicació, disposem d'una opció per permetre-li recuperar la seva contrasenya en cas que l'hagi oblidat o necessiti restablir-la. Això ofereix comoditat i facilita l'experiència de l'usuari, ja que no haurà de passar per un procés complex per recuperar l'accés al seu compte.

D'altra banda, aquest projecte també consta d'un subapartat que permet als usuaris eliminar permanentment el seu compte. Aquesta opció és important per donar-los control total sobre les seves dades i la possibilitat de decidir si volen romandre o no a la plataforma.

Amb aquests dos funcionalitats, proporcionem als usuaris les eines necessàries per gestionar el seu compte d'autenticació de manera eficient i segura, assegurant una experiència positiva i satisfactòria en l'aplicació.

6.4.1 Recuperació de contrasenya

En aquesta secció, s'explicarà detalladament com es pot recuperar la contrasenya, quines implicacions té aquesta acció, quins passos s'han de seguir per poder completar el procés de manera segura i quines facilitats ens proporciona Firebase per poder dur a terme aquest procés.

1. *Comunicació amb Firebase:* La funció “resetPassword()” establerta en el model, en el controlador, estableix una comunicació amb Firebase per poder enviar una sol·licitud de restabliment de contrasenya. Aquesta comunicació permet que Firebase gestioni el procés de recuperació de contrasenya per al correu electrònic proporcionat. Podem veure aquesta funció en la *figura 39*.

```
func resetPassword() {
  Auth.auth().sendPasswordReset(withEmail: email) { error in
    if error != nil {
      self.alertTitle = "Error"
      self.alertMessage = error?.localizedDescription ?? "Unknown error"
    } else {
      self.alertTitle = "Success"
      self.alertMessage = "An email has been sent to reset your password. Check your inbox."
    }
    self.showAlert = true
  }
}
```

Figura 39. Codi per resetejar contrasenya

2. *Verificació d'errors:* En cas que hi hagi algun error durant el procés de recuperació de contrasenya, Firebase enviarà un missatge d'error a l'aplicació. Aquest missatge es capturarà i es mostrarà al usuari mitjançant l'ús de les propietats “alertTitle” i “alertMessage” que podem observar al codi de la *figura 40*, que ens mostra l'error de que el correu establert no correspon a cap usuari, aleshores no es realitza cap gestió per recuperar contrasenya.

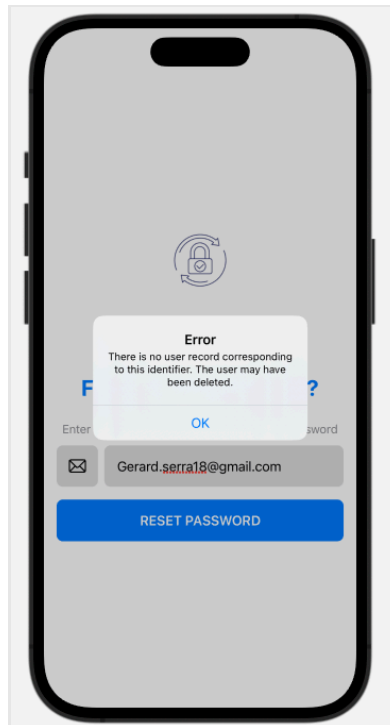


Figura 40. Verificació error

3. *Confirmació d'èxit:* En cas que el procés de recuperació de contrasenya sigui exitós Firebase envia un correu electrònic a l'usuari amb les instruccions per poder restablir la seva contrasenya. Aquesta confirmació d'èxit es torna a mostrar al usuari amb el mateix funcionament de les propietats utilitzades per informar en cas d'error, però en aquest cas, d'èxit informant que s'ha enviat un correu electrònic amb les instruccions pertinents, tal com es pot observar en la *figura 41*.

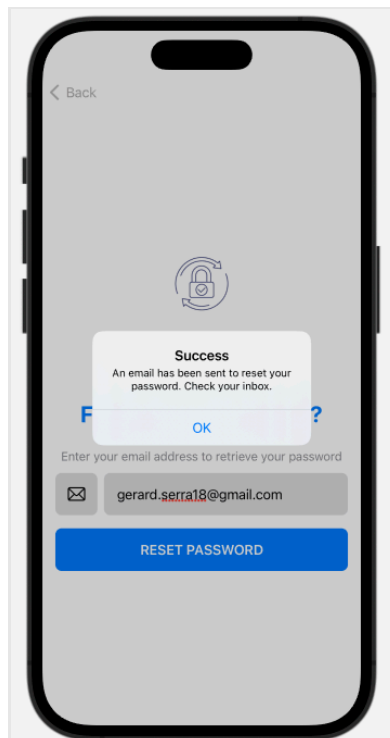


Figura 41. Verificació success

4. *Recepció del correu electrònic:* L'usuari rep el correu electrònic de Firebase a la seva bústia d'entrada, aleshores aquest correu electrònic conté informació sobre la sol·licitud de recuperació de contrasenya i un enllaç amb el text per poder dur a terme aquesta recuperació, tal com es pot observar a la *figura 42*.

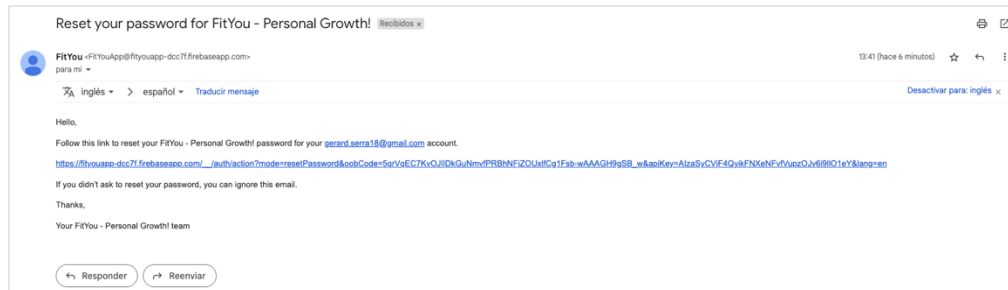


Figura 42. Pantalla correu regenerador de contrasenya

5. *Pàgina de canvi de contrasenya:* En accedir a la pàgina de canvi de contrasenya, *figura 43*, l'usuari veurà un formulari o una interfície on podrà introduir una nova contrasenya. Firebase valida les dades introduïdes i actualitza la contrasenya de l'usuari si tot es correcte.

A screenshot of a web form titled "Reset your password". Below the title, it says "for gerard.serra18@gmail.com". There is a text input field labeled "New password" with a blue underline and a toggle icon (an eye) to its right. At the bottom right of the form is a blue button labeled "SAVE".

Figura 43. Formulari contrasenya nova

6. *Confirmació de canvi de contrasenya:* Un cop l'usuari ha introduït la nova contrasenya i ha premut el botó de guardar, Firebase processa la sol·licitud i actualitza les dades de l'usuari amb la nova informació.

Aquests son els passos a dur a terme, que permeten a l'usuari recuperar i actualitzar la seva contrasenya de forma segura i senzilla mitjançant l'ús de Firebase. Aquesta funcionalitat ofereix als usuaris una forma ràpida i eficient de poder recuperar l'accés al seu compte en cas d'oblidar-se de les seves credencials.

6.4.2 Eliminació d'usuari

En aquesta secció, s'explicarà detalladament com es pot eliminar un compte en aquest projecte, quines implicacions té aquesta acció, quins passos s'han de seguir per poder-ho dur a terme i quines facilitats ens proporciona Firebase per completar el procés de manera segura i eficient.

En la vista **SettingsView**, figura 44, es presenta un botó "Delete Account" que permet a l'usuari eliminar el seu compte de l'aplicació. Aquesta funcionalitat està implementada utilitzant les funcions del ViewModel **SettingsViewModel** que interactua amb Firebase per gestionar l'eliminació del compte.

Després de clicar el botó, es produeixen tots els següents passos:

1. *Salt d'alerta*: Quan l'usuari prem el botó per dur a terme l'acció d'eliminar el seu compte permanentment, salta una alerta, figura 44, en la qual podem observar com apareixen dos botons. El botó primari (destructive) amb el text "Delete" i el botó secundari amb el text "Cancel".

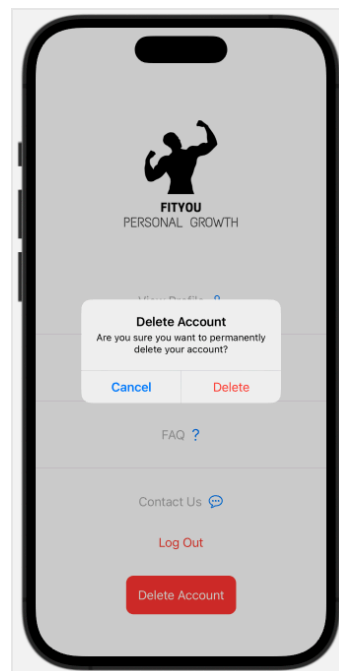


Figura 44. Pantalla alerta d'eliminació

2. *Crida a "deleteAccount()"*: Quan l'usuari prem el botó "Delete", es crida a la funció "deleteAccount()" del controlador, que com podem observar en la figura 45, dins d'aquesta funció invoquem al mètode "deleteAccount()" del "AuthenticationManager" per poder eliminar així l'usuari a Firebase. A més podem observar com s'actualitza l'estat de l'aplicació i es redirigeix a l'usuari a la pantalla de Login, sense cap compte obert i sense l'existència d'aquell compte. És a dir, com si aquell usuari obrís de nou l'aplicació per tornar a començar.

```
func deleteAccount() {
    authenticationManager.deleteAccount()
    UserDefaults.standard.set(false, forKey: "loggedIn")
    isUserAuthenticated = false
    if let windowScene = UIApplication.shared.connectedScenes.first as? UIWindowScene,
        let window = windowScene.windows.first {
        window.rootViewController = UIHostingController(rootView: LoginView())
        window.makeKeyAndVisible()
    }
}
```

Figura 45. Codi d'eliminació conta permanentment

6.5 Servei de localització

El servei de localització és una part important d'aquest projecte, ja que permet obtenir la ubicació precisa de l'usuari i utilitzar aquesta informació per diverses funcionalitats a l'aplicació.

En el meu cas, el servei de localització que he dut a terme en aquest projecte, ha estat el següent:

1. *Core Location*: És un framework de iOS, l'utilitzo per poder accedir a les funcionalitats de localització, ja que aquest framework em proporciona una API per poder interactuar amb els serveis de localització del dispositiu en concret.
2. *Sol·licitud d'autorització*: Abans d'obtenir la localització de l'usuari, s'ha de sol·licitar i obtenir la seva autorització per poder accedir a la seva ubicació, ja que tal com està documentat a la pàgina oficial d'Apple [8] és necessari que l'usuari doni permisos per poder treballar amb la seva ubicació, és un motiu de privacitat de l'usuari.
3. *Creació d'un CLLocationManager*: Un cop l'usuari ha donat permisos i autorització per l'ús de la seva ubicació, creo una instància de CLLocationManager, que és la classe principal del framework Core Location, explicat prèviament. Aquest gestor de localització actua com a un punt d'entrada per accedir a les funcionalitats de localització.

En resum, el servei de localització utilitzat en aquest projecte és gestionat a través del framework Core Location (com s'ha explicat) d'iOS. Mitjançant la sol·licitud d'autorització de l'usuari i la creació d'un CLLocationManager, es pot accedir a la ubicació precisa de l'usuari. Això ens permet habilitar diverses funcionalitats basades en la seva ubicació, aleshores després d'haver parlat sobre el funcionament del servei de localització podem passar a descriure com s'ha implementat l'agrupació de marcadors dels gimnasos al mapa.

6.5.1 Agrupació de marcadors de gimnasos al mapa

En aquesta secció es descriurà l'agrupació de marcadors de gimnasos en el mapa de l'aplicació, com s'ha dut a terme i que és el que s'ha fet servir per poder-ho realitzar.

Per poder implementar l'agrupació de marcadors de gimnasos en el mapa s'ha fet servir la classe de "MKMapView" extreta de l'API de MapKit d'Apple, que és la llibreria que s'ha fet servir per crear el mapa. També he utilitzat la classe de "CLLocationManager", que tal com s'indica a la documentació oficial d'Apple Developers [9], un objecte CLLocationManager és el lloc central per gestionar els comportaments relacionats amb la localització de l'aplicació, el qual utilitza un objecte LocationManager per configurar, iniciar i aturar serveis de localització, que aleshores em serveix per obtenir la ubicació de l'usuari, amb l'autorització prèvia d'aquest, tal com s'ha explicat prèviament.

Per poder afegir aquests marcadors i poder visualitzar-los al mapa, s'afegeixen els nous marcadors "GymAnnotation" al mapa mitjançant el mètode "addAnnotations". Els

marcadors s'obtenen a partir de la propietat “nearbyGyms” del “LocationManager”, que és un array de “GymAnnotation” que representa els gimnasos propers.

Una de les funcions importants és la funció de **mapView(_:viewFor:)** que podem observar a la *figura 46*, que permet personalitzar la vista dels marcadors en el mapa. En aquesta funció, s'obté l'objecte d'anotació i es comprova si és de tipus **GymAnnotation**.

```
func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation) -> MKAnnotationView? {
    guard annotation is GymAnnotation else { return nil }

    let identifier = "Gym"
    var annotationView = mapView.dequeueReusableAnnotationView(withIdentifier: identifier) as?
        MKMarkerAnnotationView

    if annotationView == nil {
        annotationView = MKMarkerAnnotationView(annotation: annotation, reuseIdentifier: identifier)
        annotationView?.canShowCallout = true
        annotationView?.rightCalloutAccessoryView = UIButton(type: .detailDisclosure)
    } else {
        annotationView?.annotation = annotation
    }

    return annotationView
}
```

Figura 46. Codi gym al mapa

Quan l'usuari fa clic a l'accessori de detall d'un marcador, s'activa la funció **mapView(_:annotationView:calloutAccessoryControlTapped:)** que podem apreciar a la *figura 47*. En aquesta funció, s'obté l'objecte **GymAnnotation** de l'anotació i es crea un **MKPlacemark** amb les seves coordenades. A continuació, es crea un **MKMapItem** amb aquest **MKPlacemark** i es configura el nom del gimnàs. Finalment, es crida el mètode **openInMaps** de **MKMapItem** per obrir l'aplicació Mapes amb l'opció de mostrar les direccions al gimnàs seleccionat.

```
func mapView(_ mapView: MKMapView, annotationView view: MKAnnotationView, calloutAccessoryControlTapped control: UIControl) {
    guard let gymAnnotation = view.annotation as? GymAnnotation else { return }

    let placemark = MKPlacemark(coordinate: gymAnnotation.coordinate)
    let mapItem = MKMapItem(placemark: placemark)
    mapItem.name = gymAnnotation.title

    mapItem.openInMaps(launchOptions: [MKLaunchOptionsDirectionsModeKey: MKLaunchOptionsDirectionsModeDriving])
}
```

Figura 47. Codi anotacions al mapa

En resum, a través de la combinació de **MKMapView**, **CLLocationManager** i les classes auxiliars com **GymAnnotation**, s'implementa l'agrupació de marcadors de gimnasos al mapa.

6.6 Servei de pagament

En aquesta part de la documentació, s'explica com s'utilitza la passarel·la de pagament, utilitzant la plataforma de Stripe en l'aplicació, ja que és una forma senzilla i segura de realitzar pagaments en línia. Primer de tot, comentar que quan vaig plantejar la

integració de pagaments en el projecte, vaig estar investigant diverses opcions i vaig acabar decidint per utilitzar Stripe, per les seves excel·lents característiques.

Per poder començar a utilitzar Stripe, es va d'instal·lar la llibreria Stripe SDK per a iOS. Per configurar alguns aspectes del procés de pagament, aquests es realitzen al mateix portal web del servei Stripe. Un cop fetes les configuracions es pot integrar la llibreria amb l'aplicació a desenvolupar i permet integrar les finestres configurades prèviament, i afegir la lògica per poder processar la transacció i així gestionar les respostes de Stripe.

En aquestes figures, *figura 48*, *figura 49*, *figura 50*, es pot apreciar les diferents pantalles de pagament realitzades amb Stripe per utilitzar en el projecte, a les quals es pot observar com a les pantalles es mostra l'opció de poder seleccionar el mètode de pagament desitjat, com pot ser targeta de crèdit, o altres opcions disponibles. També es mostren detalls de la compra, com ara l'import total i la descripció del producte o servei.

FitYou	
Pagar a FitYou	
2,50 €	
Premium plan	2,50 €
Subtotal	2,50 €
Impuesto	0,00 €
Total a pagar	2,50 €

Pago con tarjeta

Correo electrónico

Información de la tarjeta

1234 1234 1234 1234

MM / AA CVC

Nombre del titular de la tarjeta

País o región

España

Guardar mis datos de forma segura para un proceso de compra en un clic

Introduce tu número de teléfono para crear una cuenta de Link y pagar con mayor rapidez en FitYou y en todos los comercios que acepten Link.

link · Más información

Pagar

Si confirmas el pago, le permitiremos a FitYou que cargue en tu tarjeta este pago y futuros pagos conforme a las condiciones estipuladas.

Figura 48. Vista compra 1

FitYou	
Pagar a FitYou	
1,00 €	
Customized workout & nutrition	1,00 €
Subtotal	1,00 €
Impuesto	0,00 €
Total a pagar	1,00 €

Información de contacto

Correo electrónico

Método de pago

Tarjeta IDEAL gropay Eps

Información de la tarjeta

1234 1234 1234 1234

MM / AA CVC

Nombre del titular de la tarjeta

País o región

España

Guardar mis datos de forma segura para un proceso de compra en un clic

Paga más rápido en FitYou y en todos los comercios que acepten Link.

Pagar

Powered by stripe | Condiciones Privacidad

Figura 49. Vista compra 2

Figura 50. Vista compra 3

Així doncs, gràcies a l'API de Stripe, que proporciona una interfície completa i potent per poder interactuar amb la plataforma de pagament, he pogut integrar i personalitzar la funció de pagament, he pogut proporcionar una forma ràpida i senzilla per els usuaris. Així que hem pogut observar com ens ha proporcionat, potència, flexibilitat i facilitat d'integració.

6.7 Servei de missatgeria

La implementació del servei de missatgeria de l'aplicació consisteix en diverses parts clau, és una part molt important ja que permet als usuaris comunicar-se i intercanviar missatges en temps real.

S'ha de tenir en compte que aquesta implementació utilitza Firebase, com a plataforma per emmagatzemar, i sincronitzar les dades de missatgeria en temps real. Aquest xat permet als usuaris veure els missatges enviats per altres persones, enviar els seus propis i veure el nom de l'usuari que ha enviat cada missatge.

Per començar amb la implementació, he creat el controlador de la vista, el qual és responsable de gestionar totes les dades que passen i s'intercanvien per el xat global. Aquesta classe es un ObservableObject, el que significa que permetrà actualitzar la interfície d'usuari automàticament quan les dades canviïn. Dins d'aquesta classe, es fa ús de *Firestore* de Firebase, per tal de poder interactuar amb la base de dades en temps real. S'inicialitza una instància de *Firestore* i es defineix una funció anomenada *setupListener*. Aquesta funció crea una referència a la col·lecció de missatges de Firebase i s'afegeix un *snapshot listener* per tal de poder escoltar els canvis que es produeixin en la col·lecció. Quan es produeix un canvi, es recupera el snapshot de les dades, s'analitzen els documents per convertir-los en instàncies i actualitzar així les dades del controlador, i així la interfície d'usuari es refrescarà automàticament amb els nous missatges.

Al controlador, també he afegit una funció per poder recollir el nom de l'usuari que envia un missatge. Aquesta funció realitza una consulta a la col·lecció d'usuaris de Firebase

i recupera el document corresponent a l'ID d'usuari proporcionat, s'extreu el nom d'usuari i aleshores es crida a la funció de finalització proporcionada. Aquesta funció del projecte permet recuperar el nom, com s'ha explicat prèviament i enviar el missatge de l'usuari en concret.

Per poder enviar el missatge, s'ha creat la funció *sendMessage*, que és l'encarregada d'enviar el missatge en concret. Primer de tot es verifica que l'usuari estigui autenticat. Si ho està, s'obté el seu ID d'usuari i aleshores es crida a la funció per aconseguir el nickname de l'usuari associat. Amb el missatge de text a enviar, l'ID de l'usuari, el temps actual i el nom, es crea un diccionari de dades, el qual s'anomena "messageData". A continuació, s'afegeix aquest diccionari com un nou document a la col·lecció de missatges de Firebase, i aleshores es mostra com un nou missatge en la interfície d'usuari.

La implementació del xat global en el projecte permet disposar d'una plataforma de comunicació eficient i interactiva que els usuaris podran fer servir per connectar-se i compartir missatges.

6.8 Estadístiques codi projecte

Com s'ha descrit anteriorment, el codi del projecte es extens i a l'apartat d'implementació només s'ha comentat algunes de les parts més importants del projecte, aleshores per veure la magnitud del projecte desenvolupat al TFG, les línies de text, he utilitzat el comando "cloc .", que és una mesura utilitzada per quantificar la grandària d'un programa de software, en el meu cas, ens mostra la quantitat de fitxers, les línies en blanc, comentaris i diversos llenguatges de programació, i tal com veiem a la *figura 51* en aquest projecte podem observar el següent:

Language	files	blank	comment	code
Swift	77	764	842	4171
JSON	27	0	0	615
XML	9	0	0	214
SUM:	113	764	842	5000

Figura 51. Contingut del projecte

- Per al llenguatge Swift, tinc un total de 77 fitxers, els quals contenen 764 línies en blanc, 842 línies de comentaris i 4171 línies de codi font.
- Els fitxers JSON, són un total de 27 fitxers, no hi ha cap línia en blanc, ni cap línia de comentari, ocupen 615 línies de codi JSON.
- Els fitxers XML del projecte, són un total de 9 fitxers, segueix la mateixa dinàmica que els fitxers JSON, sense línies en blanc i sense comentaris, amb un total de 172 línies de codi XML.

En resum, el meu projecte consta d'un total de 113 fitxers, amb 764 línies en blanc, 842 línies de comentaris en general i la quantitat de línies de codi en total del projecte són 5000. Aleshores aquestes són les estadístiques que em proporcionen la mida i l'estructura total del meu projecte.

7 Avaluació

Es 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.

També és important demostrar el disseny final d'aplicació per poder comparar amb el disseny desenvolupat a l'apartat de disseny UI/UX. És a dir, no sempre el disseny de prototip queda igual com a la versió implementada, estaran tots els errors arreglats, inclús algunes vistes estaran completament diferents.

7.1 Crear new User

Començarem amb la funció bàsica de l'aplicació, com la creació d'un compte i l'inici de sessió, és a dir, s'ha de verificar que l'usuari pugui registrar-se amb èxit i accedir al sistema sense cap problema.

Un cop iniciada l'aplicació l'usuari veurà la pantalla mostrada a la *figura 52* que és la pantalla principal de l'aplicació, aleshores l'usuari es dirigirà a la pantalla de registre, tal com veiem a la *figura 53* i a la *figura 54* per tal d'introduir les seves dades. En aquesta part del projecte, podem observar com estem verificant com es crea l'usuari i s'afegeixen les seves dades a Firebase *figura 55*, aquesta és una part crucial del procés, ja que implica que tot el flux de creació d'usuaris i emmagatzematge de dades funciona de manera fiable i segura.

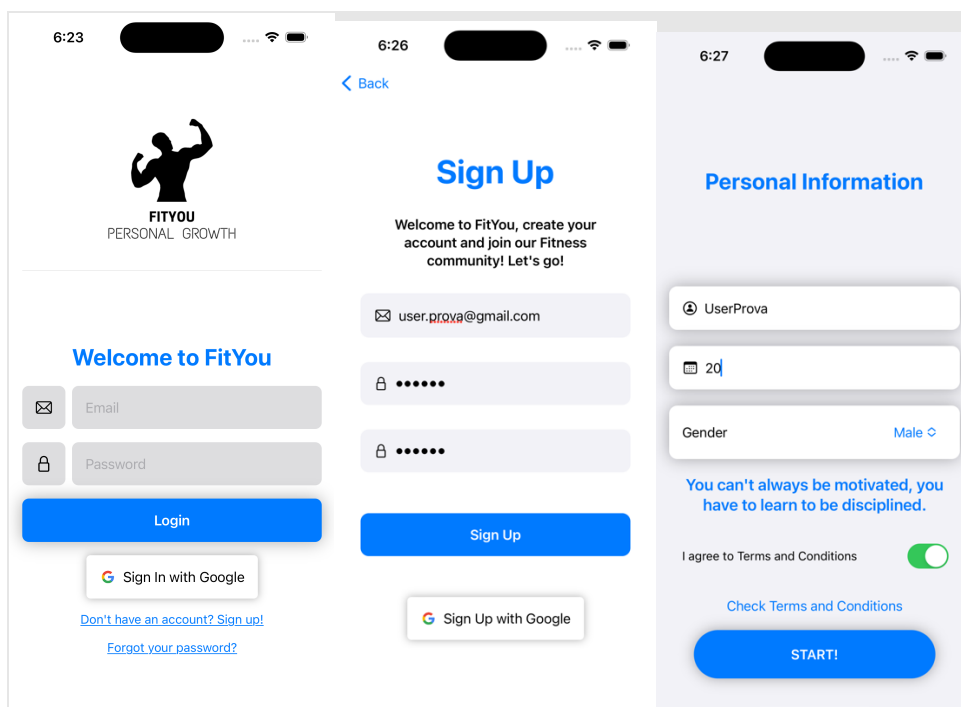


Figura 52. Pantalla principal **Figura 53.** Pantalla registre (verificació) **Figura 54.** Pantalla informació (verificació)

<input type="text" value="Buscar por dirección de correo electrónico, número de teléfono o UID de usuario"/> Agregar usuario ↻ ⋮				
Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
user.prova@gmail.com	✉	21 may 2023	21 may 2023	E61etepu9pdujXejhDC6sgpN4oI2

Figura 55. Verificació funcionament Firebase

7.1.1 Visualització perfil

En aquest punt, es pretén avaluar la funcionalitat de la vista del perfil, per assegurar que el perfil creat en el punt anterior (ja s'ha vist prèviament com s'havia registrat) es pot visualitzar sense cap problema en la vista de perfil de l'usuari.

Primer de tot, es verifica la visualització de les dades del usuari. Es comprova que totes les dades emmagatzemades, com el nom, l'edat i el gènere siguin presentades de manera coherent i precisa tal com s'aprecia a la *figura 56*. S'observa com l'usuari disposa de l'opció d'afegir una fotografia al seu perfil. A més a més, d'afegir la fotografia també es comprova que l'usuari pugui editar la seva biografia, si més no, la informació que consideri interessant.

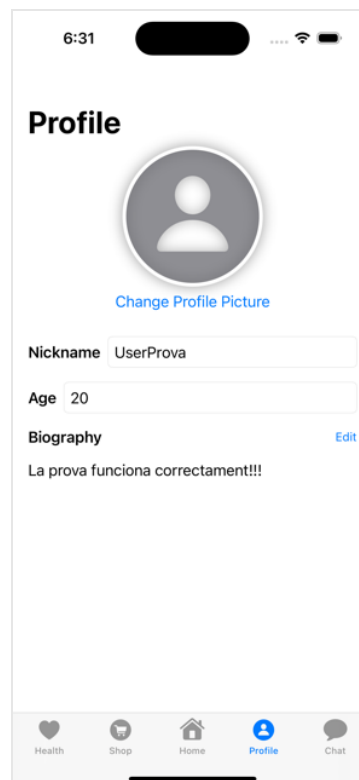


Figura 56. Verificació dades usuari

Així doncs, es pot apreciar que gràcies a les imatges prèviament vistes, podem observar com el funcionament de la pantalla d'usuari amb les seves dades funciona correctament i no presenta cap problema.

7.2 Verificació visualització principal aplicació

En aquest apartat, ens centrarem en la verificació de les vistes principals de l'aplicació. La fase de verificació implica posar a prova les funcionalitats clau i assegurar que els apartats d'esports i nutrició (els principals) proporcionen una experiència fluida i coherent als usuaris amb totes les seves interaccions possibles.

7.2.1 Esports

7.2.1.1 Visualització exercicis

En aquesta fase de verificació del projecte, s'està posant a prova la visualització dels exercicis a través de la llista principal. Ens centrem a assegurar-nos que tots els exercicis es mostren adequadament en la llista, permetent als usuaris veure una visió general del contingut disponible.

Un cop a dins de la vista d'un exercici específic com podem veure en la *figura 57*, avaluem el funcionament de la interfície i el vídeo explicatiu associat. També verifica que l'usuari pugui accedir al vídeo sense problemes, reproduir-lo i visualitzar la informació explicativa relacionada amb l'exercici.

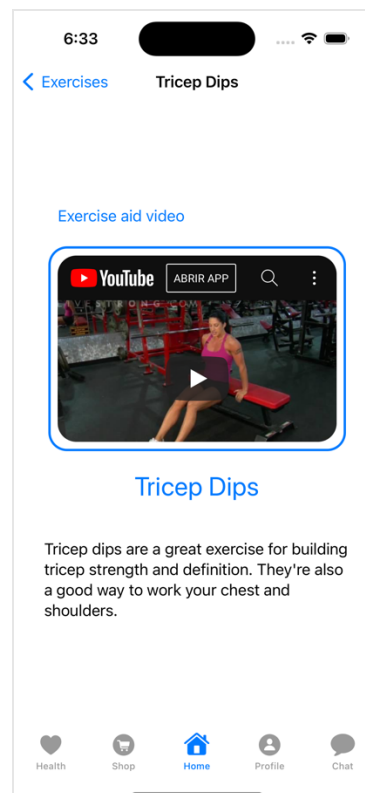


Figura 57. Verificació funcionament exercicis

Aquesta verificació és essencial per a l'èxit del nostre projecte i ens permet garantir una interacció fluida i satisfactòria amb el contingut dels exercicis.

7.2.1.2 Visualització rutines

A la fase de verificació de les rutines de l'aplicació, es comença verificant que la llista de rutines es mostra de manera adequada, i que permet als usuaris tenir una visió general dels diferents plans d'entrenament disponibles. És important que aquesta llista sigui clara i fàcil d'entendre, mostrant les rutines de forma ordenada i accessible.

Un cop un usuari selecciona una rutina en particular, ens centrem a avaluar la funcionalitat de la vista de detalls de la rutina com podem apreciar en la *figura 58*. Volem garantir que l'usuari tingui tota la informació necessària, com ara els exercicis, les repeticions i les sèries, per seguir la rutina amb precisió.

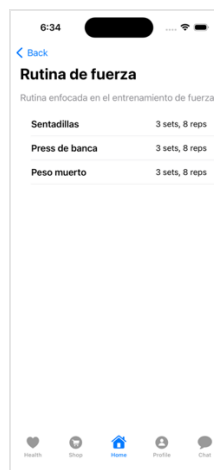


Figura 58. Verificació funcionament rutines

A més de les rutines predefinides, s'està comprovant com l'usuari pot afegir les seves pròpies rutines personalitzades. Tanmateix, s'està verificant que l'aplicació ofereixi les eines adequades per a aquesta funcionalitat, com ara la possibilitat d'afegir exercicis, establir repeticions i sèries, i emmagatzemar les rutines personalitzades per utilitzar-les posteriorment per tal que l'usuari les pugui visualitzar després a la llista de les seves rutines, tal com podem apreciar a la *figura 59*, després d'haver fet una prova d'afegir una rutina tal com veiem a la *figura 60*, correctament afegida a Firebase a la col·lecció del usuari en qüestió.

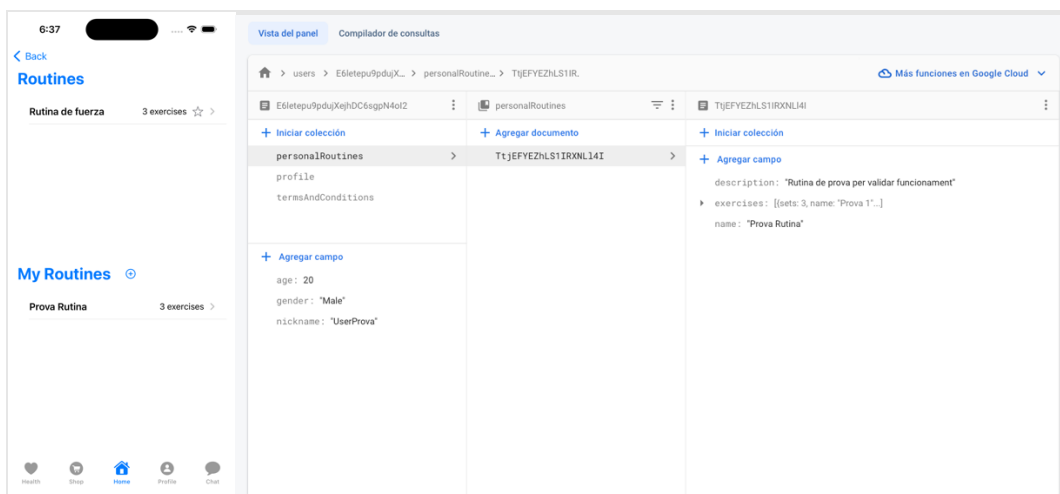


Figura 59. Verificació rutina afegida Figura 60. Verificació funcionament Firebase rutina afegida

7.2.1.3 Verificació funcionament preferits

En aquest punt de la verificació, es durà a terme la verificació del funcionament dels preferits, tal com hem pogut observar a les imatges de les llistes principals de les vistes d'exercicis i rutines en el punt del disseny de l'aplicació, es pot apreciar un botó de preferit (estrella), aleshores la funcionalitat d'aquest botó es que l'usuari pugui marcar tots aquells entrenaments que més li agradin i que es puguin emmagatzemar en aquestes llistes apartades, per tal de poder donar un accés més ràpid i lleuger.

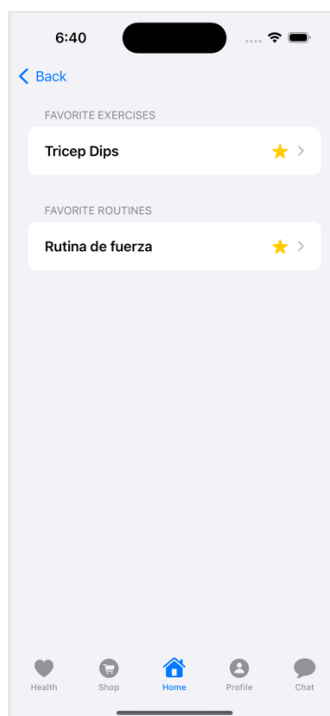


Figura 61. Verificació preferits

Tal com es pot apreciar a la *figura 61*, el funcionament del botó de preferits funciona correctament complint totes les expectatives i donant-li al usuari aquesta rapidesa buscada i lleugeresa a l'hora de tornar a visualitzar aquells entrenaments que es volia emmagatzemar.

7.2.2 Nutrició

7.2.2.1 Visualització dietes

En aquesta etapa del procés de verificació, durement a terme la verificació del funcionament de la visualització de les dietes que proporciona l'aplicació. Es pot verificar que les dietes es mostren adequadament a la pantalla principal, tal com es mostra a la *figura 62*.

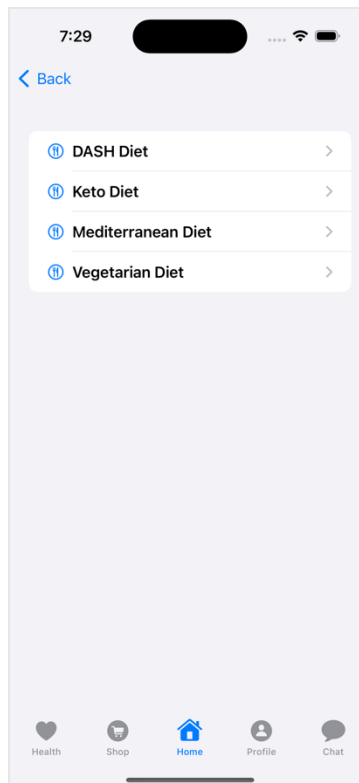


Figura 62. Verificació funcionament llista dietes

Un cop l'usuari selecciona una dieta concreta, podem veure com l'aplicació dirigeix al usuari a la vista detall, tal com es mostra a la *figura 63*, proporcionant una descripció concisa per la dieta seleccionada, amb els diferents àpats i amb una descripció per cada àpat en concret. Es pot apreciar que la informació es mostra sense cap problema i que les dades s'agafen perfectament de Firebase.

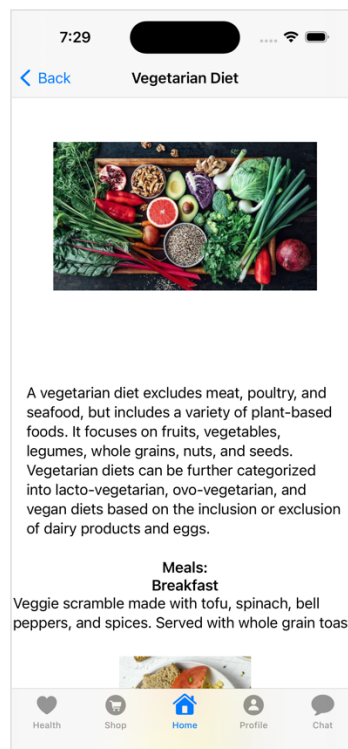


Figura 63. Verificació vista detall dieta

7.2.2.2 Visualització receptes

Seguint amb el procés de verificació, realitzem les tasques per comprovar la visualització adient de les dietes, aquí verificarem el funcionament de les receptes, que funciona de la mateixa manera que les dietes. Tal com podem observar en la *figura 64*, veiem que les dades s'agafen correctament de Firebase i es mostren adequadament a la llista de la vista principal d'aquest apartat, proporcionant el títol, una imatge, etc.

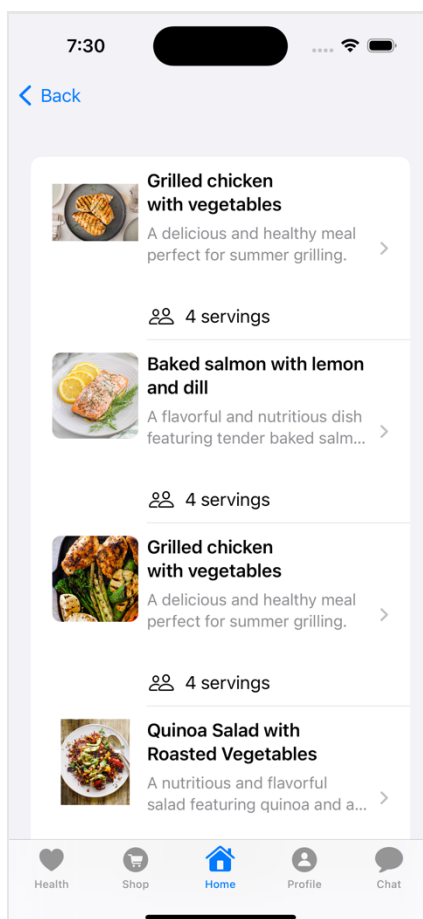


Figura 64. Verificació llista receptes

Un cop l'usuari selecciona una dieta concreta, es pot veure com l'aplicació es dirigeix a la vista detall (mateix funcionament que totes les vistes principals), s'ha verificat que si les imatges no es recarreguen instantàniament, surt la rodeta de "loading" indicant que s'està descarregant la imatge i sent preparada per mostrar-se.

7.2.2.3 Visualització Healthy food

Segueix amb el mateix funcionament que les pantalles de dietes i les pantalles de receptes. S'observa a la *figura 65*, i es veu que les dades s'agafen correctament de Firebase i es mostra adequadament a la pantalla de "Healthy Food".

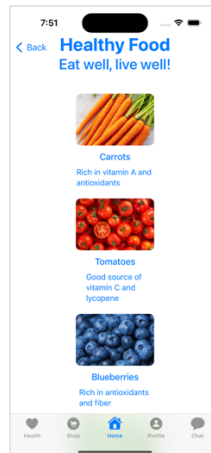


Figura 65. Verificació vista menjars saludables

En aquesta pantalla s'ha avaluat molt ràpidament les funcionalitats requerides, ja que només consta de recuperar les dades de Firebase i mostrar-les al usuari, en aquest apartat l'usuari no té cap interacció directa amb la interfície, només veu les dades existents.

7.2.3 Visualització Health View

En aquest apartat, s'avaluarà les funcionalitats de la pantalla de salut, s'ha de verificar que el càlcul de l'operació es realitzi sense problemes, es guardin al historial de càlculs i que Firebase registri tots els càlculs del usuari, amb l'alçada corresponent, l'altura i la data en que s'ha realitzat aquell càlcul.

A continuació faig referència a la pàgina web [10] d'on he tret la fórmula per poder calcular l'índex de massa corporal i la taula de classificació de la OMS de l'estat nutricional en relació a l'esmentat índex de massa corporal.

$$BMI = \frac{weight(kg)}{height(m^2)}$$

Tal com es pot veure a la *figura 66* els càlculs es realitzen sense cap problema i surt el missatge corresponent respecte el resultat obtingut.

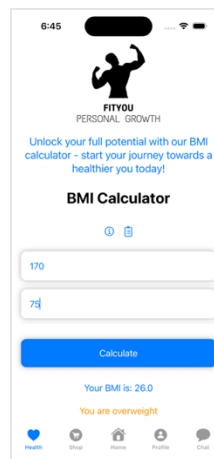


Figura 66. Verificació funcionament càlcul BMI

Després d'haver verificat el funcionament del càlcul, s'ha de verificar si l'historial disponible que té l'usuari per poder consultar tots els càlculs realitzats prèviament funciona correctament.

S'observa a la *figura 67* com els càlculs que l'usuari ha realitzat prèviament es troben en la pantalla del historial, es verifica que cada càlcul ens dirigeix a la seva pantalla de detall perquè així l'usuari pot comprovar les dades que va posar en aquell moment. Aleshores, s'observa com es poden copiar les dades al porta papers del dispositiu i s'enganxen correctament, tal com s'aprecia a la *figura 68*.

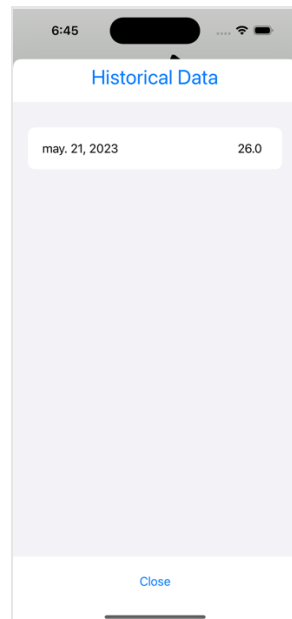


Figura 67. Verificació historial BMI

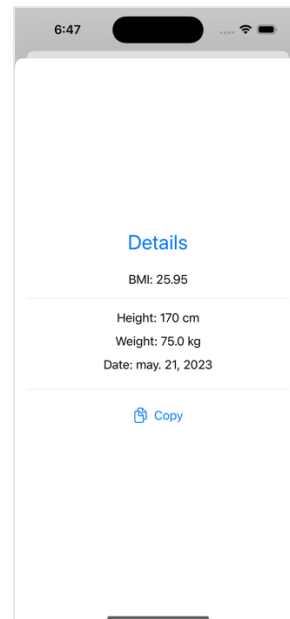


Figura 68. Verificació vista detall historial

7.3 Map View

En aquesta part del projecte, estem avaluant i verificant les funcionalitats que s'apliquen a la vista de Mapa (MapView). L'objectiu és permetre als usuaris visualitzar la seva ubicació actual i trobar gimnasos propers a través de l'aplicació.

Primer, s'inicialitza el LocationManager i es configura les preferències relacionades amb la localització. Es demana autorització a l'usuari per poder utilitzar la ubicació del dispositiu per quan s'estigui utilitzant l'aplicació, tal com es pot apreciar a la *figura 69*, i es defineix demanar a l'usuari el servei de localització que ens ofereix la millor precisió i distància de filtratge per a les dades de localització.

S'ha pogut comprovar i verificar que la visualització de la ubicació de l'usuari és precisa i concreta després de que aquest autoritzi l'ús de la seva ubicació i que les anotacions dels gimnasos propers es mostren de manera adequada. A més, les interaccions de l'usuari amb el mapa, com ara tocar una anotació per obtenir més informació o obtenir indicacions per arribar a un gimnàs, funcionen perfectament.

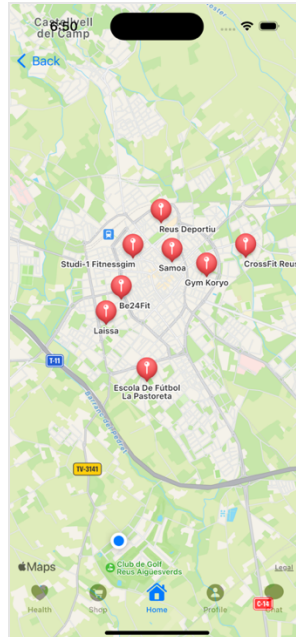


Figura 69. Verificació funcionament mapa

7.4 Global Chat App

En aquesta punt de verificació, avaluarem i verificarem les funcionalitats del xat global implementat en el projecte. L'objectiu d'aquest es permetre que els usuaris es puguin comunicar entre ells en temps real.

En el GlobalChatViewModel, hem comprovat que l'establiment del listener per escoltar els canvis a la col·lecció de missatges es realitza correctament. Quan hi ha un canvi en la col·lecció de missatges, els documents són processats adequadament, convertits en instàncies de ChatMessage i afegits a la llista de missatges publicada. Això permet una actualització en temps real del xat global i podem apreciar a la *figura 70* com es guarden tots els missatges amb el nickname de l'usuari en concret i totes les dades corresponents a Firebase.

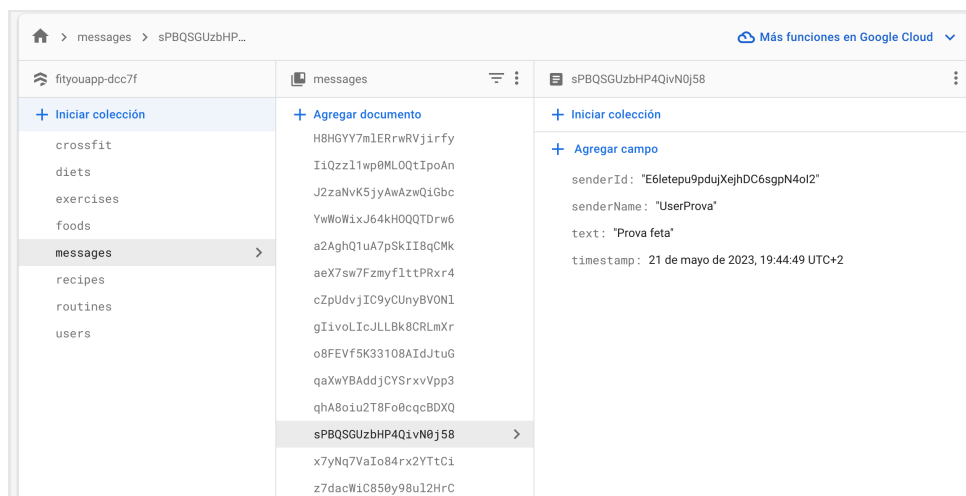


Figura 70. Funcionament Firebase missatges registrats

També hem verificat que l'enviament de missatges funciona com s'espera. Quan es fa clic al botó "Send", tal com es mostra a la *figura 71*.

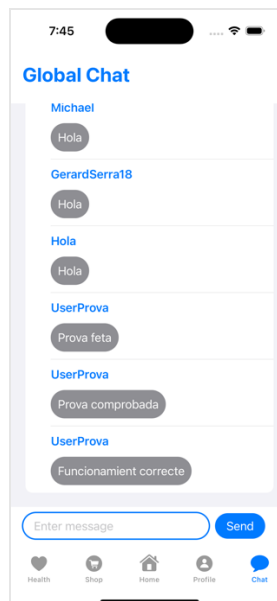


Figura 71. Verificació funcionament xat

Amb aquesta avaluació del xat global de l'aplicació, s'ha comprovat i verificat que funcionen totes les funcionalitats adequadament i s'ofereix al usuari una experiència fluida i interactiva amb el xat per tal de poder-se comunicar amb els altres usuaris de l'aplicació.

7.5 Verificació Plans

En aquest punt veurem i realitzarem l'avaluació per a l'apartat de plans que hi ha al projecte. Tal com podem veure a la *figura 72*, els tres plans es mostren adequadament a la pantalla principal, s'estan mostrant a l'usuari de manera perfecta, cadascun d'ells amb el títol i la informació corresponent.

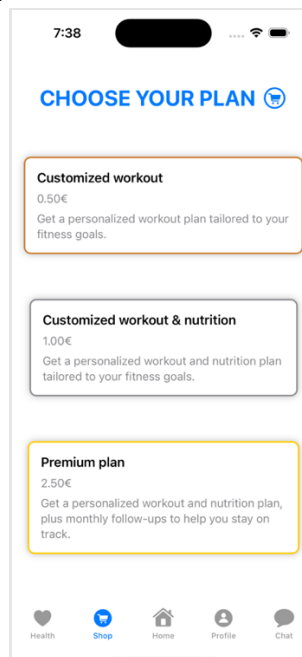


Figura 72. Pantalla amb els diferents plans

Podem observar com l'usuari selecciona un pla, s'obre la vista detall d'aquell pla corresponent, tal com observem a la *figura 73* tots els detalls del pla en concret també es mostren sense cap problema.

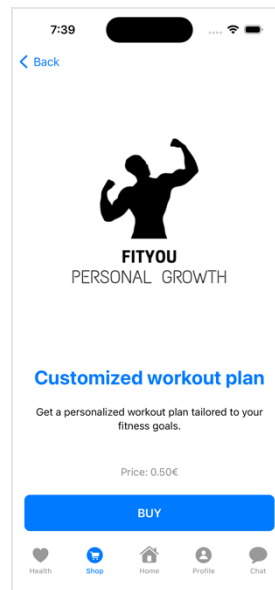


Figura 73. Vista detall pla seleccionat

En aquest apartat, després d'haver avaluat l'avaluació de la visualització, s'ha de realitzar la verificació i comprovació del funcionament de la integració que s'ha portat a terme amb la passarel·la Stripe per tal de poder efectuar els pagaments. Aleshores, per tal de corroborar aquesta implementació, podem agafar el pla de la *figura 73*, clicar sobre el botó de comprar i podem observar com l'aplicació envia a l'usuari a internet a la finestra de pagament de la nostra aplicació per tal de poder efectuar un pagament de manera segura i efectiva, tal com es veu a la *figura 74*, la finestra de pagament es la corresponent al pla escollit per l'usuari.

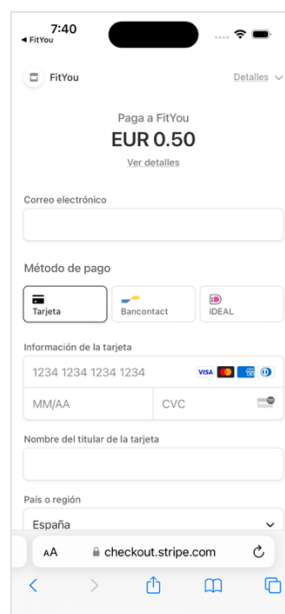


Figura 74. Pantalla pagament

Hem pogut apreciar que l'apartat dels diferents plans de l'aplicació funciona correctament sense problemes i l'aplicació proporciona una experiència fluida i segura als usuaris per tal de poder realitzar les transaccions necessàries.

7.6 Validació del funcionament “Settings”

7.6.1 Verificació funcionament Log Out

En aquest apartat de verificació, mirem com la funcionalitat de tancament de sessió (log out) funciona sense problemes. Aquest procés de verificació, és un procés senzill, ja que l'usuari veurà la vista de la *figura 75*, que és la pantalla de configuració i senzillament haurà de polsar el botó per tancar sessió i tornar a la pantalla principal de l'aplicació per tal de tornar a iniciar sessió.

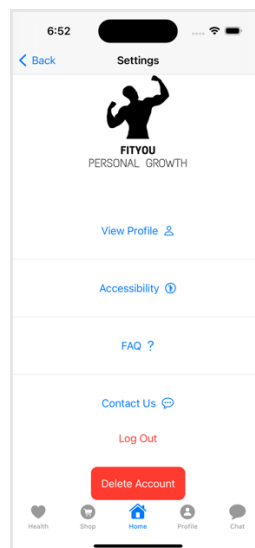


Figura 75. Pantalla global configuració

Veiem com l'usuari després de clicar en el botó, se l'ha redirigit a la pantalla principal, *figura 76*, per tal de verificar que el tancament de sessió funciona correctament.

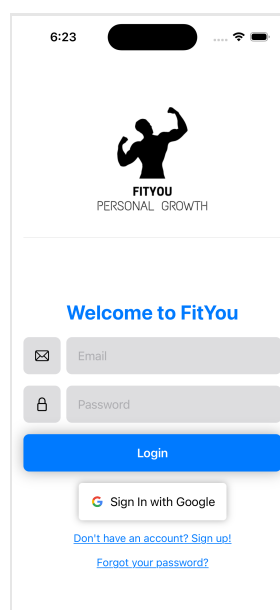


Figura 76. Pantalla inicial

7.6.2 Verificació eliminar usuari

L'últim punt a verificar, és la funcionalitat d'eliminació d'usuaris. És important que el projecte proporcioni als usuaris una manera segura i senzilla d'eliminar el seu compte de manera permanent de l'aplicació, aleshores aquest procés implica eliminar el registre de l'usuari amb totes les seves dades de Firebase.

Primer de tot quan l'usuari visualitza la pantalla de la *figura 75* i clica sobre el botó d'eliminar el compte podem apreciar com surt una alerta, *figura 77*, aquesta alerta serveix per garantir que l'usuari estigui segur de voler procedir amb l'eliminació del seu compte, per tal d'evitar eliminacions accidentals o involuntàries.

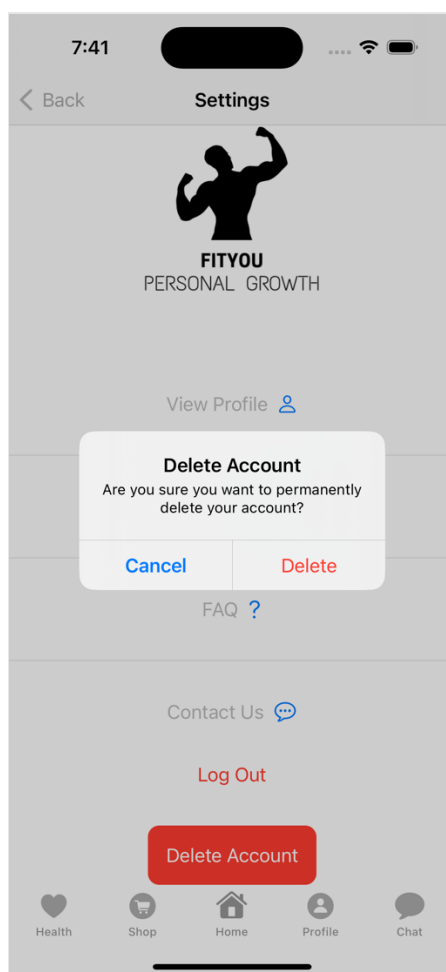


Figura 77. Avís eliminació permanent

Un cop l'usuari confirma l'eliminació del seu compte, s'està verificant que el registre d'usuari esborri de manera efectiva de Firebase, és a dir, aquest procés implica tal com s'ha explicat prèviament, que tota la informació del usuari com pot ser el nom, el correu electrònic, l'edat, el gènere, rutines que aquell usuari hagi generat, missatges que aquell usuari hagi enviat, exercicis preferits, etc. Tota informació relacionada amb aquell usuari quedi eliminada permanentment. Podem apreciar a la *figura 78* com el registre de l'usuari ja no existeix a la base de dades de Firebase.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
adeu2@gmail.com	✉	15 may 2023	15 may 2023	cJUD0xv0XFemYf0wSX7rBHA8L8...
gerard.serra18@gmail.com	✉	7 may 2023		s0GySMGuxcM9z9Zzi7A0xqt5cr52
adeu@gmail.com	✉	13 abr 2023	13 abr 2023	4umuy8kQ9oODLQ6efALiziuVMEb2
hola@gmail.com	✉	9 abr 2023	4 may 2023	ALRh4vg95AbckLHZJ3wxRBQEqe...
hola2@gmail.com	✉	8 abr 2023	8 abr 2023	0iBr6bSFMcPe5IPpLwwK88s0J02
gerard.serra@estudiants.ur...	✉	16 mar 2023	6 abr 2023	UKA3cHol9gYIRjyPOTyYH5ca0tjD3

Filas por página: 50 1 - 6 of 6

Figura 78. Comprovació funcionament eliminació compte a Firebase

Finalment, amb aquest usuari hem realitzat amb èxit totes les verificacions de les funcionalitats de l'aplicació. Després de finalitzar les proves, hem procedit a eliminar el compte (verificació feta prèviament també) de manera satisfactòria. Aleshores amb això podem confirmar que les funcionalitats de l'aplicació funcionen correctament i compleixen els requisits establerts i que funcioni correctament tot el projecte.

Durant el procés de desenvolupament, hem tingut l'oportunitat d'observar de prop l'evolució del disseny final de l'aplicació en comparació amb el disseny previ adjuntat amb errors de botons, etc. Amb cada iteració, hem pogut apreciar com els errors han estat solucionats i com s'han realitzat millores significatives (No apareixen tres botons de tornar enrere, etc.). No només s'ha tingut cura de corregir els problemes tècnics, sinó que també s'ha prestat atenció a aspectes visuals i d'experiència d'usuari cada cop que se seguia endavant amb la implementació del projecte.

Un dels aspectes on hem pogut analitzar grans diferències és a l'apartat de salut de l'aplicació. En les primeres versions, aquesta secció semblava mancar de cohesió i claredat, teníem un calendari i l'intent d'una gràfica que se'ns superposava i no se'ns apreciava bé a la pantalla, les dades no estaven organitzades adequadament, i els usuaris tenien dificultats per trobar la informació que necessitaven; aleshores es pot considerar com el disseny final ha sigut totalment diferent i molt més atractiu i intuïtiu.

En resum, hem pogut analitzar en aquest apartat d'avaluació que el procés de disseny i desenvolupament és una evolució constant, i que fins i tot els errors inicials poden ser corregits i superats amb perseverança i dedicació, que no sempre la plantilla principal acaba sent el fruit del final.

8 Conclusions

Amb aquest projecte es crea una aplicació nativa per la plataforma de iOS que permet als usuaris registrar-se en un servei fitness. L'usuari disposa d'una aplicació mòbil que li permet realitzar diverses accions com pot ser visualitzar exercicis, visualitzar un mapa amb les localitzacions de gimnasos relativament propers, afegir rutines personalitzades, calcular l'índex de massa corporal, visualitzar dietes, etc.

Amb aquesta aplicació també es pot implementar un xat que permeti als usuaris interrelacionar-se en temps real. Considero que la implementació d'aquesta aplicació pot ajudar als amants de l'esport a resoldre diferents situacions relacionades amb la realització d'exercicis i/o a l'obtenció de respostes a diferents circumstàncies en l'exercici diari.

A l'inici del projecte només hi havia un sol objectiu que es centrava en l'idea crear una aplicació mòbil focalitzada només en proporcionar rutines d'exercicis i centrada al món del gimnàs, sense contemplar res de nutrició i altres esports, com CrossFit, etc. Un cop es va madurar la idea i després de diverses iteracions i descripció acurada de necessitats, van anar sorgint noves característiques a incorporar a l'aplicació. La maduració del projecte ha estat reiterativa, des del disseny de les primeres pantalles sense lògica fins a la implementació de les consultes complexes a la base de dades en temps real. Durant aquest projecte he pogut aplicar els coneixements obtinguts durant la carrera d'enginyeria en Sistemes i Serveis de Telecomunicacions, especialment la branca de Telemàtica.

Prèviament a aquest projecte, no havia abordat grans projectes de forma independent. Ja que durant els meus estudis universitaris, a les pràctiques, treballàvem en equips en els quals dividíem els projectes més grans en diferents parts. El treball en equip és essencial per a tot enginyer, però jo volia desafiar-me i dur a terme un gran projecte sense un equip per posar en pràctica els coneixements que havia adquirit. Volia comprovar si tenia prou habilitats per a implementar i completar aquest projecte de manera individual, anant més enllà de la programació i assumint un rol més semblant a l'arquitecte de programari o analista.

Durant el desenvolupament d'aquest projecte, vaig adquirir nous coneixements de programació per a la plataforma iOS. A la meitat del projecte, si sorgia algun error, era capaç de solucionar-lo ràpidament, però per altres errors requerien més temps. Tot i això, sempre aconseguia trobar una solució. Al final del projecte, aquests problemes havien desaparegut i no tenia dificultats amb la implementació de les diverses tasques. A més de crear una aplicació mòbil, el meu objectiu principal en aquest projecte era aprendre coses noves i aprofundir en els meus coneixements en el desenvolupament d'aplicacions iOS, especialment en l'ús del llenguatge Swift.

Al treballar al projecte em van sorgir moltes idees noves, fins i tot, com podria millorar el projecte en les versions futures. Primer de tot es podria realitzar la implementació d'una pàgina web de l'aplicació, per aquella gent que no sap que realitza l'aplicació poder consultar tota mena d'informació a la pàgina web oficial. D'aquesta manera, en les futures versions afegiré la pàgina web comentada prèviament per poder facilitar a l'usuari una informació prèvia de totes les funcions disponibles a l'aplicació, com per exemple, un xat global, rutines, exercicis, etc. També afegiré molts més esports per poder ampliar la xarxa d'usuaris de l'aplicació, per tal de poder englobar una comunitat molt més gran i que serveixi per a la majoria de gent, que no només es dedica al món del fitness, sinó al món dels esports

en general. Al final, es pot arribar a tenir una aplicació molt útil pel món del fitness i també per tota aquella gent que no té la suficient valentia per endinsar-se en un món com aquest, que necessiten l'empenta per poder arrancar a construir un camí en el món del fitness i estar sempre informat amb les millors informacions nutricionals i millors rutines de l'esport que més els hi convingui.

9 Referències

Webgrafies:

- [1] Planificador Notion:
<https://www.notion.so/es-es>
- [2] Aplicacions natives:
<https://www.3androides.com/actualidad/100-que-es-el-desarrollo-de-aplicaciones-nativas>
- [2] Eina per descriure arquitectura iOS:
https://developer.apple.com/documentation/#!/apple_ref/doc/uid/TP40007898-CH1-SW1
- [3] Android Developers:
<https://developer.android.com/docs?hl=es-419>
- [4] Eina per descriure arquitectura Android:
<https://developer.android.com/guide/platform/index.html?hl=es>
- [5] Descripció requisits funcionals i no funcionals:
<https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>
- [6] Arquitectura MVC:
<https://www.swiftbeta.com/arquitectura-model-view-controller-en-swift/#model-view-controller>
- [7] Avantatges i desavantatges de les bases de dades SQL:
<https://www.geeksforgeeks.org/advantages-and-disadvantages-of-sql/>
- [8] Privacitat i Localització a iOS:
<https://support.apple.com/es-es/HT203033>
- [9] CLLocationManager iOS:
<https://developer.apple.com/documentation/corelocation/cllocationmanager>
- [10] Wikipedia IMC:
https://es.wikipedia.org/wiki/%C3%8Dndice_de_masa_corporal