



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

FitHive: Una aplicación centralizada de entrenamientos y
Fitness

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Castillejos Doménech, Iván

Tutor/a: Andrés Martínez, David de

CURSO ACADÉMICO: 2022/2023

Resum

La incorporació de les tecnologies en la vida esportiva de les persones és alguna cosa que porta ocorrent des de fa diverses dècades. Relotges intel·ligents, pulsòmetres i en l'última dècada els dispositius mòbils s'han obert pas en la vida de les persones. Degut a la seua versatilitat, moltes persones han optat per usar estos últims per portar un seguiment de la seua vida esportiva o de la seua dieta.

Aquest projecte pretén abordar eixa necessitat de portar un seguiment del nostre progrés, creant una plataforma en la qual puguem gestionar tots els aspectes de la nostra vida esportiva. Els usuaris que utilitzen la nostra aplicació comptaran amb diferents formes en les quals crear i emmagatzemar tant els plans d'entrenament que necessiten com portar un registre de les calories diàries i mesures corporals al llarg dels dies.

Paraules clau: Android; fitness, entrenaments, esport, aplicació mòbil

Resumen

La incorporación de las tecnologías en la vida deportiva de las personas es algo que lleva ocurriendo desde hace varias décadas. Relojes inteligentes, pulsómetros y en la última década los dispositivos móviles se han abierto paso en la vida de las personas. Debido a su versatilidad, muchas personas han optado por usar estos últimos para llevar un seguimiento de su vida deportiva o de su dieta.

Este proyecto pretende abordar esa necesidad de llevar un seguimiento de nuestro progreso, creando una plataforma en la que podamos gestionar todos los aspectos de nuestra vida deportiva. Los usuarios que utilicen nuestra aplicación contarán con diferentes formas en las que crear y almacenar tanto los planes de entrenamiento que necesitan como llevar un registro de las calorías diarias y medidas corporales a lo largo de los días.

Palabras clave: Android; fitness, entrenamientos, deporte, aplicación móvil

Abstract

The incorporation of technologies into people's sporting lives has been happening for several decades. Smartwatches, heart rate monitors, and in the last decade, mobile devices have made their way into people's lives. Due to their versatility, many people have chosen to use these devices to track their sports life or their diet.

This project aims to address this need to track our progress, creating a platform where we can manage all aspects of our sports life. Users who use our application will have different ways in which to create and store both the training plans they need and to keep a record of daily calories and body measurements over the days.

Key words: Android; fitness, sports, trainings, sport, mobile application

Índice general

Índice general	VII
Índice de figuras	IX
Índice de tablas	X
<hr/>	
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Metodología	2
1.4 Estructura de la memoria	3
2 Estudio de aplicaciones de <i>fitness</i>	5
2.1 Estudio de mercado	5
2.1.1 MyFitnessPal	5
2.1.2 Hevy	6
2.1.3 Tabata Timer	7
2.1.4 My workout Plan - Daily Workout Planner	7
2.1.5 Adidas Training: workout HIIT	8
2.2 Análisis de características	9
3 Especificación	13
3.1 Capa de presentación	14
3.2 Capa de lógica de negocio	25
3.3 Capa de datos	27
4 Implementación	31
4.1 Estudio de tecnologías y herramientas empleadas	31
4.1.1 Figma	31
4.1.2 Locofy	32
4.1.3 Github	33
4.1.4 Visual studio code	34
4.1.5 Android Studio	34
4.1.6 Firebase	35
4.1.7 React Native	36
4.2 Arquitectura Android app	37
4.3 Implementación de la aplicación	38
4.3.1 Obtención del código inicial de la aplicación	38
4.3.2 Estructura del código	39
4.3.3 Navegación	40
4.3.4 Listas	41
4.3.5 Autenticación	42
4.3.6 Base de datos en tiempo real	44
4.3.7 Icono de la aplicación	45
5 Resultados	47
5.1 Test de calidad	47
5.1.1 Experiencia Visual	47

5.1.2	Funcionalidad	48
5.1.3	Rendimiento y estabilidad	48
5.1.4	Privacidad y seguridad	49
5.2	Pruebas de usabilidad	49
5.3	Interfaz de la versión final	50
6	Conclusiones	61
6.1	Relación con los estudios cursados	62
6.2	Trabajo Futuro	62

Apéndices

A	Objetivos de desarrollo sostenible	67
B	Casos de uso	69

Índice de figuras

1.1	Arquitectura de código de 3 capas.	2
2.1	Logotipo de la aplicación MyFitnessPal.	5
2.2	Logotipo de la aplicación Hevy	6
2.3	Logotipo de la aplicación Tabata Timer	7
2.4	Logotipo de la aplicación My workout Plan - Daily Workout Planner.	8
2.5	Logotipo de la aplicación Adidas Training: workout HIIT	8
3.1	Arquitectura global de la aplicación	13
3.2	Pantalla de inicio de sesión y registro	14
3.3	Pantalla Olvide la contraseña	15
3.4	Pantalla de inicio	16
3.5	Pantalla de submenu	17
3.6	Pantallas de medidas corporales y estadísticas	17
3.7	Pantalla perfil de usuario	18
3.8	Pantallas de opciones del perfil	19
3.9	Pantallas de inicio entrenamientos	19
3.10	Pantallas de realizar entrenamientos	19
3.11	Pantallas de creación de planes	20
3.12	Pantallas biblioteca de entrenamientos	21
3.13	Pantallas creación de entrenamientos	22
3.14	Pantallas biblioteca de ejercicios	23
3.15	Pantallas inicio nutrición	23
3.16	Pantallas flujo nutrición	24
3.17	Pantallas biblioteca de ejercicios	25
3.18	Estructura de la base de datos	27
3.19	Estructura de las funciones de entrenamiento	28
3.20	Estructura de las funciones de comida, plan activado y entradas de calendario	28
4.1	Logotipo de Figma	31
4.2	Logotipo de Locofy	32
4.3	Logotipo de Github	33
4.4	Logotipo de Visual studio code	34
4.5	Logotipo de Android Studio	35
4.6	Logotipo de Firebase	35
4.7	Logotipo de React Native	36
4.8	Arquitectura de la aplicación.	37
4.9	Estructura de archivos del proyecto	39
4.10	Despliegue de ejercicios mediante Flatlist	42
4.11	Logotipo de Fithive.	45
5.1	Creación del usuario de Marta	51
5.2	Inicio de sesión con los datos	51

5.3	Submenú de la aplicación	52
5.4	Inicio de sesión con los datos	52
5.5	Biblioteca de planes de entrenamiento	52
5.6	Creador de planes	52
5.7	Pestaña para elegir entrenamiento a agregar	53
5.8	Creador de entrenamientos	53
5.9	Pestaña para elegir ejercicio a agregar	54
5.10	Creador de ejercicios	54
5.11	Entrenamiento a agregar	54
5.12	Plan con todos los datos añadidos	54
5.13	Activación del plan	55
5.14	Pantalla de entrenamientos con el plan activado	55
5.15	Selector de objetivo calórico	56
5.16	Selector de comida a añadir	57
5.17	Creador de comidas	57
5.18	Creador de alimentos	58
5.19	Comida a añadir.	58
5.20	Pantalla de nutrición actualizada	58
5.21	Biblioteca de medidas corporales	59
5.22	Creador de medidas corporales	59

Índice de tablas

2.1	Comparación de características de las aplicaciones analizadas	9
2.2	Análisis MoSCoW	10
A.1	Objetivos de desarrollo sostenible	67
B.1	CU-01, Registro en la aplicación	69
B.2	CU-02, Inicio de sesión	70
B.3	CU-03, Olvidar contraseña	70
B.4	CU-04, Cerrar Sesión	72
B.5	CU-05, Añadir medida corporal	72
B.6	CU-06, Modificar medida corporal	73
B.7	CU-07, Borrar medida corporal	74
B.8	CU-08, Visualizar estadísticas	74
B.9	CU-09, Modificar datos de la cuenta	75
B.10	CU-10, Iniciar entrenamiento	76
B.11	CU-011, Crear plan de entrenamiento	77
B.12	CU-012, Modificar plan de entrenamiento	78
B.13	CU-013, Borrar plan de entrenamiento	79
B.14	CU-014, Crear entrenamiento	80
B.15	CU-015, Modificar entrenamiento	81
B.16	CU-016, Borrar entrenamiento	81
B.17	CU-017, Crear ejercicio	82
B.18	CU-018, Modificar ejercicio	83
B.19	CU-019, Borrar ejercicio	83

B.20 CU-020, Cambiar objetivo calórico	84
B.21 CU-021, Añadir comida	85
B.22 CU-022, Modificar comida	86
B.23 CU-023, Borrar comida	87

CAPÍTULO 1

Introducción

En la era actual, la tecnología se ha convertido en una herramienta fundamental para mejorar nuestra salud y condición física, las aplicaciones de fitness son un ejemplo claro de ello.

1.1 Motivación

Tras la pandemia, se produjo un auge en la popularidad de los gimnasios. El confinamiento prolongado y la falta de actividad física han llevado a un aumento en la preocupación por la salud y la forma física. Muchas personas han comenzado a darse cuenta de la importancia de mantenerse activas y saludables para mantener un sistema inmunológico fuerte y reducir el riesgo de enfermedades [1].

El principal problema al que se enfrenta la gente que comienza a hacer actividad deportiva es que, en la mayoría de los casos, no saben por dónde empezar. Con el auge de los dispositivos móviles y el acceso a Internet que se dispone en la actualidad, este problema se intenta atajar mediante páginas web como “darebee”¹ o “Musclewiki”² o mediante algunas aplicaciones como pueden ser “hevy”³ o “myfitnesspal”⁴.

Estos recursos ayudan a resolver muchos problemas, pero a su vez generan otros nuevos: interfaces poco claras, funcionalidades importantes bloqueadas detrás de una barrera de pago, notificaciones basura (*spam*), imposibilidad de almacenar los entrenamientos etc.

Todos estos problemas nos plantean la necesidad de crear una aplicación que permita solucionarlos.

1.2 Objetivos

El objetivo principal de este trabajo es la creación una aplicación para dispositivos móviles que nos permita abordar estos problemas de manera sencilla para ayudar a la gente a organizar mejor sus entrenamientos y su dieta para poder conseguir sus objetivos deportivos.

La aplicación dispondrá de varias funcionalidades, por un lado, contará con gestión de entrenamientos que nos permitirá crear planes, entrenamientos y ejercicios, así como

¹<https://darebee.com/>

²<https://musclewiki.com>

³<https://www.hevyapp.com>

⁴<https://www.myfitnesspal.com/es>

anotar nuestro progreso según vayamos realizándolos en nuestro día a día. Contará también con un apartado nutricional que nos permitirá marcar nuestros objetivos calóricos, almacenar comidas y llevar un seguimiento de nuestras calorías diarias para así hacer la pérdida o ganancia de peso mucho más llevadera. También incluirá un apartado que nos permita anotar nuestras medidas corporales, vital para poder llevar un seguimiento de nuestro progreso.

Por último, la aplicación contará con un apartado de estadísticas que nos permitirá observar mediante gráficos nuestro progreso a lo largo del tiempo, nos mostrará como hemos progresado en el ejercicio que queramos o cómo ha fluctuado nuestro peso corporal u otras de nuestras medidas corporales que anotaremos en el apartado correspondiente. Todo esto lo queremos abordar con una interfaz clara e intuitiva. Adicionalmente, la aplicación incluirá funciones de accesibilidad que permitirán a los usuarios personalizarla según sus necesidades.

1.3 Metodología

Como metodología de desarrollo de software se ha elegido una metodología ágil, ya que como nos indica [2] su desarrollo iterativo e incremental nos permitirá ser flexibles y adaptarnos a los cambios o problemas del proyecto a medida que vayan surgiendo.

En lugar de seguir un plan rígido predefinido, trabajaremos en ciclos cortos también conocidos como *sprint* que nos permitirán ajustar el enfoque del proyecto según vayamos desarrollando la aplicación.

La flexibilidad que nos otorga este enfoque respecto al resto nos permitirá también tener una reducción en los costes finales de la aplicación ya que con la corrección de errores y con los ajustes que hagamos en cada *sprint* nos permitirá atajar los problemas de manera eficaz según vayan surgiendo en el desarrollo.

Emplearemos también el método MoSCoW que tal y como cuenta [3] para identificar, definir y priorizar los requisitos de la aplicación en las fases de análisis del proyecto.

En materia de arquitectura de código, se empleará el modelo de programación por capas como podemos observar en la figura 1.1, utilizando además el modelo más habitual según IBM [4], el basado en 3 capas:

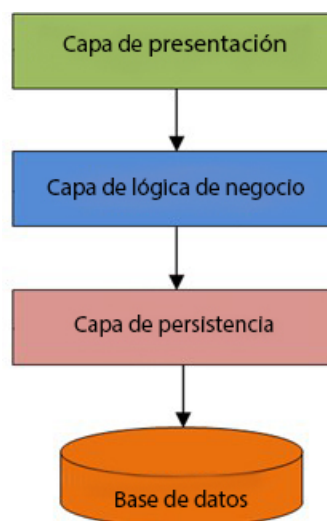


Figura 1.1: Arquitectura de código de 3 capas.

- La capa de presentación, que contiene las interfaces gráficas y es la capa con la que el usuario interactuará.
- La capa de lógica de negocio, que contendrá los procesos para realizar las múltiples funciones de la aplicación, y comunicará tanto con la capa de presentación para recibir instrucciones del usuario como con la capa de persistencia para consultar o guardar datos.
- La capa de persistencia, que será la encargada de comunicarse con la base de datos.

1.4 Estructura de la memoria

La memoria está estructurada de la forma siguiente.

- **Introducción:** En este capítulo se definirá la motivación del trabajo, así como sus objetivos, también se expondrá la metodología que se empleará y se mostrará la estructura general del documento.
- **Estado del arte:** En este apartado se realizará un análisis del mercado para poder determinar el estado actual de las aplicaciones relacionadas con nuestro trabajo. También se estudiará qué funciones debería tener nuestra aplicación empleando un sistema de tipo MoSCoW.
- **Especificación:** En esta sección se presentarán las diferentes capas del modelo que estamos desarrollando, también se presentarán los bocetos de la interfaz gráfica tras haber introducido los casos de uso del problema.
- **Diseño:** Este capítulo presentará la arquitectura global de la solución propuesta, también presentará las diversas tecnologías, herramientas y metodologías de las que disponemos y que vamos a utilizar en el desarrollo, así como el porqué.
- **Implementación:** Este apartado expondrá los aspectos relevantes del proceso de implementación como, por ejemplo, las decisiones de implementación que tomemos planteando diversas alternativas y el porqué de la solución adoptada, así como problemas de mayor dificultad que hayan tenido que tratarse con mayor profundidad.
- **Resultados:** Esta sección incluirá el desarrollo final de la solución, las pruebas que se hayan realizado, etc. Todo lo necesario para poder demostrar el buen funcionamiento de la solución.
- **Conclusiones y trabajo futuro:** Contendrá un pequeño resumen del trabajo realizado, indicando el grado de consecución de los objetivos planteados. Aquí también indicaremos las posibles expansiones que pueden realizarse a partir de este trabajo.

CAPÍTULO 2

Estudio de aplicaciones de *fitness*

En esta sección haremos un estudio de mercado de numerosas aplicaciones relacionadas con el mundo del deporte. De ellas analizaremos sus características para poder decidir qué incluir en nuestra aplicación.

2.1 Estudio de mercado

Empezaremos analizando diferentes aplicaciones del mercado.

2.1.1. MyFitnessPal



Figura 2.1: Logotipo de la aplicación MyFitnessPal.

MyFitnessPal¹, cuyo logotipo se muestra en la Figura 2.1, podría decirse que es una de las aplicaciones de referencia en el mundo del *fitness*. Cuenta con una gran cantidad de funcionalidades que nos ayudan a organizar nuestra alimentación para poder conseguir nuestros objetivos. Su buscador de alimentos es uno de los principales atractivos de la aplicación, ya que cuenta con un lector de código de barras con el que puedes obtener los macro nutrientes y las calorías de cualquier alimento que quieras al momento. Junto con este buscador, el creador de recetas te permite controlar las calorías y nutrientes que tomas en cada comida de forma muy sencilla y clara. Además, cuenta con una biblioteca para poder guardar las recetas y los alimentos que queramos.

Otras características menos conocidas de esta aplicación serían i) el contador de pasos, que te permite conectar tu reloj inteligente para monitorizar los pasos que recorres al día y ponerte metas que cumplir diariamente, ii) una función para añadir entrenamientos,

¹<https://play.google.com/store/apps/details?id=com.myfitnesspal.android>

tanto aeróbicos como de musculación, y iii) una biblioteca predeterminada de ejercicios que cuenta con la posibilidad de crear los tuyos propios.

La aplicación cuenta también con funciones sociales, como la posibilidad de leer artículos escritos por otros usuarios desde la aplicación, agregar amigos para compartir recetas, entrenamientos e incluso intercambiar mensajes.

Como hemos podido observar, esta aplicación es de las más completas del mercado, pero no todo va a ser positivo. La interfaz es poco clara a la hora de registrar tanto alimentos como entrenamientos, ya que incluye en casi todas las pestañas un anuncio para que te unas al plan *premium* de la aplicación, lo que reduce bastante el espacio de trabajo, haciendo que sea mucho menos cómoda de utilizar. La calculadora de calorías diarias funciona de manera irregular. Si tu objetivo es mantener el peso corporal no habría ningún problema, mientras que, si tu objetivo es aumentar o disminuir el mismo, la calculadora no da un valor del todo acertado. La función social no está muy bien implementada ya que, cuando quieres leer artículo, abre un navegador interno que no funciona adecuadamente en vez de abrir los artículos en la propia aplicación. Por último, cabe destacar que la aplicación cuenta con un número de funcionalidades escondidas tras un plan de pago.

2.1.2. Hevy



Figura 2.2: Logotipo de la aplicación Hevy

La aplicación Hevy², cuyo logotipo se muestra en la Figura 2.2, podría considerarse el ejemplo a seguir a la hora de implementar la gestión de entrenamientos en la aplicación. Cuenta con una amplia biblioteca de ejercicios entre los que elegir y te permite crear tus propios ejercicios personalizados para que no estés limitado a las opciones que te ofrece su biblioteca.

El creador de rutinas emplea una interfaz sencilla y clara que te permite seleccionar los ejercicios, ya sea buscando en su biblioteca o creándolos tú mismo. También te permite seleccionar el número de series y de repeticiones que queremos realizar por ejercicio. Adicionalmente, podemos indicar el peso que queremos utilizar y el tiempo de descanso entre series y ejercicios, ya que la aplicación dispone de un temporizador incorporado para ayudarnos. Si acaso no queremos crear nuestra propia rutina, la aplicación cuenta también con una amplia biblioteca alimentada tanto por los desarrolladores como por los propios usuarios

La aplicación también cuenta con un apartado para apuntar las medidas corporales. Esta función, junto con el registro de entrenamientos, nos permitirá utilizar las estadísticas que proporciona la aplicación para observar nuestro progreso de forma clara.

²<https://play.google.com/store/apps/details?id=com.hevy>

Por último, esta aplicación funciona también como una red social, ya que permite buscar y agregar amigos con los que podremos compartir nuestras rutinas y progreso.

Existe una versión de pago que te permite crear más rutinas y ejercicios que en la versión gratuita y no limita el historial de estadísticas a solo 3 meses.

2.1.3. Tabata Timer



Figura 2.3: Logotipo de la aplicación Tabata Timer

Esta sencilla aplicación³, cuyo logotipo se muestra en la Figura 2.3, cuenta con un cronometro diseñado para entrenamientos de tipo HIIT (del inglés *High-Intensity Interval Training*) o tabata, con el que puedes cambiar la duración de cada intervalo del entrenamiento que tengas pensado de manera rápida y sencilla. Se puede ajustar la cantidad de ejercicios que quieres realizar e incluso poner nombres a cada uno de ellos para que sea fácilmente reconocible a la hora de hacerlos. Otra funcionalidad que incluye es la opción de importar o exportar entrenamientos guardados en tu teléfono o en Google Drive.

La aplicación también incluye un calendario que registra los días que has hecho ejercicio usando la aplicación, indicándote su duración y el tipo de entrenamiento.

La personalización de la interfaz también forma parte de esta aplicación, pudiendo cambiar tanto los sonidos a la hora de cambiar de ejercicio o de serie, como modificar los colores de la interfaz para poder leerla mejor.

Por último, cabe destacar que existe una versión de pago que da acceso al almacenamiento de entrenamientos de forma ilimitada, te libra de los anuncios y permite utilizar la aplicación en segundo plano.

2.1.4. My workout Plan - Daily Workout Planner

Esta aplicación⁴, cuyo logotipo se muestra en la figura 2.4, te permite almacenar tus entrenamientos de manera sencilla. Incluye una biblioteca de ejercicios bastante completa que te permite filtrar por diferentes categorías, ya sea por el tipo de músculo que quieres entrenar, por el tipo de equipamiento del que dispones o si es de musculación o de estiramiento. Una vez creada la rutina, puedes ajustar los tiempos de descanso entre cada serie y ejercicio y así poder reproducirla en la misma aplicación.

Cada vez que realizas un entrenamiento, la aplicación lo almacena para poder mostrarte las estadísticas de progreso con diferentes tipos de gráficos que te ayudan a observar tanto tu progreso como dónde has podido estancarte para poder mejorar. Es posible guardar tus medidas corporales en la aplicación para poder añadirlas a las estadísticas y

³<https://play.google.com/store/apps/details?id=com.simplevision.workout.tabata>

⁴<https://play.google.com/store/apps/details?id=com.myworkoutplan.myworkoutplan>



Figura 2.4: Logotipo de la aplicación My workout Plan - Daily Workout Planner.

así llevar un registro para poder tomar nota del progreso. Lamentablemente, para usar esta función debes estar suscrito al plan de pago.

Existe la posibilidad de compartir los entrenamientos que hayas diseñado con el resto de los usuarios, además de buscar y descargar rutinas creadas por otros miembros de la comunidad.

La versión de pago te permite crear diferentes planes de entrenamiento, sincronizar tus datos entre distintos dispositivos, añadir fotos a tus ejercicios personalizados y te garantiza el acceso a todas las nuevas funcionalidades que los desarrolladores implementen en la aplicación.

2.1.5. Adidas Training: workout HIIT



Figura 2.5: Logotipo de la aplicación Adidas Training: workout HIIT

La aplicación de Adidas⁵, cuyo logotipo se muestra en la figura 2.5, te permite crear entrenamientos a partir de su biblioteca de ejercicios. Podremos filtrarlos entre las diferentes partes del cuerpo que trabajan y el nivel de dificultad, para así facilitarnos la tarea de buscar el ejercicio que necesitamos para nuestro entrenamiento. La aplicación también cuenta con entrenamientos predeterminados de diferentes variedades. El único inconveniente es que no se nos permite guardar el entrenamiento en nuestra cuenta, teniendo que volver a buscarlo la próxima vez que queramos hacerlo. Si seleccionamos uno de los entrenamientos predeterminados de la aplicación se nos permite reproducirlo en vídeo para poder ver más claramente cómo realizar los ejercicios y así llevar, de forma más cómoda, el tiempo de descanso entre los ejercicios, la duración de estos, etc.

Una vez realicemos algún entrenamiento, la aplicación registrará sus estadísticas para poder observar nuestro progreso. También se ofrece al usuario la posibilidad de crear re-

⁵<https://play.google.com/store/apps/details?id=com.runtastic.android.results.lite>

tos para motivarle a conseguir sus metas. Existen tanto retos personalizados, que crearían los propios usuarios, como retos predeterminados, que incluirían los desarrolladores. Algunos ejemplos de dichos retos serían: correr 30 km a lo largo de un mes, entrenar 3 veces en una semana, o consumir x número de calorías en un día.

La aplicación cuenta con guías escritas por profesionales sobre distintos temas, como la nutrición o los distintos tipos de entrenamientos que existen. También dispone de un apartado de noticias relacionadas con el deporte y la salud, que algunos usuarios podrían encontrar de utilidad.

Uno de los posibles inconvenientes de la aplicación es la obligación de registrarte para poder usar la. También existen quejas de varios usuarios por la cantidad de noticias basura que genera la aplicación.

2.2 Análisis de características

Tabla 2.1: Comparación de características de las aplicaciones analizadas

Características	MyFitnessPal	Heavy	Tabata Timer	My Workout plan -Daily Workout	Adidas Training: Workout HIIT
Cuidad de alimentación	✓	✗	✗	✗	✗
Gestión de entrenamientos	✓	✓	⚠	✓	✓
Calendario	✓	✓	⚠	✓	✗
Estadísticas de progreso	✓	✓	✗	⚠	✓
Registro de medidas corporales	⚠	✓	✗	✗	✗
Red social	✓	✓	✗	✓	✗
Limitar Funciones bajo previo pago	✓	✓	✓	✓	✓
Obligación a registrarse	✗	✗	✗	✗	✓

En la tabla 2.1 se han cruzado todas las características analizadas entre las aplicaciones para observar cuáles eran las más comunes entre estas. Podemos observar como la gestión de entrenamientos es común en todas las aplicaciones, ya que es el pilar fundamental de las aplicaciones de *fitness*. Las funciones que suelen incluir son la creación de entrenamientos, la inclusión de una biblioteca de ejercicios y rutinas, y la posibilidad de reproducir los entrenamientos en la aplicación. También cabe destacar la inclusión de un contador de pasos en algunas de las aplicaciones, incluyendo la opción de sincronizar con Google Fit en la aplicación “My workout plan -Daily Workout”. Una característica interesante que incluyen algunas de las aplicaciones es el registro de medidas corporales. Esto es, para muchos, una función esencial a la hora de llevar un seguimiento del progreso entrenando, ya que luego también se podrá comparar el cambio de peso muscular, el cambio del tamaño de los músculos, etc.

Las estadísticas de progreso y el calendario están presentes en casi todas las aplicaciones, ya que es una función muy importante para cualquier deportista que quiera llevar un registro de su vida deportiva. Esta característica suele incluir gráficos y tablas, en conjunto con un calendario, para poder ver en qué áreas progresas y el ritmo al que lo haces.

El cuidado de la alimentación solo está presente en la aplicación MyFitnessPal ya que esta es la única que está pensada para algo más que un registro de entrenamientos. Estas características suelen incluir una calculadora de calorías diarias, en la cual introduces datos como la edad, el peso corporal y similares, y te proporciona un objetivo de calorías diarias. Después, te permite añadir comidas mediante un creador de recetas y un buscador de alimentos, para así poder llevar la cuenta de las calorías consumidas en cada comida del día. La biblioteca de alimentos en esta aplicación te permite leer el código de barras de los productos para poder obtener sus características nutricionales fácilmente.

Muchas aplicaciones funcionan también como red social, incluyendo la posibilidad de agregar amigos, compartir entrenamientos, ejercicios, recetas, así como un buscador para encontrar un plan de entrenamiento o alimenticio que se ajuste a las necesidades del usuario. En la aplicación de Adidas, se incluye un apartado de noticias escritas por profesionales que se ajustan a nuestros intereses. En "MyfitnessPal" se incluye una característica similar, pero sin llegar a estar implementada en la aplicación, siendo simplemente una redirección a la página web de la empresa, empleando un navegador interno para poder leer los artículos.

Por desgracia, todas las aplicaciones que hemos analizado incluían una versión de pago que desbloqueaba todas las funcionalidades de la aplicación. Por ejemplo, en la aplicación "My workout plan -Daily workout", al desbloquear la versión de pago, obtenías acceso a las estadísticas de tus entrenamientos, y en la aplicación "Hevy", te permitía aumentar el número de ejercicios personalizados y de entrenamientos distintos que podías guardar. Aunque es entendible que sea necesario crear una versión de pago para poder mantener la aplicación en funcionamiento, limitar tantas características detrás de una barrera de pago puede resultar en una reacción negativa por parte de los usuarios.

Tras este análisis vamos a describir cuales van a ser las funcionalidades de nuestra aplicación empleando el método MoSCoW.

Tabla 2.2: Análisis MoSCoW

Must	Should	Could	Won't
Cuidado de la alimentación	Red social		Limitación de funcionalidades bajo previo pago
Gestión de entrenamientos			
Calendario			
Estadísticas de progreso			
Registro de medidas corporales			

Según vemos en la tabla 2.2, existen cinco características que son indispensables para nuestra aplicación. Por una parte, necesitamos que la solución pueda gestionar la alimentación de nuestros usuarios, proporcionarles métodos para calcular sus objetivos y guardar sus comidas. También debe ser capaz de gestionar la vida deportiva de los usuarios,

permitiéndoles guardar los planes de entrenamiento que tengan, así como los entrenamientos y ejercicios que contienen. Por otra parte, la aplicación contará con un calendario que le permitirá ver qué entrenamientos realizó ese día o si completó su objetivo de calorías para esa fecha. Contará también con una forma de almacenar sus medidas corporales, esencial para medir el progreso del cliente, ya que también permitirá calcular el índice de grasa corporal de manera sencilla. Por último, existirá la posibilidad de observar el progreso que ha conseguido el usuario mediante gráficos con los datos almacenados de sus entrenamientos y medidas corporales.

Se podría plantear la creación de una red social a partir de la aplicación, es decir, añadir funcionalidades como, por ejemplo, añadir y buscar amigos, ver los entrenamientos o el progreso de otras personas, y ofrecer una mayor personalización para el perfil del usuario. Estas son solo algunas ideas; si el tiempo de desarrollo lo permite, se podrían añadir a nuestra solución, pero no será nada a lo que le demos prioridad.

Por último, comentaremos que algo que no tiene cabida en nuestra solución es la existencia de una barrera de pago para poder utilizar ciertas funciones en la aplicación. Este hecho, tan común en muchas aplicaciones del mercado, es algo que no queremos incluir en nuestro proyecto, ya que, muchas veces, el usuario se ve forzado a descargar varias aplicaciones: una para la alimentación y otra para los entrenamientos, por ejemplo. Esto va en contra de la filosofía con la que diseñamos esta aplicación.

CAPÍTULO 3

Especificación

En esta sección se estudiará la arquitectura de código que vamos a emplear en este trabajo, así como las capas del modelo que vamos a desarrollar. También se presentarán los casos de uso del problema y los bocetos que hemos desarrollado para la aplicación,

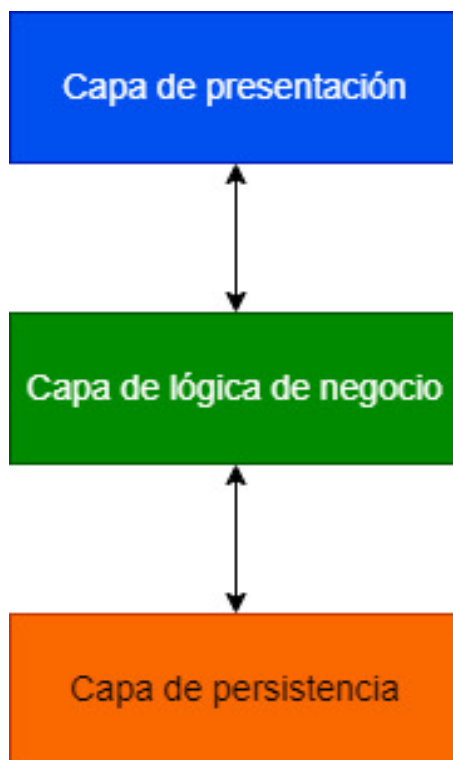


Figura 3.1: Arquitectura global de la aplicación

La arquitectura de tres niveles o capas es una de las arquitecturas de software predominante para las aplicaciones cliente-servidor tradicionales según [4].

Esta arquitectura organiza las aplicaciones en tres capas distintas como observamos en la figura 3.1; La capa de presentación que será la que contiene las diferentes interfaces y con la que interaccionará el usuario; la capa de negocio, que será la que contenga los procesos para realizar las funciones de la aplicación y comunicará con las otras dos capas; y la capa de persistencia que será aquella que se encargue de administrar los datos de la aplicación.

La principal ventaja de este estilo de desarrollo es la flexibilidad que proporciona, ya que cada capa se ejecuta independientemente, se pueden desarrollar de forma simultá-

nea por equipos distintos y se pueden actualizar o escalar de manera independiente sin afectar al resto de capas.

Durante años según IBM este ha sido el modelo preferido para desarrollar soluciones informáticas. En la actualidad, gracias a los avances en las tecnologías de la nube, se ha facilitado en gran medida la implementación de este modelo.

3.1 Capa de presentación

En esta sección presentaremos los bocetos de la aplicación, esto nos permitirá conocer como interactúa con la aplicación el usuario. A continuación, mostraremos los bocetos que empleamos para desarrollar la aplicación.

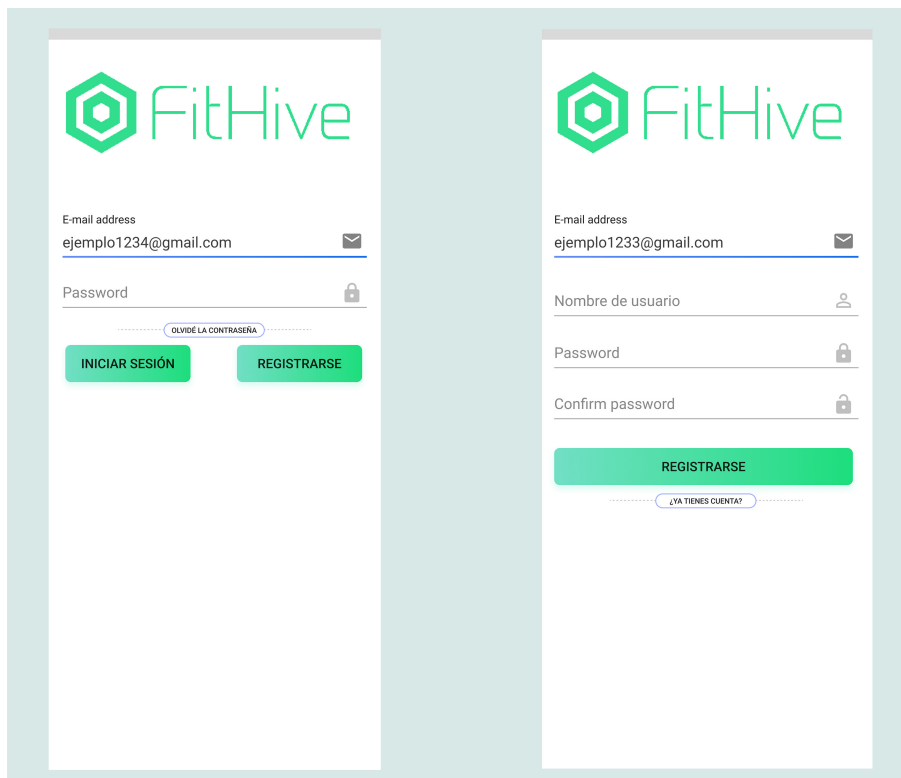
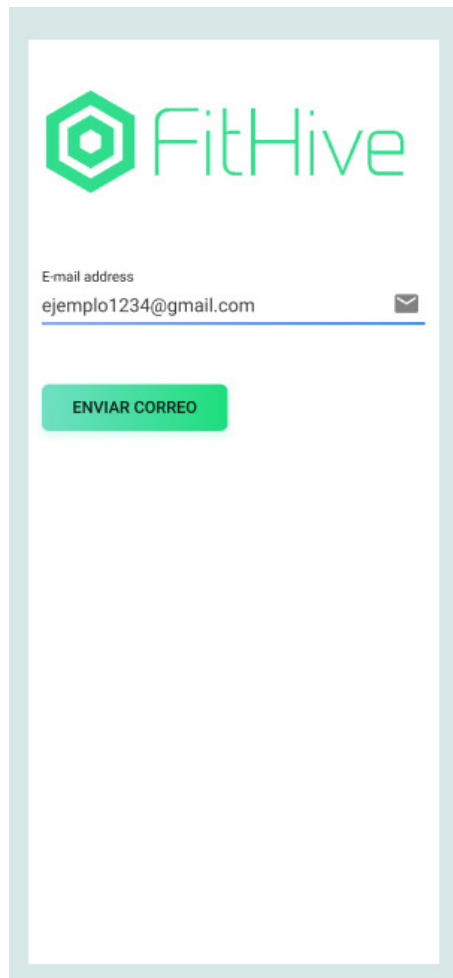


Figura 3.2: Pantalla de inicio de sesión y registro

Comenzaremos por las pantallas de inicio de sesión, que se muestran en la Figura 3.2. La pantalla izquierda incluye opciones para poder registrarnos en la aplicación y para recuperar nuestra contraseña, mientras que la pantalla derecha será la que nos redirigirá para poder rellenar con nuestros datos los campos necesarios y así registrarnos en la aplicación.



The image shows a mobile application screen for password recovery. At the top, there is a green hexagonal logo with a white 'F' inside, followed by the text 'FitHive' in a green, sans-serif font. Below the logo, there is a label 'Email address' in a small, grey font. Underneath the label is a text input field containing the email address 'ejemplo1234@gmail.com'. To the right of the input field is a small grey envelope icon. Below the input field is a solid green button with the text 'ENVIAR CORREO' in white, uppercase letters.

Figura 3.3: Pantalla Olvide la contraseña

En la figura 3.3, tendremos la forma de recuperar la contraseña de nuestros usuarios, con una interfaz muy sencilla en la que solo tendrán que introducir su correo electrónico para poder continuar.

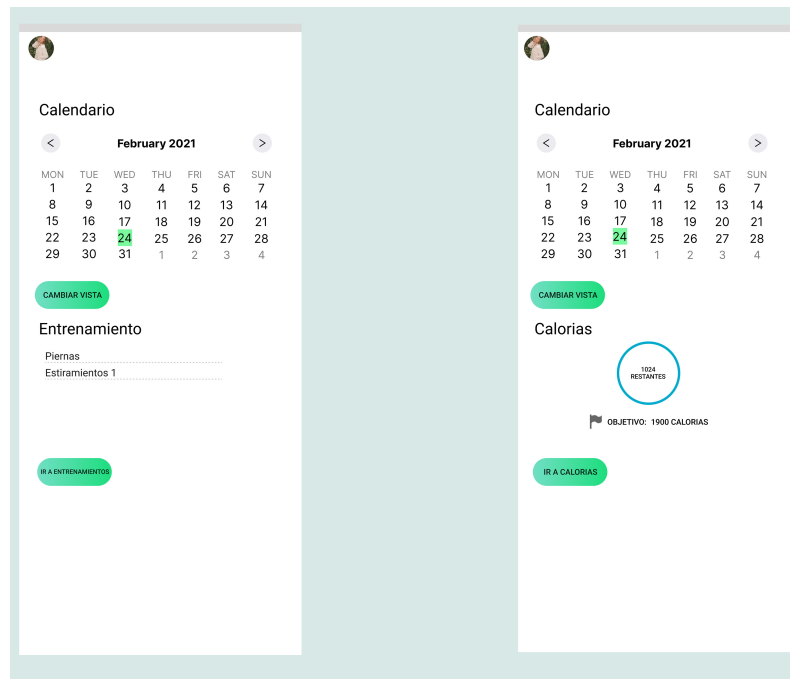


Figura 3.4: Pantalla de inicio

En cuanto iniciemos sesión, la primera vista que tendremos será la figura 3.4, en ella dispondremos de un calendario en el que seleccionando el día podremos ver los entrenamientos que realizamos en ese día. Si pulsamos el botón de cambiar vista, esta cambiará para mostrarnos las calorías restantes del día. También dispondremos de un botón que nos redirigirá a la pantalla de entrenamientos y a la de calorías respectivamente.

Una vez iniciada la sesión, todas las pantallas dispondrán de un botón en la parte superior izquierda con la foto de perfil del usuario. Al pulsarlo, se desplegará un submenú de navegación como el de la figura 3.5, desde el que podremos ir a cualquier apartado de la aplicación.

En la figura 3.6 tenemos la pantalla de estadísticas, que nos permitirá ver el progreso del usuario a lo largo del tiempo, mostrando los datos elegidos en los selectores que se encuentran debajo del gráfico. Las 2 pantallas de la derecha representan la función de medidas corporales. Pulsando en una entrada de la tabla se podrá editar el contenido de la entrada y, si el usuario pulsa el botón, podrá crear una nueva.

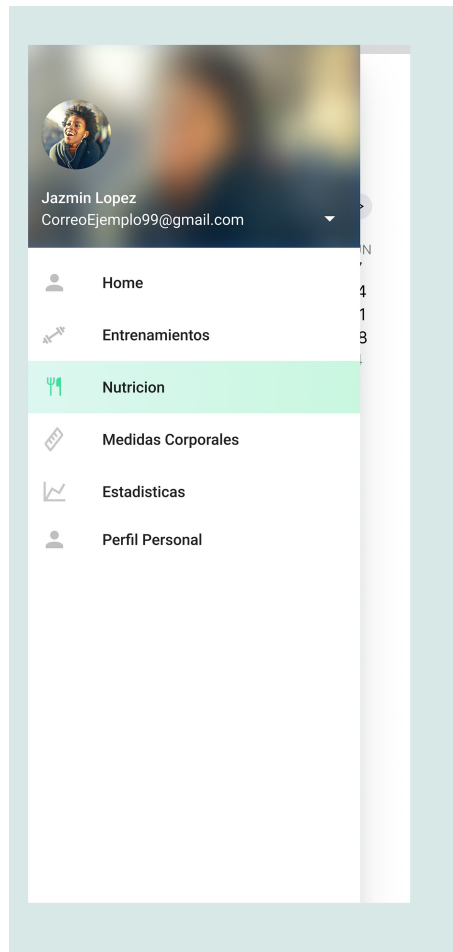


Figura 3.5: Pantalla de submenu

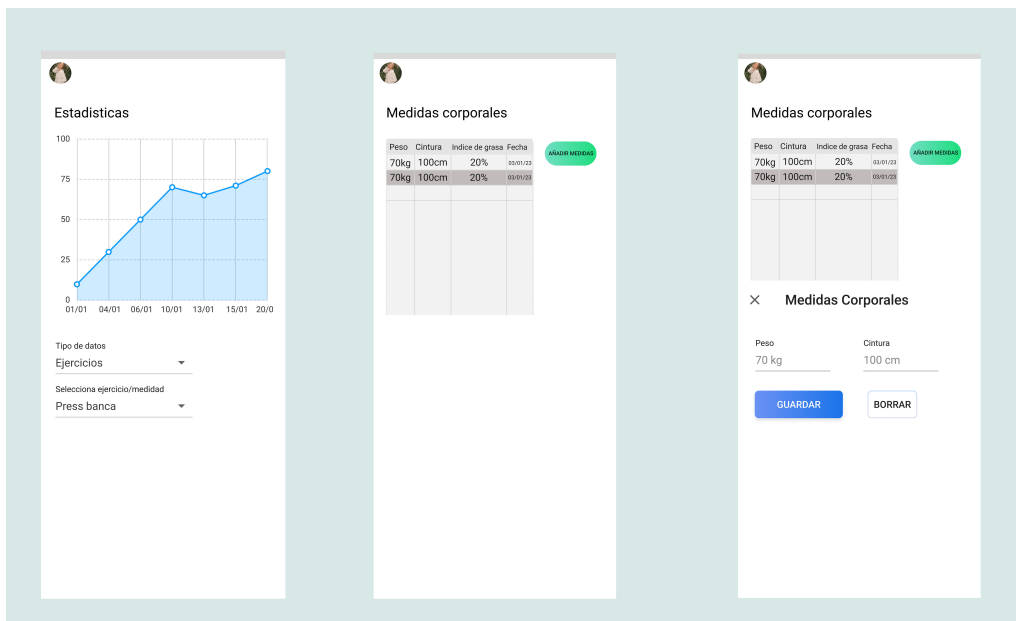


Figura 3.6: Pantallas de medidas corporales y estadísticas

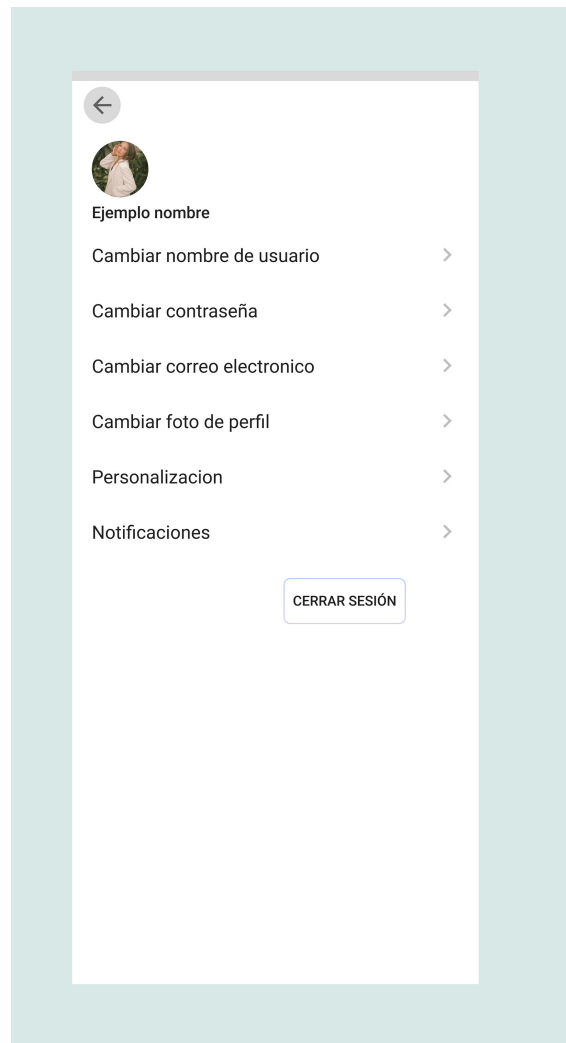


Figura 3.7: Pantalla perfil de usuario

En la figura 3.7 se muestran los ajustes de la aplicación. Desde aquí podremos cambiar tanto los datos de nuestra cuenta, como la configuración de la aplicación. También dispondremos de un botón para cerrar nuestra sesión. En la figura 3.8 mostraremos los diferentes apartados que se podrán acceder desde el menú de la figura 3.7.

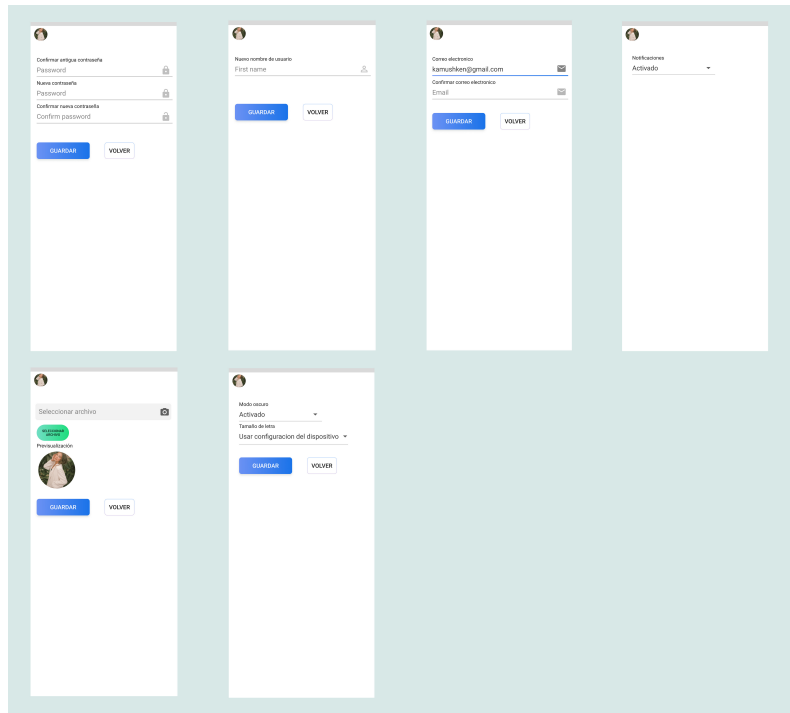


Figura 3.8: Pantallas de opciones del perfil

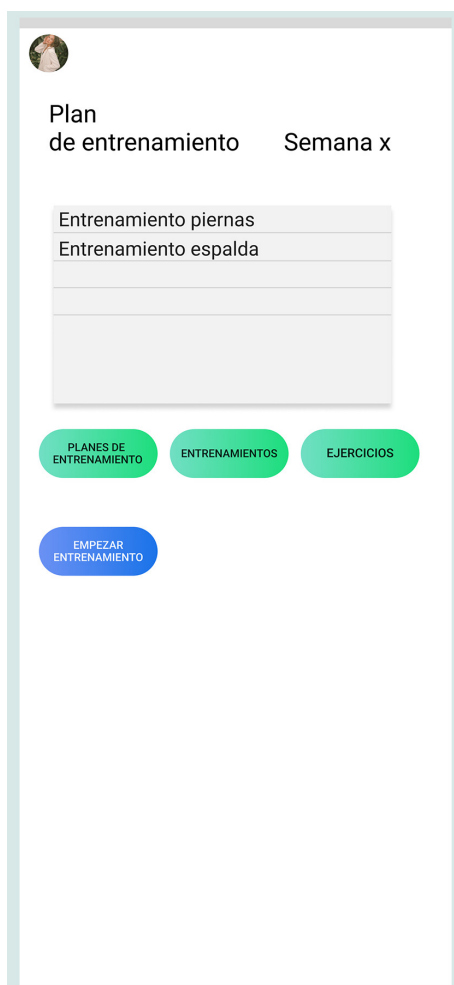


Figura 3.9: Pantallas de inicio entrenamientos

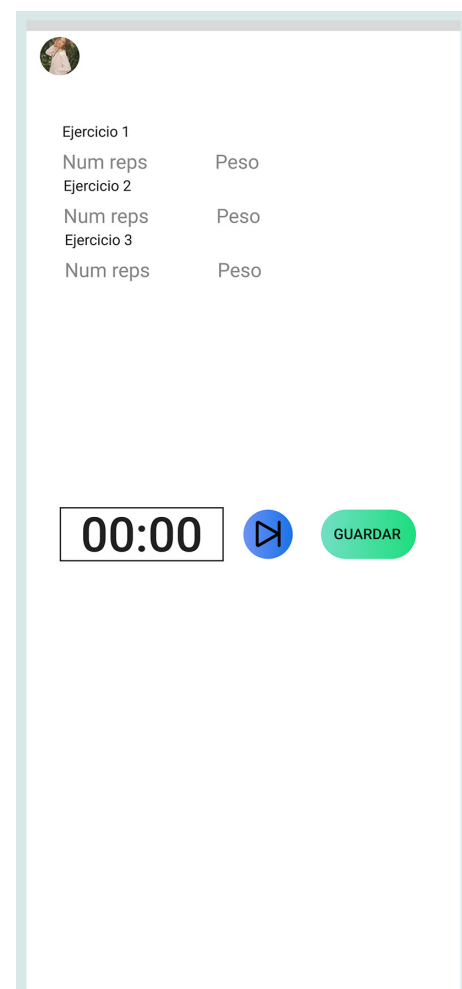


Figura 3.10: Pantallas de realizar entrenamientos

En la figura 3.9, tenemos el inicio de la sección de entrenamientos. En esta se nos mostrará una tabla con los entrenamientos que tenemos pendientes y el número de semana del plan en que nos encontramos. Desde aquí podremos navegar al creador de planes, a la biblioteca de entrenamientos, a la biblioteca de ejercicios y a la pantalla de realizar entrenamientos.

La figura 3.10, muestra la pantalla de realizar entrenamientos, en la que dispondremos de un cronómetro y una lista de los ejercicios que tendremos que hacer en este entrenamiento a los que podremos apuntar el peso y el número de repeticiones que realizamos.

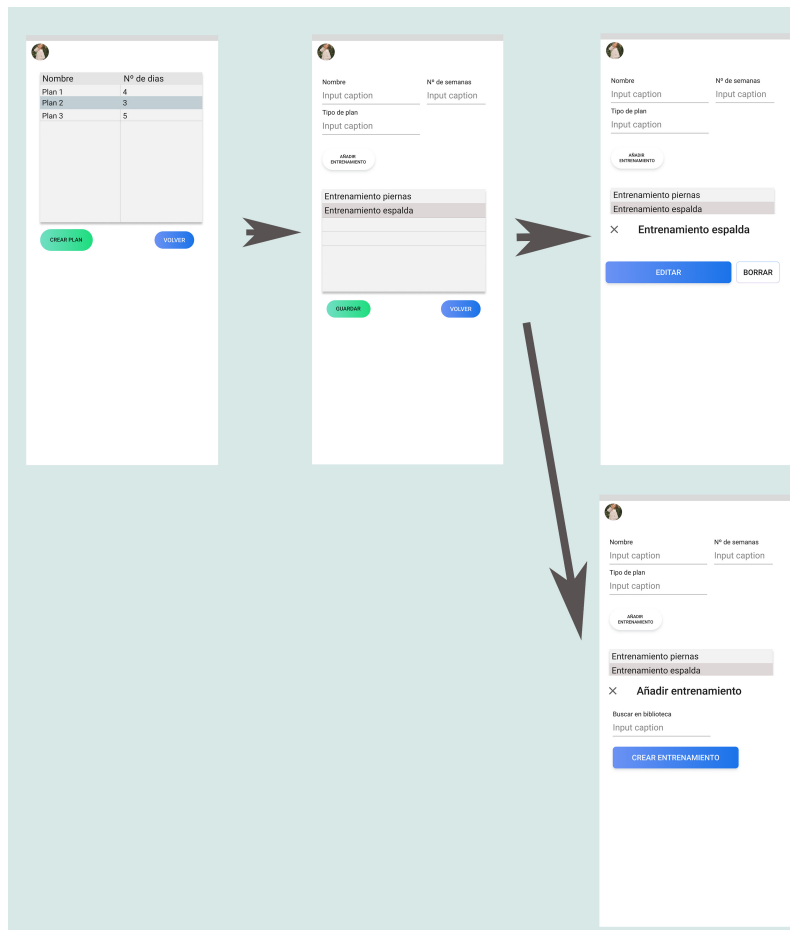


Figura 3.11: Pantallas de creación de planes

En la figura 3.11, mostramos el flujo para crear un plan de entrenamiento. En la primera pantalla podremos editar alguno de los que ya estén registrados o podremos crear uno con el botón de crear plan. En la siguiente pantalla podremos rellenar los datos de nuestro plan, podremos editar los entrenamientos que agreguemos, y tendremos la opción tanto de crear un entrenamiento desde cero como de añadir uno que ya tengamos en nuestra biblioteca.

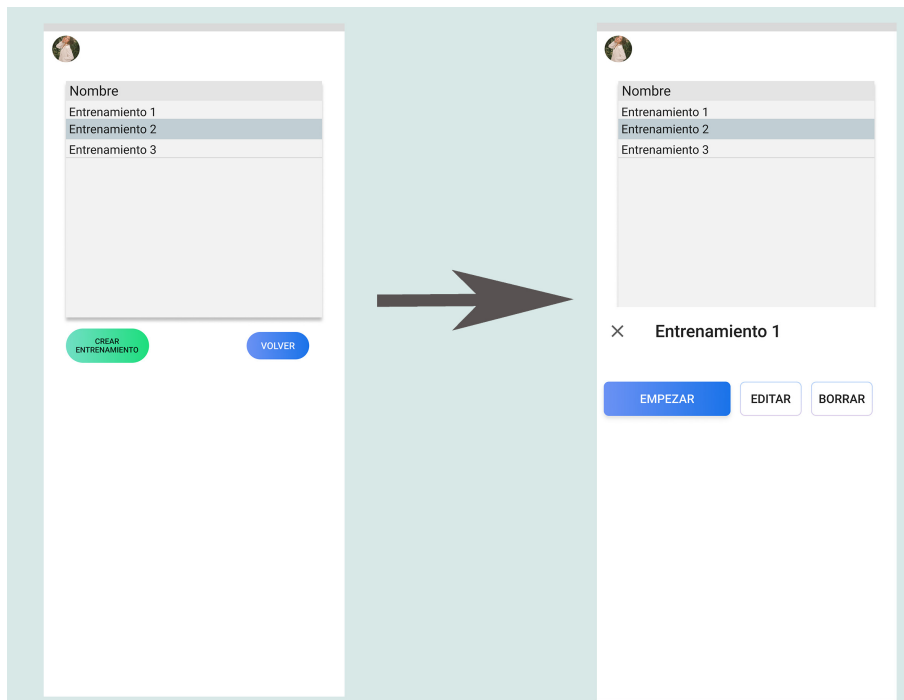


Figura 3.12: Pantallas biblioteca de entrenamientos

En la figura 3.12, mostramos el flujo de la biblioteca de entrenamientos. Desde ella podremos realizar o editar los entrenamientos que estén guardados o crear uno desde cero. En la figura 3.13 mostraremos el flujo de creación de entrenamientos.

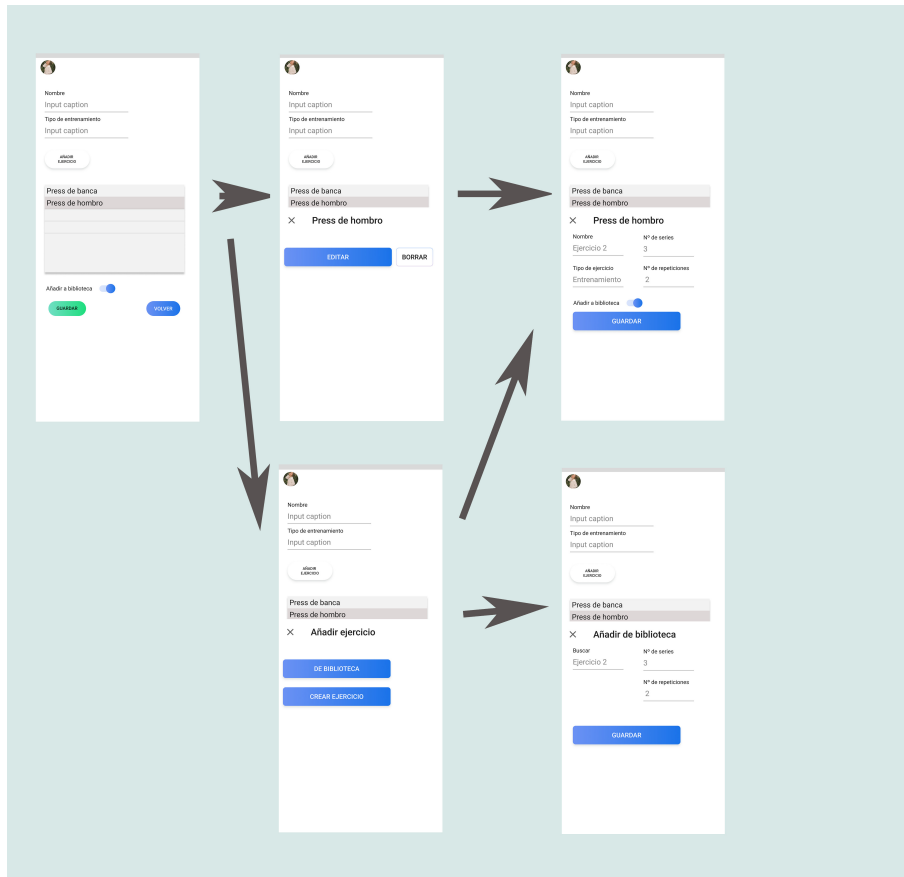


Figura 3.13: Pantallas creación de entrenamientos

Se podrá acceder a la creación de entrenamientos desde el creador de planes y desde la biblioteca de entrenamientos. Aquí podremos rellenar los datos de estos, cuenta con la capacidad de añadir ejercicios de la biblioteca o crearlos nosotros mismos, también puedes editar y eliminar los que ya estén agregados.

La figura 3.14 incluye el flujo de las pantallas de la biblioteca de ejercicios, desde las que podremos editar, crear o borrar los ejercicios que deseemos.

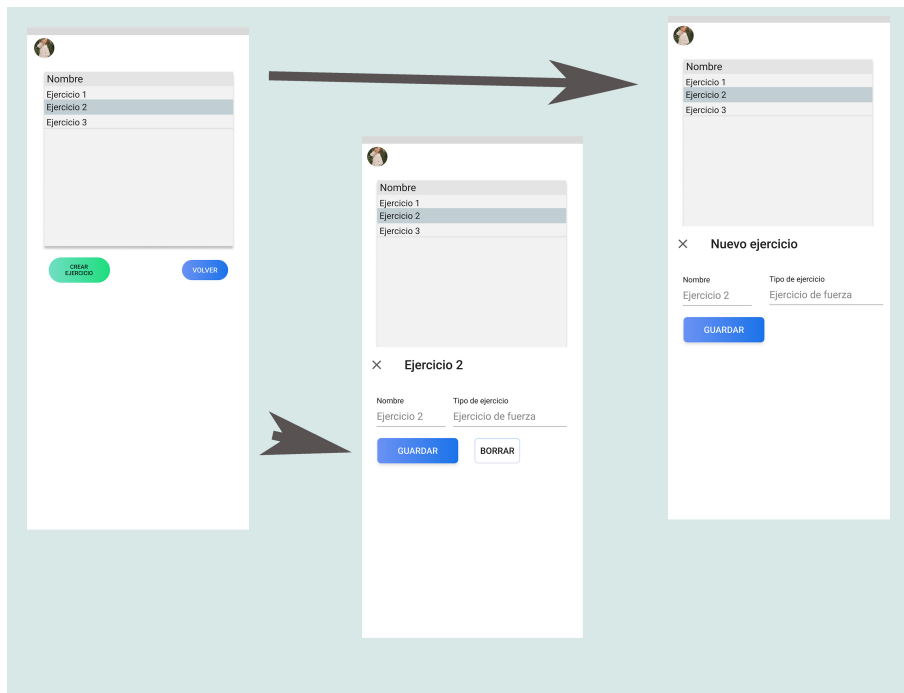


Figura 3.14: Pantallas biblioteca de ejercicios

A continuación mostraremos el flujo de las pantallas de la sección de nutrición de la aplicación. En la figura 3.15 incluimos la pantalla de inicio y la pestaña de cambio de objetivos, en la que podremos ver el objetivo de calorías diarias que hemos seleccionado y podremos cambiarlo fácilmente con la calculadora incluida en la aplicación.

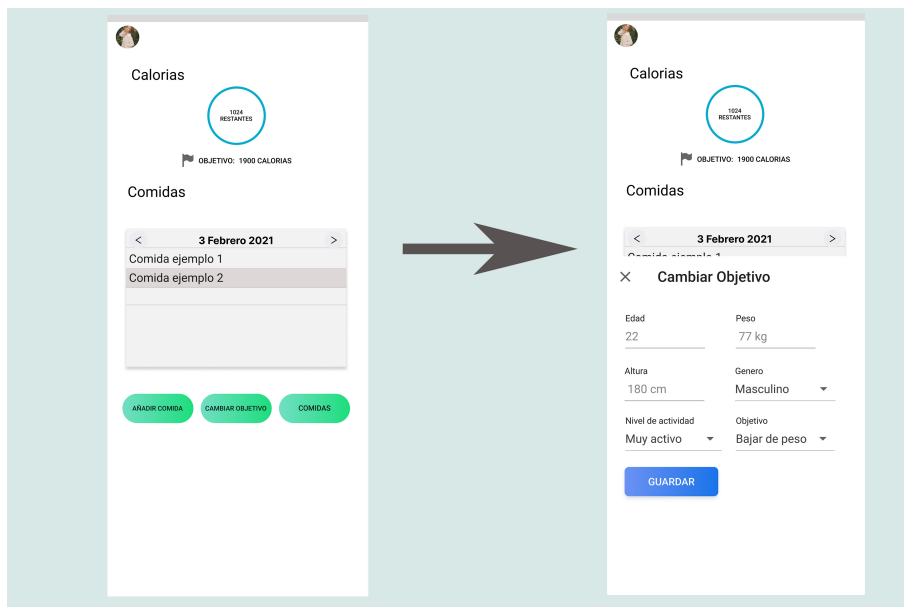


Figura 3.15: Pantallas inicio nutrición

Desde la pantalla de inicio de la sección podremos también añadir comidas al día y también podremos ver la biblioteca con las comidas que hemos ido almacenando. En la figura 3.16, mostraremos el flujo de la aplicación para realizar este proceso.

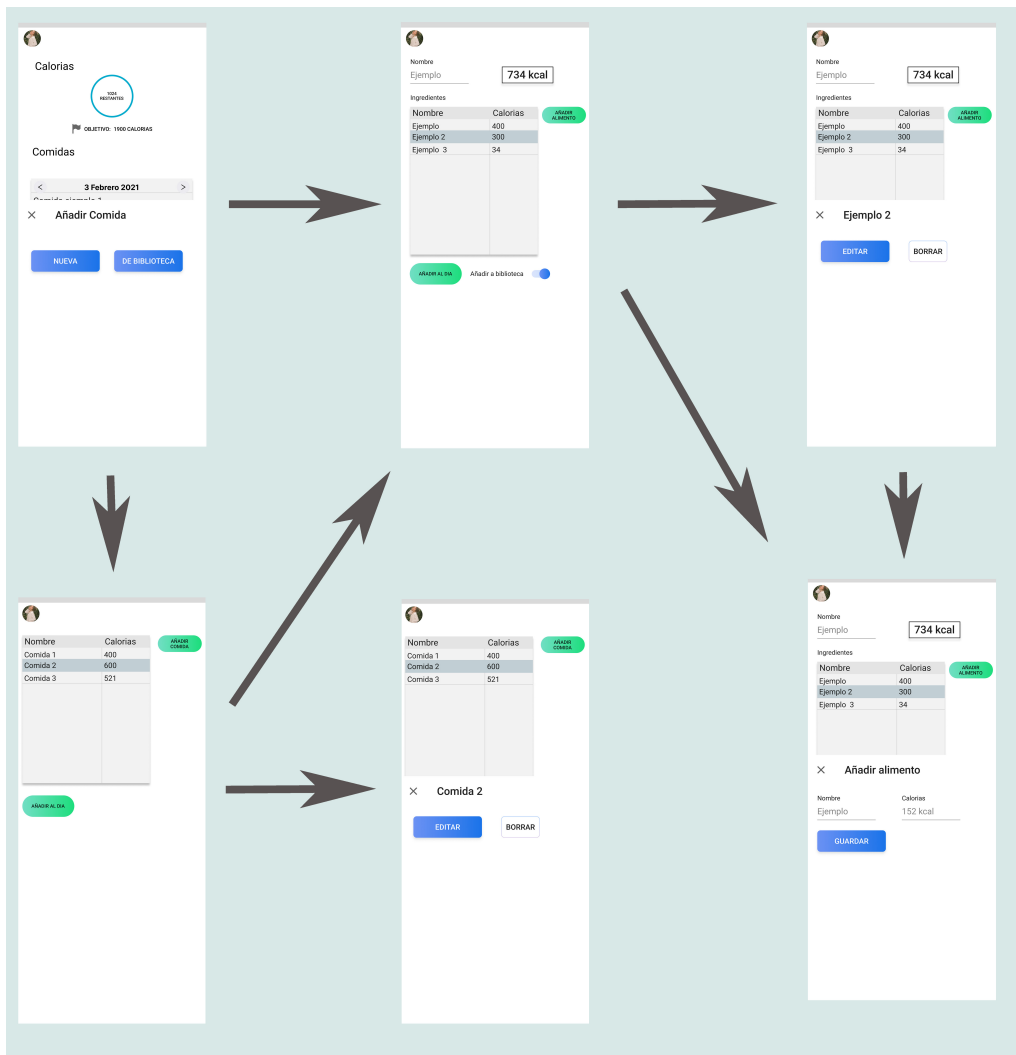


Figura 3.16: Pantallas flujo nutrición

Podremos añadir comidas a nuestro día creándolas al momento o seleccionándolas desde la biblioteca. En la pantalla de creación de comidas podremos añadir los ingredientes y la aplicación calculará de manera automática las calorías de esta. Tras esto, al añadirla al día, se nos añadirá al progreso de calorías diarias.

En resumen, la capa de presentación de nuestra aplicación se ha diseñado para ser intuitiva y eficaz, facilitando al máximo la interacción del usuario con la aplicación.

Los bocetos presentados proporcionan una representación visual de cómo se plasmarán estas funcionalidades en la interfaz de usuario. Estos bocetos son solo una primera aproximación y es posible que durante el proceso de implementación se realicen algunos ajustes para mejorar la experiencia del usuario o para adaptarse a limitaciones técnicas que puedan surgir.

El conjunto de estos elementos forman la base de nuestra capa de presentación, definiendo cómo los usuarios interactuarán con nuestra aplicación y cómo se presentará la información a ellos. En las siguientes secciones, abordaremos los detalles de las otras capas de la aplicación, donde describiremos cómo se gestionará la lógica de negocio y cómo se manejarán los datos de la aplicación.

3.2 Capa de lógica de negocio

Si bien la capa de presentación brinda una interfaz visual y la capa de datos se encarga del almacenamiento y gestión de la información, es en la capa de negocio donde realmente reside la esencia de las operaciones, encargándose de aplicar las reglas y lógicas que definirán cada interacción del usuario. Esta capa no solo asegura que la aplicación funcione de acuerdo a las expectativas y requerimientos establecidos, sino que también garantiza que todas las acciones se realicen de manera coherente, segura y eficiente. En esta sección desentrañaremos las operaciones y los componentes de la capa de negocio en nuestra aplicación, para proporcionar una visión clara de cómo funciona y cómo interactúa con el resto de componentes del sistema.

Empezaremos la sección mostrando el diagrama de casos de uso representado en la figura 3.17, después se describirán brevemente los casos de uso que estarán incluidos en el apéndice B.

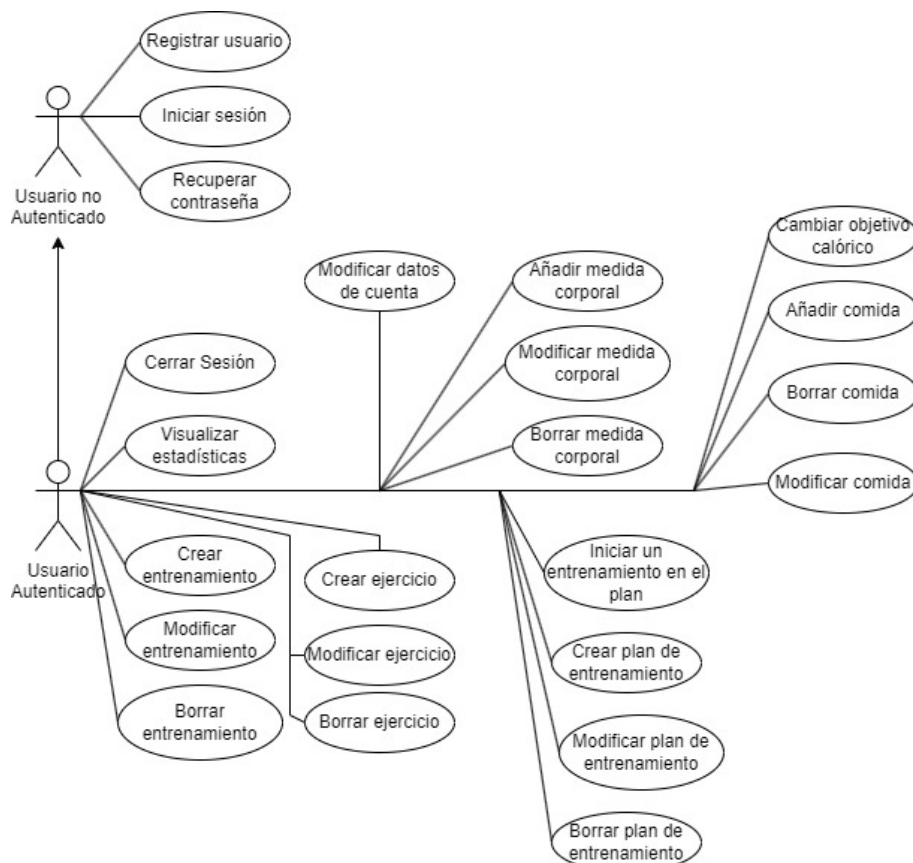


Figura 3.17: Pantallas biblioteca de ejercicios

Como bien indica la figura 3.17, existen dos actores: el usuario no autenticado que será aquel que aún no haya iniciado sesión en la aplicación; este solamente podrá registrarse en la aplicación, iniciar sesión y recuperar la contraseña de su cuenta. Por otro lado, se encuentra el usuario autenticado que será aquel que haya iniciado sesión en la aplicación; este podrá acceder a todas las funcionalidades que ofrece esta (crear entrenamientos, planes, añadir medidas corporales, etc.).

En el apéndice B podremos observar los siguientes casos de uso con más detalle. A continuación, describiremos de manera breve cada uno de ellos:

- **CU-01:** En este caso de uso el usuario creará una cuenta en la aplicación usando su correo electrónico.
- **CU-02:** En este caso de uso el usuario se autenticará usando su correo electrónico y contraseña.
- **CU-03:** En este caso de uso el usuario recuperará la contraseña de su cuenta mandando un correo a su correo electrónico desde la aplicación.
- **CU-04:** En este caso de uso el usuario cerrará su sesión en la aplicación.
- **CU-05:** En este caso de uso el usuario registrará sus medidas corporales (peso, altura, anchura de cintura y cuello) en la aplicación para así calcular su índice de grasa corporal.
- **CU-06:** En este caso de uso el usuario modificará una entrada de medidas corporales de la aplicación.
- **CU-07:** En este caso de uso el usuario borrará una entrada de medidas corporales de la aplicación.
- **CU-08:** En este caso de uso el usuario observará el progreso realizado en sus ejercicios o en sus medidas registradas a lo largo del tiempo.
- **CU-09:** En este caso de uso el usuario modificará los datos de su cuenta (nombre, contraseña, correo).
- **CU-10:** En este caso de uso el usuario registrará en el día los datos de un entrenamiento (número de repeticiones, pesos en cada serie).
- **CU-11:** En este caso de uso el usuario creará un plan de entrenamiento que incluirá una lista de entrenamientos, nombre del plan y el número de semanas que va a tener dicho plan.
- **CU-12:** En este caso de uso el usuario modificará un plan de entrenamiento ya registrado.
- **CU-13:** En este caso de uso el usuario borrará un plan de entrenamiento ya registrado.
- **CU-14:** En este caso de uso el usuario creará un entrenamiento que incluirá una lista de ejercicios, el nombre que le quiera dar al mismo y el tipo de entrenamiento que será (fuerza, calentamiento, estiramientos).
- **CU-15:** En este caso de uso el usuario modificará un entrenamiento ya registrado.
- **CU-16:** En este caso de uso el usuario borrará un entrenamiento ya registrado.
- **CU-17:** En este caso de uso el usuario creará un ejercicio indicando el nombre y el tipo de ejercicio.
- **CU-18:** En este caso de uso el usuario modificará un ejercicio ya registrado.
- **CU-19:** En este caso de uso el usuario borrará un ejercicio ya registrado.
- **CU-20:** En este caso de uso el usuario introducirá sus datos (edad, peso, altura, género, nivel de actividad física, objetivo) y la aplicación calculará el objetivo de calorías que deberá seguir.

- **CU-21:** En este caso de uso el usuario creará una comida indicando el nombre y la lista de ingredientes (nombre de ingrediente, calorías de cada ingrediente), dependiendo de lo que elija el usuario se guardará en el día actual, en la biblioteca de alimentos o en ambos.
- **CU-22:** En este caso de uso el usuario modificará una comida ya registrada.
- **CU-23:** En este caso de uso el usuario borrará una comida ya registrada.

3.3 Capa de datos

En esta sección, se abordará la arquitectura de la base de datos, detallando su estructura organizativa. Además, se expondrán las medidas y protocolos adoptados para garantizar la seguridad de los datos en nuestra aplicación, atendiendo a los estándares contemporáneos en informática. La comprensión de estos elementos es fundamental para asegurar la eficiencia y fiabilidad de cualquier aplicación móvil.

La base de datos de nuestra aplicación está desarrollada con *realtime database*; en nuestro caso concreto, será una base de datos NoSQL. La información se almacenará en objetos con campos, similar al formato JSON. La estructura de esta base de datos se muestra en la figura 3.18. Al emplear esta tecnología, obtenemos numerosas ventajas, como señala el equipo de PandoraFMS [5]. Nos permitirá manejar grandes cantidades de datos sin provocar cuellos de botella, son bases de datos con gran escalabilidad, se adaptan más fácilmente a las necesidades de los proyectos que los modelos de entidad-relación y pueden ejecutarse en máquinas con pocos recursos. Debido a estas razones, hemos optado por este tipo de bases de datos para nuestra solución.

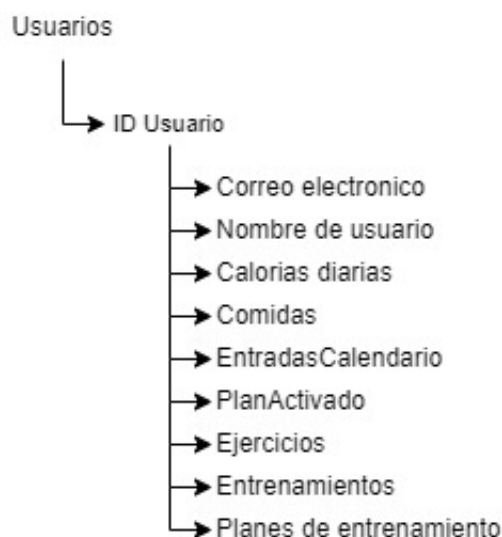


Figura 3.18: Estructura de la base de datos

La figura 3.18 es un diagrama simplificado de la estructura de la base de datos. En los 3 primeros campos serán de tipo *string* y nos servirán para almacenar objetos con un solo valor. El resto de campos son listas que contendrán los distintos objetos que creará nuestra aplicación.

A continuación, en las figuras 3.19 y 3.20 mostraremos la estructura interna de las listas de entrenamientos, ejercicios, planes de entrenamiento, comidas, entradas del calendario y del plan que se encuentra activado para el usuario.



Figura 3.19: Estructura de las funciones de entrenamiento



Figura 3.20: Estructura de las funciones de comida, plan activado y entradas de calendario

En la figura 3.19 podemos observar las diferentes estructuras de cada lista de objetos. En la primera, “planes de entrenamiento”, dispondrá de varios campos para identificar el plan. Luego, tendrá un campo de tipo *booleano* llamado “activado” para indicar qué plan es el que está usando el usuario en este momento. Por último, también contará con una lista de entrenamientos con el mismo formato que la lista general de entrenamientos que incluye cada usuario. La lista de entrenamientos general incluirá 3 campos de tipo string para identificar a cada entrenamiento. Cabe destacar el campo “plan de entrenamiento”, que nos indicará si este entrenamiento está incluido en algún plan creado por el usuario. Los entrenamientos también incluirán una lista de ejercicios que tendrá una estructura distinta a la de la lista general de ejercicios, ya que en esta añadiremos los valores de peso y repeticiones.

En la figura 3.20 disponemos de las estructuras de las listas restantes. Comenzaremos describiendo “PlanActivado”; esta indicará a la aplicación qué plan es el que está realizando el usuario y modificará su estructura para adecuarla al uso que le dará la aplicación. Después, estará la lista de comidas, en la cual los dos primeros campos servirán para identificar a la comida, y el último será una lista de ingredientes en la cual cada entrada tendrá dos campos: uno tipo *string* y otro tipo *integer*. Por último, comentaremos la lista de “EntradasCalendario”; en esta se almacenarán los datos de cada día que el usuario registre: un entrenamiento o una comida. La estructura de las dos listas será la misma que sus correspondientes listas generales.

Estas estructuras representan la información que se almacena en el servidor. La aplicación móvil solamente consulta la información y la modifica según indique el usuario; no se almacena nada localmente en el dispositivo donde se instale la aplicación.

La seguridad de los datos es un tema muy importante para cualquier aplicación, por eso, la base de datos estará protegida de tal manera que cada usuario registrado en la aplicación solo podrá leer o modificar los datos de su cuenta.

CAPÍTULO 4

Implementación

En este capítulo entraremos en detalle en el desarrollo de nuestra solución. Primero describiremos las tecnologías que hemos empleado para desarrollarla, así como el razonamiento detrás de cada elección. Profundizaremos en la estructura interna de la aplicación, desglosando cada capa para ofrecer una visión más detallada de cómo interactúan los diferentes componentes entre sí. Por último, discutiremos los detalles específicos de la implementación, como la organización de paquetes, la gestión de actividades o la navegación.

A través de este apartado, esperamos ofrecer una visión clara y comprensible de las decisiones técnicas tomadas en el desarrollo de nuestra aplicación y las razones que nos llevaron a ellas.

4.1 Estudio de tecnologías y herramientas empleadas

Empezaremos analizando las diferentes tecnologías y herramientas empleadas durante el desarrollo.

4.1.1. Figma

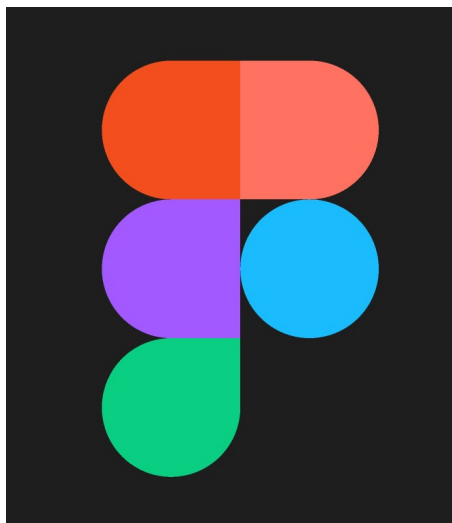


Figura 4.1: Logotipo de Figma

Figma¹, cuyo logotipo se muestra en la figura 4.1, es una plataforma diseñada para crear interfaces para aplicaciones informáticas, tanto web como móvil. Es una de las herramientas más utilizadas en la actualidad por los diseñadores, ya que, al estar basada en navegador, permite trabajar en la nube desde cualquier dispositivo. Además, es posible añadir al equipo de trabajo al proyecto para trabajar simultáneamente, similar a cómo se trabaja en un documento de Google Docs. Figma también es compatible con *plugins* creados tanto por los desarrolladores como por los usuarios, lo que facilita una amplia gama de herramientas que agilizan el desarrollo. Asimismo, su capacidad para exportar en diferentes formatos de imagen, como .jpg o .png, es sumamente útil.

Optamos por esta herramienta para diseñar las interfaces de nuestra aplicación debido a las características mencionadas anteriormente: la usabilidad de sus herramientas, la vasta cantidad de *plugins* creados por la comunidad y la portabilidad que ofrece al trabajar desde la nube nos convencieron de nuestra elección.

Por último, es importante señalar que esta herramienta tiene una versión de pago que ofrece numerosas ventajas orientadas a empresas más grandes con diversos proyectos simultáneos. Sin embargo, la versión gratuita resulta más que adecuada para el usuario promedio.

4.1.2. Locofy



Figura 4.2: Logotipo de Locofy

Locofy², cuyo logotipo está representado en la figura 4.2, es una herramienta en la nube diseñada para ayudar a los desarrolladores a transferir sus diseños a código listo para producción. Gracias a la inteligencia artificial, tiene la capacidad de convertir diseños de Figma, Adobe XD y herramientas similares en múltiples tecnologías, tales como React, React Native, HTML-CSS, Gatsby, Next.js, entre otras.

Las principales características de Locofy incluyen la conversión de código mencionada anteriormente, la creación de componentes reutilizables que ahorran tiempo de desarrollo y aumentan la eficiencia, la posibilidad de visualizar los diseños y comprobar su funcionalidad en tiempo real. Además, su integración con GitHub facilita un control de versiones de manera sencilla e intuitiva.

La transformación de los diseños de Figma en código útil se logra a través de un *plugin* gratuito diseñado por el propio equipo de Locofy. Con este *plugin*, etiquetamos cada

¹<https://www.figma.com>

²<https://www.locofy.ai>

componente de nuestro diseño para especificar la función que deseamos que desempeñe en nuestra aplicación. Luego, con tan solo pulsar un botón, Locofy convierte el diseño en código que requiere mínimas modificaciones.

Optamos por utilizar esta herramienta por varias razones. Nuestros diseños fueron realizados en Figma y, debido a la complejidad de la aplicación, contábamos con una vasta cantidad de pantallas que programar. Con la ayuda de Locofy, pudimos llevar a cabo este proceso de manera rápida y con pocos inconvenientes, teniendo solo que adaptar el código para añadir funcionalidades adicionales a la aplicación.

4.1.3. Github

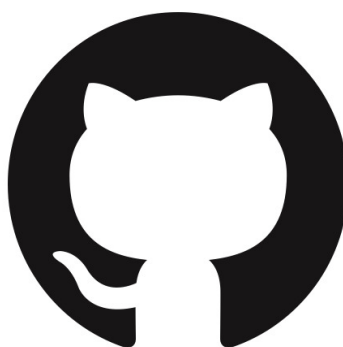


Figura 4.3: Logotipo de Github

Para el desarrollo de nuestra aplicación, requeríamos de una plataforma robusta para el almacenamiento del código y el control de versiones. Por esta razón, nos decantamos por GitHub³, cuyo logotipo se encuentra en la figura 4.3. GitHub es una herramienta que facilita a los programadores la colaboración en proyectos, permitiendo mantener un registro detallado de las modificaciones del código, lo cual agiliza la cooperación y monitoreo del avance del proyecto.

La esencia de su operatividad radica en la creación de repositorios, espacios designados para albergar todos los archivos y el historial de revisiones del proyecto. Los usuarios tienen la capacidad de clonar repositorios a sus máquinas locales o realizar una copia personal de un repositorio ajeno con el propósito de trabajar en variantes propias. GitHub proporciona un sistema de ramificación para implementar pruebas o desarrollar nuevas funcionalidades sin interferir con el código base. Una vez concluido el desarrollo en una rama, ésta puede ser fusionada con la rama principal, comúnmente denominada *main*. Adicionalmente, GitHub ofrece integración con múltiples aplicaciones externas, como el ya mencionado Locofy. Además, presenta herramientas de control de acceso al repositorio y mecanismos para automatizar pruebas y despliegues directamente desde la plataforma.

Como es evidente, GitHub se consolida como una herramienta imprescindible. La sinergia entre el control de versiones de Git y las capacidades de colaboración ofrecidas por GitHub propician un entorno de trabajo eficiente, conservando un historial estructurado de toda la labor realizada.

³<https://github.com>

4.1.4. Visual studio code

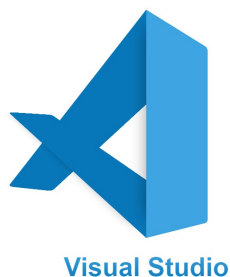


Figura 4.4: Logotipo de Visual studio code

Visual Studio Code⁴, cuyo logotipo se presenta en la figura 4.4, fue desarrollado por Microsoft. Ha surgido en los últimos años como uno de los editores de código más populares y preferidos en la comunidad de desarrollo. Esta popularidad no es innecesaria, ya que combina una interfaz amigable y personalizable con una serie de características innovadoras que describiremos más adelante, permitiéndole adaptarse a una amplia gama de lenguajes de programación y plataformas. Su naturaleza extensible, gracias a un vasto mercado de extensiones, lo convierte en una herramienta versátil, capaz de ajustarse a las necesidades específicas de cualquier proyecto, desde el desarrollo web hasta la programación de sistemas.

VS Code brinda una interfaz de usuario clara y eficaz, personalizable en múltiples aspectos, desde la apariencia hasta la disposición de los paneles. Una de las grandes fortalezas de este programa es el soporte para una amplia variedad de lenguajes de programación ofreciendo auto completado inteligente y la posibilidad de añadir incluso más idiomas soportados a través de las extensiones de la comunidad. Las extensiones son, de hecho, una de las características distintivas de VS Code. Existe un vasto mercado alimentado tanto por empresas como por los propios usuarios que permite añadir características nuevas o mejorar las ya existentes, así como personalizar aún más las interfaces. Otro punto a favor de VS Code es una integración nativa para el control de versiones con Git, lo cual facilita las tareas de seguimiento de cambios, la gestión de ramas o demás problemas que pueden surgir en el control de versiones. Una de las tareas esenciales en el desarrollo de software es la depuración. VS Code integra potentes herramientas de depuración que permiten a los desarrolladores identificar y corregir errores de manera eficiente. Este editor incluye una terminal dentro de la propia interfaz que es especialmente útil para tareas como la gestión de paquetes o el control de versiones. Por último con herramientas de navegación rápida, búsqueda global o plegado de código, nos facilita la gestión y navegación por nuestros proyectos.

Nuestra elección de Visual Studio Code se basó en estas capacidades. Su versatilidad y soporte multilenguaje nos permitía trabajar con diversas tecnologías sin la necesidad de cambiar constantemente de entorno. Además La integración con Git, complementada con extensiones para GitHub, optimizó nuestro flujo de trabajo, facilitando el control de versiones y la corrección de problemas.

4.1.5. Android Studio

⁴<https://code.visualstudio.com>



Figura 4.5: Logotipo de Android Studio

Android Studio⁵, cuyo logotipo se muestra en la figura 4.5, es el entorno de desarrollo integrado (IDE) oficial para Android, lanzado por Google en 2014. Basado en el robusto editor de código y el conjunto de herramientas de IntelliJ IDEA⁶, Android Studio proporciona una amplia gama de herramientas y características diseñadas para facilitar no solo el diseño de interfaces, sino también la depuración y optimización de aplicaciones.

Este editor nos ofrece un sistema de compilación avanzado basado en Gradle, acompañado de un emulador eficaz y completo. Proporciona un ambiente unificado para poder desarrollar en todos los dispositivos Android. Permite ediciones en vivo, actualizando los elementos editados en tiempo real tanto en emuladores como en dispositivos físicos. Esta plataforma cuenta con plantillas de código para facilitar la creación de funciones comunes en aplicaciones. Cuenta con integración con GitHub para facilitar el control de versiones. Este editor incluye un amplio conjunto de herramientas de prueba y herramientas Lint para detectar y resolver problemas de rendimiento y usabilidad. Por último, cabe destacar que su integración con Google Cloud Platform simplifica el uso de servicios como Google Cloud Messaging.

Todas estas características hacen que Android Studio sea un referente en el mundo del desarrollo en Android, pero para nuestra solución, empleamos únicamente el emulador integrado para probar nuestra aplicación, el cual aprendimos a usar gracias al libro *Android programming* [6]; su sencilla configuración y su integración con otras plataformas nos permitieron desarrollar nuestra aplicación de manera mucho más cómoda.

4.1.6. Firebase



Figura 4.6: Logotipo de Firebase

Firebase⁷, cuyo logotipo se encuentra en la figura 4.6, es una plataforma para el desarrollo de aplicaciones web y móvil. Comenzó siendo una base de datos en tiempo real; sin embargo, con el paso de los años, se añadieron más y más funciones que sirvieron para agrupar los SDK de productos Google en un mismo sitio.

⁵<https://developer.android.com/studio>

⁶<https://www.jetbrains.com/idea/>

⁷<https://firebase.google.com>

La principal funcionalidad de esta plataforma es hacer más sencilla la creación de aplicaciones web o móviles. Las herramientas que ofrece son variadas y sencillas de utilizar. Firebase, al simplificar el desarrollo del *backend* y al agrupar las tareas de gestión del mismo en una sola plataforma, resulta muy interesante para los desarrolladores que deseen dedicar menos tiempo al *backend*, tanto por cuestiones de desarrollo como por mantenimiento.

Nos centraremos en dos de las herramientas que nos proporciona Firebase: Realtime Database y Autenticación de usuarios. Realtime Database es una de las herramientas esenciales de Firebase; estas bases de datos en tiempo real se alojan en la nube, son NoSQL y almacenan los datos como si fueran objetos en formato JSON. Firebase se encarga de enviar automáticamente los eventos a las aplicaciones cuando cambian los datos. Por otro lado, la autenticación de los usuarios es una herramienta necesaria en la mayoría de las aplicaciones. Firebase pone a nuestra disposición un sistema de autenticación que permite tanto el registro mediante correo electrónico y contraseña como el acceso a través de Facebook, Google o Twitter. Con todo esto, el proceso se ve altamente simplificado, ya que a través de la plataforma, los administradores son capaces de gestionar los accesos y proteger los datos. También cabe destacar que Firebase cuenta con la posibilidad de guardar los datos de inicio de sesión en la nube con total seguridad, de esta forma el usuario no tendrá que identificarse cada vez que se abra la aplicación.

Escogimos Firebase como tecnología para desarrollar el *backend* de nuestra aplicación gracias a las dos herramientas anteriormente explicadas. Queremos que nuestra aplicación comparta datos en tiempo real, por lo tanto, la base de datos que nos proporciona Realtime Database es perfecta para nuestra solución y la autenticación de Firebase nos permite simplificar el desarrollo y gestión de esta funcionalidad sin tener que preocuparnos por la seguridad o el descontento de los usuarios.

4.1.7. React Native

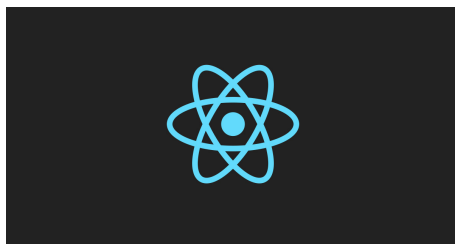


Figura 4.7: Logotipo de React Native

React Native⁸ [7], ilustrado en la figura 4.7, es un marco de desarrollo de código abierto creado por Facebook para la construcción de aplicaciones móviles de alto rendimiento utilizando JavaScript. Al aprovechar las capacidades del lenguaje React⁹, React Native posibilita el desarrollo simultáneo en iOS y Android, reduciendo significativamente los tiempos asociados al desarrollo de aplicaciones móviles multiplataforma. Su amplia comunidad y colección de extensiones y bibliotecas proporcionan a los desarrolladores los recursos y apoyos necesarios durante el ciclo de desarrollo.

Las características más destacadas de este marco de desarrollo serían la capacidad de desarrollar simultáneamente para plataformas iOS y Android. El empleo de componentes nativos para maximizar el rendimiento y la fluidez en la aplicación. La reutilización de código para facilitar una producción más rápida y eficiente. Al poseer un sistema de

⁸<https://reactnative.dev>

⁹<https://es.react.dev>

recarga en caliente permite a los desarrolladores visualizar los cambios que realicen en el código en tiempo real sin tener que recompilar el proyecto entero. Por último la extensa comunidad cuenta con una variedad de bibliotecas que nos permiten ampliar aun más las capacidades y versatilidad de este marco.

Al evaluar diferentes marcos para la creación de nuestra aplicación móvil, React Native emergió como la elección óptima. Su habilidad intrínseca de trabajar eficientemente en plataformas como iOS y Android resultó ser invaluable. Un punto destacado fue la recarga en caliente, que potenció nuestro flujo de trabajo al mostrar cambios de manera inmediata, eliminando la tediosa espera de recompilaciones. Adicionalmente, la solidez del código generado por React Native garantizó un funcionamiento fluido y permitió una reutilización efectiva de componentes a lo largo del proyecto. Esta amalgama de ventajas consolidó nuestra decisión de adoptar React Native como nuestro marco principal.

4.2 Arquitectura Android app

Al estar creada con React Native y con Firebase, nuestra aplicación no seguirá la arquitectura recomendada por los desarrolladores de Android. En esta sección, describiremos cómo están implementadas las tres capas en nuestro código.

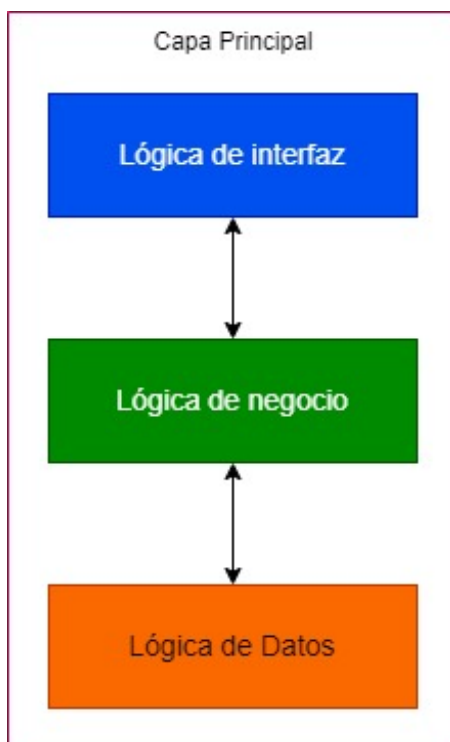


Figura 4.8: Arquitectura de la aplicación.

La arquitectura de nuestra solución está formada por: la lógica de la interfaz, encargada de administrar todos los elementos gráficos que interactuarán con el usuario; la lógica de negocio, que se comunicará tanto con la interfaz como con la lógica de datos; y por último, la lógica de datos, que será la encargada de gestionar el almacenamiento de los datos de la aplicación. Todas estas secciones estarán entrelazadas entre sí, esto es debido a que nuestra aplicación estará formada por clases que representarán a las pantallas de nuestra solución.

Estos archivos estarán divididos en dos secciones: la primera, en la que se incluirán todos los elementos gráficos que se mostrarán al usuario; después, se incluirá un apar-

tado en el que estará definido el comportamiento de los elementos de la interfaz. En muchas ocasiones, cuando la pantalla lo requiera, también se accederá a los datos desde este código.

La lógica de la interfaz, como bien hemos nombrado antes, será la encargada de mostrar al usuario todas las ventanas con las que interactuará. La principal función de esta capa será la de mostrar la información proveniente de la capa de datos en pantalla. Si los datos se actualizan, ya sea por acción del usuario o por algún cambio externo, la interfaz deberá actualizarse para reflejar esos cambios. Esta capa estará formada principalmente por dos tipos de elementos: aquellos que renderizan y muestran los datos en la pantalla y aquellos que controlan la lógica de los datos, comunicándose con la capa de datos.

La lógica de negocio será la encargada de definir qué comportamiento tendrán todos los elementos que conforman la capa de la interfaz y también se encargará de pedir los datos convenientes a la capa de datos, transformándolos de la manera que sea necesaria para poder mostrarlos en la interfaz. Esto lo hará a través de funciones implementadas dentro de cada una de las pantallas.

Por último, la capa de datos será la que contenga los recursos necesarios para acceder al origen de los datos, en nuestro caso particular, la base de datos de Firebase. En nuestra solución, el patrón de diseño que hemos empleado para el DAO (objeto de acceso a datos) nos permite separar la lógica de acceso a datos de la lógica de negocios. El DAO actuará como intermediario entre la aplicación y la fuente de nuestros datos. Al estar diseñado de esta forma, podremos mantener un código más limpio y organizado, ya que la lógica de negocio no necesita saber cómo se accede o se guarda la información. El código que hemos diseñado para nuestro DAO podrá ser reutilizado en las diferentes partes de la aplicación.

Si por alguna razón tuviéramos que modificar la estructura de la base de datos, solamente habría que modificar el código del DAO sin tener que cambiar la forma en la que se comporta.

4.3 Implementación de la aplicación

En este apartado, describiremos el proceso de desarrollo de nuestra solución, comentaremos los problemas con los que nos hemos encontrado y la forma en la que los hemos sorteado. También describiremos cómo hemos organizado nuestra aplicación y explicaremos en profundidad los procesos que contiene.

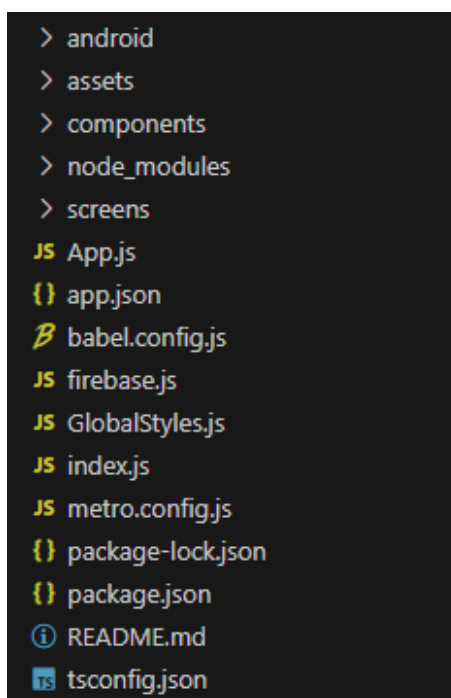
4.3.1. Obtención del código inicial de la aplicación

Al crear el proyecto con la herramienta Locofy, se creó con la herramienta Expo, la cual nos permitía desarrollar nuestra aplicación, pero no nos otorgaba acceso a los archivos nativos de Android. No tener acceso a estos archivos provocó que muchos de los módulos que teníamos pensado usar, como por ejemplo, aquellos que nos servirían para conectar nuestra aplicación con Firebase, no fuera posible configurarlos. Por ello, tuvimos que transformar nuestro proyecto para que tuviera la estructura y configuración de un proyecto básico de React Native. Para lograrlo, simplemente tuvimos que ejecutar un proceso de Expo llamado *Ejectin*. Para lograrlo, solamente tendríamos que ejecutar la línea de código referenciada en 4.1 y seguir las indicaciones del CLI para terminar de configurarlo.

```
1 expo eject  
2 npx react-native run-android
```

Listing 4.1: Ejection en expo

Una vez que logramos desarrollar nuestra aplicación para que operase adecuadamente en dispositivos Android, surgió el cuestionamiento sobre la posibilidad de desarrollar simultáneamente una versión para iOS. No obstante, tras un detenido análisis, decidimos que la versión inicial no contaría con soporte para iOS. La falta de dispositivos físicos disponibles para pruebas fue un factor determinante en esta decisión. Si bien podríamos haber recurrido al uso de emuladores para este sistema operativo, nuestros estándares de calidad exigían pruebas exhaustivas tanto en entornos emulados como en dispositivos físicos. Además, desarrollar en paralelo para iOS representaría un considerable aumento en el tiempo requerido, un coste que consideramos demasiado elevado para abordar en esta etapa del proyecto.

4.3.2. Estructura del código**Figura 4.9:** Estructura de archivos del proyecto

En la figura 4.9 se ilustra la estructura de nuestro proyecto. Dentro de la carpeta raíz, es apreciable la presencia de múltiples archivos de configuración. Como ejemplo, se tiene el archivo *GlobalStyles.js*, que se destina a determinar los estilos globales de la aplicación, tales como tamaño de letra, tipo de fuente y colores de diversos componentes. El conjunto de archivos restantes desempeña funciones de ajuste de parámetros o configuración de paquetes/módulos del proyecto, siendo este el caso de *package.json* y *package-lock.json*. Las funcionalidades específicas de los archivos *app.js* y *firebase.js* serán abordadas con mayor detalle posteriormente en este apartado.

El primer directorio con el que nos topamos es *android*. En él, se resguardan todos los archivos nativos de la plataforma Android. Existen ocasiones en las que se requiere intervenir en configuraciones particulares de la plataforma. En este directorio, se hallan archivos y configuraciones esenciales para la construcción y despliegue de la aplicación en dispositivos y emuladores Android. En caso de requerir permisos especiales, integra-

ción de SDKs específicos de Android o realizar modificaciones a nivel nativo, será en este directorio donde dichas acciones se efectúen.

Procediendo, encontramos la carpeta *assets*. Esta se designa para albergar archivos de imágenes y recursos similares, como audios o vídeos. Al disponer de los recursos en este directorio, garantizamos su accesibilidad en cualquier instancia de la aplicación. Tal organización promueve una óptima gestión y actualización de estos recursos.

El directorio *components* está consagrado a contener fragmentos de código reutilizables, los cuales React denomina como componentes. Al segmentar los componentes en un directorio propio, se promueve la mantención de un código pulcro y modular, potenciando así la legibilidad del mismo y garantizando su reutilización.

Por su parte, el directorio *node_modules* tiene la misión de contener todas las bibliotecas y dependencias vitales para el correcto funcionamiento del proyecto. Cuando se instala una nueva biblioteca o paquete mediante *npm* o *yarn*, es en este directorio donde se descargan y almacenan. Debido a la naturaleza modular de la programación moderna en JavaScript, es habitual contar con una plétora de dependencias que abarcan desde utilidades diminutas hasta *frameworks* extensos. No obstante, pese a su relevancia, es un directorio que rara vez requiere intervención directa por parte de los desarrolladores.

Finalmente, el directorio *screens* es el destinado para albergar las pantallas, que constituyen los elementos con los cuales los usuarios interactúan, tales como la página de inicio o la página de registro. Dichas pantallas incorporan múltiples componentes para proporcionar al usuario una experiencia integral. Al reunir estas pantallas en un único directorio, se facilita a los desarrolladores mantener una perspectiva clara de la estructura de la aplicación.

4.3.3. Navegación

A continuación, abordaremos la gestión de navegación entre pantallas en nuestra solución. El primer archivo en consideración es *app.js*. Dado que es el primer archivo que se ejecuta al iniciar la aplicación, es en él donde se configura la interfaz de usuario y otros recursos cruciales de la aplicación.

Con este contexto establecido, centraremos nuestra atención en cómo dicho archivo administra la navegación. En nuestra implementación, optamos por la biblioteca *@react-navigation*¹⁰ para encargarnos de esta función. El primer paso consiste en importar las diferentes pantallas al archivo en cuestión. Posteriormente, en el cuerpo del archivo, construimos un *NavigationContainer*. Dentro de este contenedor, especificamos cuál será la pantalla inicial al abrir la aplicación. Seguidamente, renderizamos cada una de las pantallas dentro del *NavigationContainer* conforme a la metodología que detallamos en el fragmento 4.2.

```
1 <Stack . Screen
2   name=" PantallaCrearComida "
3   component={ PantallaCrearComida }
4   options={{ headerShown: false }}
5 />
6 <Stack . Screen
7   name=" PantallaInicio1 "
8   component={ PantallaInicio1 }
9   options={{ headerShown: false }}
10 />
```

Listing 4.2: Renderizado de dos pantallas en App.js

¹⁰<https://reactnavigation.org/>

Como se ilustra en el fragmento de código 4.2, para cada pantalla configuramos un *Stack.Screen*. Este componente nos facilita el acceso a la pantalla correspondiente en el resto de la aplicación. Una vez establecido en el archivo *App.js*, al desear incorporar esta funcionalidad en una pantalla en particular, seguimos el procedimiento descrito a continuación. En la pantalla de inicio, es esencial importar el paquete *@react-navigation*, previamente mencionado. Posteriormente, inicializamos el navegador en el código de dicha pantalla, conforme a lo estipulado en el fragmento 4.3.

```
1 const navigation = useNavigation();
```

Listing 4.3: Inicialización del navegador

Con esta implementación, podemos enriquecer nuestras funciones con la habilidad de navegar entre distintas pantallas. Para ello, simplemente integramos la línea de código reflejada en el fragmento 4.4.

```
1 const handleAbrirPantallaCrearEntrenamiento= (editar) =>{
2   navigation.navigate("PantallaCreacionDeEntrenamientos", {editar: editar ,
3     planes: false});
4 }
```

Listing 4.4: Código para navegar entre pantallas

Los parámetros de la función *navigate* son esenciales para determinar el destino de dicha navegación. Adicionalmente, nos brindan la posibilidad de transmitir variables que pueden ser utilizadas en la pantalla a la que nos dirigimos. En el ejemplo 4.4, suministramos a la pantalla de creación de entrenamientos las variables *editar* y *planes*.

4.3.4. Listas

Habiendo abordado el mecanismo de navegación entre las diversas pantallas, es imperativo explicar cómo nuestra aplicación administra y procesa listas de datos para que el usuario no sólo las visualice, sino también interactúe con ellas. Con este propósito, hemos recurrido a *FlatList*, un componente nativo de React Native. Dicho componente tiene la capacidad de convertir una lista de datos, que en nuestro contexto proviene de Firebase, en un elemento visualmente coherente y amigable para el usuario.

Para presentar al usuario una lista de datos coherente, el primer paso es convertir los datos recibidos, en este caso desde Firebase, en una variable apta para su uso en nuestra pantalla. Los detalles específicos de la conexión con Firebase se abordarán posteriormente en este mismo apartado. Tras obtener la lista de datos, incorporamos en la pantalla el componente *FlatList*.

```
1 <FlatList
2   data={ejercicios}
3   keyExtractor={({item, index}) => index.toString()}
4   renderItem={({ item, index}) => (
5     <View style= {{marginBottom: 10,top:7}}>
6       <Pressable onPress={() => handleAbrirPantallaEditarEjercicios(
7         item)}>
8         <Text style= {[styles.subheading, styles.subheadingpo]}>{item.
9           nombre}</Text>
10        {index !== ejercicios.length && <FlatListItemSeparator />}
11      </Pressable>
12    </View>
13  )}
14  contentContainerStyle={{ position: 'absolute', zIndex: 1,paddingBottom:20,
15    bottom:0, top:15}}
```

Listing 4.5: Código del componente Flatlist

Como se ilustra en el fragmento 4.5, se presenta un ejemplo representativo de cómo estructuramos las listas en nuestra aplicación. En el parámetro *data*, designamos la variable donde hemos almacenado nuestra lista proveniente de la base de datos. Posteriormente, mediante el parámetro *renderItem*, dictamos a *Flatlist* el formato de visualización de los datos. En esta instancia, se busca desplegar la lista de ejercicios del usuario; para lograrlo, se genera una *View* por cada ejercicio y se inserta el nombre correspondiente. Para optimizar la legibilidad, se introduce una línea divisoria entre cada ejercicio. El despliegue final se refleja en la figura 4.10.

**Figura 4.10:** Despliegue de ejercicios mediante Flatlist

Destacamos la versatilidad de este componente, dada su capacidad de personalización. Esta característica nos ha motivado a incorporarlo en diversas secciones de nuestro proyecto, adaptándolo a las necesidades específicas de cada contexto.

4.3.5. Autenticación

Habiendo abordado aspectos esenciales de la funcionalidad de nuestra aplicación, es imperativo centrar nuestra atención en la que consideramos su función primordial: la conexión con Firebase y su integración en la aplicación. Este elemento representa el nexo entre la aplicación móvil y nuestra base de datos, siendo fundamental para alcanzar el objetivo propuesto.

En las etapas subsecuentes del desarrollo, nos enfrentamos a un desafío significativo al intentar integrar nuestra aplicación con Firebase. Si bien la instalación de los módulos se llevó a cabo sin contratiempos, la documentación a la que recurrimos para su configuración parecía estar desfasada. Esta situación nos condujo a una serie de inconvenientes, siendo el más notorio que únicamente lográbamos operatividad en el módulo de autenticación. Para resolver este problema, diseñamos un archivo de configuración específico

para Firebase, en el cual se almacenasen las claves de autenticación de nuestro proyecto. Este recurso nos brindó la posibilidad de establecer conexiones adecuadas con todos los módulos, utilizando el citado archivo como puente, garantizando así una operatividad sin altercados en esta área.

Iniciaremos mostrando cómo establecimos la conexión entre Firebase y nuestra aplicación. Tras la instalación de los módulos correspondientes de Firebase, procedimos a generar un archivo en la raíz del proyecto destinado a albergar las claves de configuración, tal como se evidencia en el fragmento 4.6.

```
1 import { initializeApp } from '@firebase/app';
2
3 const firebaseConfig = {
4
5   apiKey: '*****',
6   projectId: 'tfgivan-b5e4b',
7   appId: '*:*****',
8 };
9
10 const app = initializeApp(firebaseConfig);
11
12 export default app;
```

Listing 4.6: Archivo de configuración de firebase

Una vez hecho esto, ya podíamos empezar a usar los módulos que instalamos previamente. Para nuestra aplicación, solamente hemos utilizado dos módulos: el módulo de autenticación, que será empleado para registrar a los usuarios y les permitirá iniciar sesión, acceder a su sección de la base de datos, mantener su sesión activa, etc., y el módulo database, que, en conjunto con el de autenticación, nos permitirá conectarnos a la base de datos y usar sus métodos para extraer o modificar datos.

Empezaremos explicando el módulo de autenticación. En nuestra aplicación, emplearemos este módulo principalmente para iniciar sesión, registrarse y para recuperar las contraseñas de los usuarios.

Para iniciar sesión, utilizaremos el método “signInWithEmailAndPassword(email, password)”. Los parámetros email y password estarán relacionados con dos campos de texto que el usuario rellenará. Con la información de esos campos, la aplicación comprobará en la base de datos si el correo electrónico se encuentra registrado y si la contraseña coincide con la introducida por el usuario.

Para registrarse, emplearemos el método “createUserWithEmailAndPassword(email, password)”, aunque, primero, la aplicación realizará una pequeña comprobación para evitar errores. Verificará que la contraseña sea suficientemente segura, que el correo exista realmente, etc. Después de esta validación, si los datos son correctos, creará el usuario y lo añadirá a la base de datos. En este apartado también haremos uso del módulo database, ya que, una vez creado el usuario, se generará también una entrada en la base de datos en tiempo real.

Para recuperar la contraseña de un usuario, usaremos el método “sendPasswordResetEmail(email)”. En la pantalla donde se realice este proceso, el usuario escribirá el correo electrónico con el que creó su cuenta. El sistema comprobará que el correo exista en la base de datos y, en caso afirmativo, Firebase se encargará de enviar un correo con el que el usuario podrá establecer una nueva contraseña.

Cabe destacar que, a lo largo de la aplicación, también hemos empleado el método “currentUser” para obtener la referencia al usuario autenticado. De esta manera, y como explicaremos más adelante, conseguiremos la referencia en la base de datos del usuario.

4.3.6. Base de datos en tiempo real

A continuación, hablaremos de cómo hemos utilizado el módulo Database de Firebase para gestionar los datos que se muestran en la aplicación para cada usuario.

En algunas pantallas necesitábamos mostrar una lista de datos, ya sea una lista de ejercicios o de entrenamientos, por poner algún ejemplo. También necesitábamos que esa lista se actualizara cada vez que un usuario modificara o añadiera algún elemento en dicha lista. Para ello, hemos empleado los métodos que nos proporciona el módulo. En este caso, los métodos “on(‘value’, handleSnapshot)” y “off(‘value’, handleSnapshot)”. Estos métodos fueron utilizados dentro de la función “useEffect”, nativa de React Native. En este método, primero creamos una referencia, es decir, el lugar de la base de datos donde queremos que busque los datos. A continuación, definimos la función “handleSnapshot”, en la cual se procesarán los datos recibidos de la referencia de la base de datos y se transformarán en listas que podrán ser usadas por otras funciones o elementos de la interfaz. El método “off”, sin embargo, nos permitirá limpiar la suscripción que hemos creado a la base de datos con el otro método, para evitar que se llene la memoria del dispositivo.

```

1 useEffect(() => {
2   const ejerciciosRef = firebase.app().database(*****).ref('users/$
   {user.uid}/ejercicios ');
3   const handleSnapshot = (snapshot) => {
4     const ejerciciosData = snapshot.val();
5     if (ejerciciosData) {
6       const ejerciciosArray = [];
7       // Convertir los ejercicios en un array y actualizar el estado
8       Object.keys(ejerciciosData).forEach((key) => {
9         ejerciciosArray.push(ejerciciosData[key]);
10      });
11      setEjercicios(ejerciciosArray);
12    }
13  };
14  ejerciciosRef.on('value', handleSnapshot);
15  return () => {
16    ejerciciosRef.off('value', handleSnapshot);
17  };
18 }, []);

```

Listing 4.7: Procesamiento de la lista de ejercicios del usuario

En el fragmento de código 4.7, podemos observar un ejemplo de cómo hemos usado el módulo database para procesar la lista de ejercicios de un usuario. Por cuestiones de seguridad, hemos censurado la referencia a la base de datos en tiempo real de nuestra aplicación.

Otro uso que hemos hecho de este módulo ha sido para crear nuevas entradas en la base de datos, como por ejemplo un usuario que quiera agregar nuevos ejercicios o entrenamientos a la aplicación. Para ello, emplearemos el método “once(‘value’, (snapshot) =>”. A través de este método, con una referencia a la base de datos, nos permite leer, escribir nuevos datos o modificar los que ya existan. Con el método “push” crearemos una nueva entrada en el lugar al que estemos apuntando; una vez creada la nueva entrada, podremos modificarla utilizando el método “set”. En el fragmento de código 4.8, mostraremos un ejemplo de cómo creamos ejercicios en la aplicación usando estos métodos.

```

1 ejerciciosRef.once('value', (snapshot) => {
2   if (snapshot.exists()) {
3     const nuevoEjercicio = {
4       nombre: nombreEjercicio,
5       tipo: tipoEjercicio,
6     };

```



```
7     const nuevoEjercicioRef = ejerciciosRef.push();
8     nuevoEjercicioRef.set(nuevoEjercicio);
9   } else {
10    const nuevoEjercicio = {
11      nombre: nombreEjercicio,
12      tipo: tipoEjercicio,
13    };
14    const nuevoEjercicioRef = ejerciciosRef.push();
15    nuevoEjercicioRef.set(nuevoEjercicio);
16    console.log('Nueva lista de ejercicios creada:', [nuevoEjercicio]);
17  }
18  });
```

Listing 4.8: Creación de ejercicios

Después de varios meses de desarrollo, nos vimos en la necesidad de reconsiderar ciertas características previamente contempladas para nuestra aplicación. Entre estas, se encontraba la funcionalidad que permitiría a los usuarios personalizar su perfil con una imagen de su elección. La razón principal detrás de esta decisión radicaba en la necesidad de crear una base de datos adicional exclusivamente para albergar estas imágenes. Esta incorporación no solo hubiera incrementado significativamente el tiempo de desarrollo, sino que también hubiera requerido una inversión adicional en recursos. Al evaluar cuidadosamente las ventajas frente a las implicaciones, determinamos que, a pesar del valor añadido que esta personalización podría brindar al usuario, era más prudente no incorporar esta característica en la versión actual de nuestra aplicación.

4.3.7. Icono de la aplicación

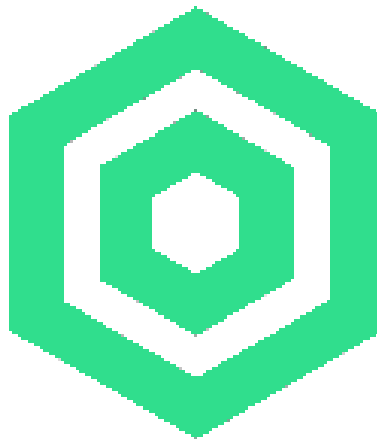


Figura 4.11: Logotipo de Fithive.

Más adelante, a la hora de generar la APK para nuestra aplicación, surgió un nuevo problema. Intentamos cambiar el icono por defecto de las aplicaciones Android por el que referenciamos en la figura 4.11. Para ello, tuvimos que generar una imagen para cada formato de icono que solicita una aplicación Android. Aún después de cambiar los iconos dentro de los archivos de nuestro proyecto, no conseguíamos que se actualizara la imagen. Para lograrlo, tuvimos que cambiar las referencias que se hacían en los archivos de configuración de Android para que escogieran nuestras imágenes sin dar ningún error.

CAPÍTULO 5

Resultados

En este capítulo mostraremos los resultados de nuestro trabajo, ofreceremos una visión detallada del resultado final de nuestro proyecto, mostraremos el diseño y funcionalidad de nuestra aplicación en su versión final y, además, realizaremos pruebas y test de fiabilidad para asegurarnos de la robustez de nuestro trabajo. Este apartado será la garantía de eficacia y eficiencia de nuestra aplicación.

5.1 Test de calidad

Las aplicaciones Android tienen una serie de requisitos mínimos de calidad que deben ser alcanzados. Para comprobar que nuestra aplicación cumple dichos requisitos, realizaremos los tests de calidad recomendados por los propios desarrolladores de Android en [8]. Haremos un listado con los tests, separándolos por las secciones indicadas en la página, y haremos un breve comentario sobre cómo nuestra aplicación pasa o no el test, si procede.

5.1.1. Experiencia Visual

Los tests de esta sección permitirán comprobar que la aplicación cumple con los estándares de interacción y diseño visual de Android para poder garantizar una experiencia coherente e intuitiva al usuario.

- **VX-N1:[✓]** Nuestra aplicación permite navegación estándar con el botón Atrás.
- **VX-N2:[✗]** Nuestra aplicación no permite la navegación con gestos para navegar a la pantalla principal.
- **VX-N3:[✓]** Nuestra aplicación preserva y restaura el estado del usuario o de la app, manteniendo el estado de ambos cuando abandona el primer plano.
- **VX-S1 y VX-S2:** Nuestra aplicación no contiene notificaciones ni es una aplicación de mensajería por ende estos test no serán relevantes.
- **VX-U1:[✓]** Nuestra aplicación no permite la orientación horizontal.
- **VX-U2:[⚠]** Nuestra aplicación utiliza prácticamente toda la pantalla y no usa el formato letterbox.
- **VX-U3:[✓]** Al no tener nada más que una orientación, solamente indicaremos que en esta no presenta problemas de renderizado ni de pérdidas de estados en el plegado o desplegado del dispositivo.

- **VX-V1:[✓]** Nuestra aplicación presenta gráficos, texto y otros elementos de la IU sin pixelados ni distorsión notables.
- **VX-V2:[✓]** Nuestra aplicación muestra textos y bloques de texto de forma visible para el idioma en que fue diseñada, no existen letras ni palabras cortadas y hay espacio suficiente entre el texto y los elementos que componen su alrededor.
- **VX-V3:[✗]** Nuestra aplicación no admite tema oscuro.
- **VX-A1:[✓]** Los elementos de los objetivos táctiles de nuestra aplicación cumplen con el estándar de tamaño establecido por este test.
- **VX-A2:[✓]** El texto y el contenido de primer plano de nuestra aplicación mantiene la relación de contraste de color establecida por los desarrolladores.
- **VX-A3:[✗]** Los elementos de la IU no están descritos mediante “contentDescription”.

5.1.2. Funcionalidad

Estos test servirán para comprobar que la aplicación reproduce audio y multimedia adecuadamente, también para comprobar que se comparta contenido de manera adecuada, nuestra aplicación no contiene ninguna de estas funciones así que salvo el test “FN-B1” el resto no tendrá ninguna relevancia.

- **FN-B1:[✓]** La aplicación no ejecuta servicios en segundo plano garantizando que el dispositivo del usuario funcione sin complicaciones.

5.1.3. Rendimiento y estabilidad

Estos test servirán para comprobar que nuestra aplicación tiene el rendimiento, la estabilidad y la capacidad de respuesta que pueden esperar los usuarios.

- **PS-S1:[✓]** La aplicación no bloquea el subproceso de IU que provoca el error de “Android no responde”.
- **PS-P1:[✓]** La aplicación carga rápidamente.
- **PS-P2:[✓]** La aplicación renderizara a 60 fotogramas por segundo.
- **PS-T1:[✓]** La aplicación se puede ejecutar en la ultima versión de la plataforma de Android sin que haya un impacto en el rendimiento o en las funcionalidades de la misma.
- **PS-T2:[✓]** La aplicación esta orientada para el SDK más reciente.
- **PS-T3:[⚠]** La aplicación sera compilada para la versión 48 o superiores del SDK de android.
- **PS-T4:[✓]** Todas las dependencias de la aplicación están actualizadas a su última versión.
- **PS-T5:[✓]** Nuestra aplicación no emplea ninguna interfaz que no pertenezca al SDK.
- **PS-T6:[✓]** En nuestra aplicación no se incluye ninguna biblioteca de depuración.
- **PS-B1:[✓]** La aplicación permite adecuadamente ejecutar las funciones de administración de energía introducidas en Android 6.0.

5.1.4. Privacidad y seguridad

Los test de esta sección permitirán comprobar si nuestra aplicación administra de forma segura los datos e información personal del usuario.

- **SC-P1:[✓]** La aplicación solicita la cantidad mínima de permisos para llevar a cabo sus funciones.
- **SC-P2:** Nuestra aplicación no necesita acceder a datos sensibles, por lo tanto este test es irrelevante para nuestra situación.
- **SC-P3:[✓]** Nuestra aplicación solamente solicita como permiso acceder a Internet, que lo hará una vez el usuario se descargue la aplicación.
- **SC-P4 y SC-P5:** Debido a solo requerir el permiso de acceder a Internet estos tests no son relevantes en nuestra solución.
- **SC-DF1:[✗]** Los datos sensible serán almacenados en la base de datos de la aplicación.
- **SC-DF2:[✓]** No se guardara ningún tipo de datos personales o sensibles de los usuarios en ningún tipo de registro.
- **SC-DF3:[✓]** La aplicación no incluye ninguna opción para autocompletar las credenciales de la cuenta de los usuarios.
- **SC-AC1 a SC-AC3:** No existen componentes que se exporten a otras aplicaciones por lo tanto estos test no son relevantes para nuestro caso.
- **SC-N1:[✓]** El tráfico de red será gestionado de manera segura a través de Firebase.
- **SC-N2:[✓]** La aplicación incluye un archivo de seguridad de red, con las claves de certificado necesarias.
- **SC-N3:** Nuestra aplicación no usa los servicios de Google Play por lo tanto este test no es relevante.
- **SC-W1 y SC-W2:** Nuestra aplicación al estar desarrollada con react native no usa los metodos `''setAllowUniversalAccessFromFileURLs()''`, `''WebViewAssetLoader''` ni `''addJavaScriptInterface()''` por lo tanto estos test no son relevantes para nuestro caso particular.
- **SC-E1:[✓]** La aplicación no cargará código dinámicamente desde fuera de la APK.
- **SC-C1:[✓]** La aplicación encriptará las credenciales de los usuarios empleando el encriptado que proporciona la autenticación con Firebase.

Tras realizar estos tests, podemos comprobar que nuestra aplicación cumple con la gran mayoría de los mismos. Por ende, podemos afirmar con seguridad que nuestro trabajo cumple con los estándares de calidad establecidos por los desarrolladores de Android.

5.2 Pruebas de usabilidad

Para probar la aplicación en un entorno real, decidimos enviar diferentes versiones de la aplicación a cuatro personas para que probaran las diferentes funcionalidades de

nuestra solución. De esta forma, podríamos detectar errores que nuestro equipo pudiera haber pasado por alto.

El primer error que recibimos de nuestros “testers” fue en la calculadora del porcentaje de grasa. Con la versión que teníamos en ese momento, salía un cálculo erróneo que provocaba que el porcentaje saliera más bajo de lo normal. Una persona que tuviera un porcentaje sano, como puede ser 19%, aparecía en nuestra aplicación con un porcentaje del 4%, algo para nada saludable. Para solucionar esto, cambiamos nuestra fuente de la cual sacamos la fórmula con la que calculamos este porcentaje. Añadimos campos nuevos para añadir las medidas restantes, tal y como mostramos en la figura 5.22, y cambiamos el código para ajustarnos a la nueva fórmula. Nos basaremos en la fórmula que emplea la marina de Estados Unidos, tal y como se muestra en esta calculadora en línea [9]. Esta fórmula proporciona unas medidas más certeras del índice de grasa que las que empleábamos anteriormente.

Otro error que recibimos fue que en algunos campos de texto no había control sobre qué tipo de caracteres podían introducirse. Esto tuvo fácil solución: añadiendo un parámetro en los campos de texto correspondientes podíamos tanto limitar el número de caracteres que se introducían como limitar el tipo de carácter que se escribía, por ejemplo, una coma o un punto delante de un campo numérico o una letra en un campo exclusivamente numérico.

También recibimos varios reportes de algunas páginas en las que el tamaño de la fuente era demasiado pequeño o no se ajustaba bien a su contenedor. Para sortear este problema, fuimos probando diferentes fuentes y tamaños que se ajustaran mejor que los que teníamos en ese momento. Además, fuimos retocando las pantallas en las que algunos elementos se habían descuadrado al añadirles alguna funcionalidad.

Debido a la falta de tiempo y de recursos, no hemos podido realizar pruebas más exhaustivas con otros usuarios, pero creemos firmemente que nuestra aplicación cumple con todas las funciones que nos propusimos implementar.

5.3 Interfaz de la versión final

En la siguiente sección, nos adentraremos en una narración entre dos individuos, permitiéndonos explorar y evidenciar el arte final de la aplicación en un contexto realista. A través de esta historia, no sólo entenderemos las funcionalidades de nuestra aplicación, si no que también comprenderemos su aplicabilidad y relevancia en escenarios cotidianos.

Marta quiere empezar a hacer ejercicio y estuvo buscando alguna manera de monitorear su progreso. David le comentó que él usaba una aplicación llamada “Fithive” y le propuso enseñarle a usarla.

Una vez la descargó, el primer paso fue registrarse en la aplicación. Marta introdujo sus datos y así creó su cuenta; tras esto, inició sesión. En las figuras 5.1 y 5.2 podemos observar el proceso en la aplicación.



E-mail address
martavlc733@gmail.com

marta1999

.....

.....

REGISTRARSE

¿YA TIENES CUENTA?



E-mail address
martavlc733@gmail.com

.....

OLVIDÉ LA CONTRASEÑA

INICIAR SESIÓN

REGISTRARSE

Figura 5.1: Creación del usuario de Marta

Figura 5.2: Inicio de sesión con los datos

Una vez iniciada la sesión, Marta quiere crear un plan de entrenamiento para empezar a monitorear su progreso. Para ello, David le indica que a través del submenú navegue a la sección de gestión de entrenamientos de la aplicación. Tras esto, accederá a la sección de planes de entrenamiento y entrará al creador de entrenamientos. Las pantallas involucradas en este proceso estarán representadas en las figuras 5.3, 5.4, 5.5 y 5.6.

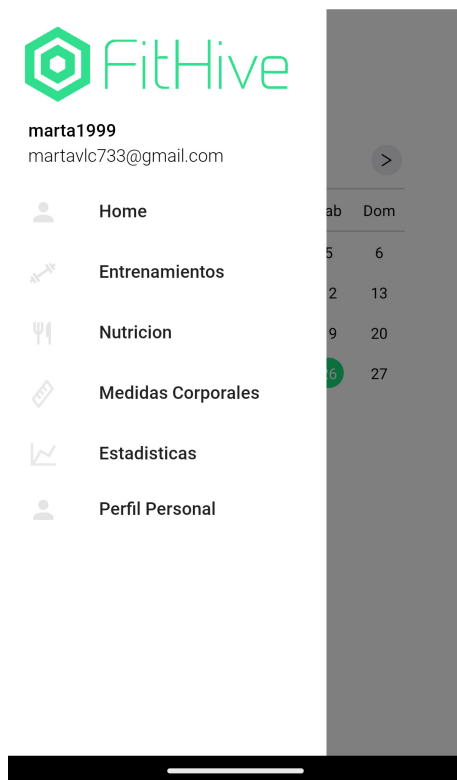


Figura 5.3: Submenú de la aplicación

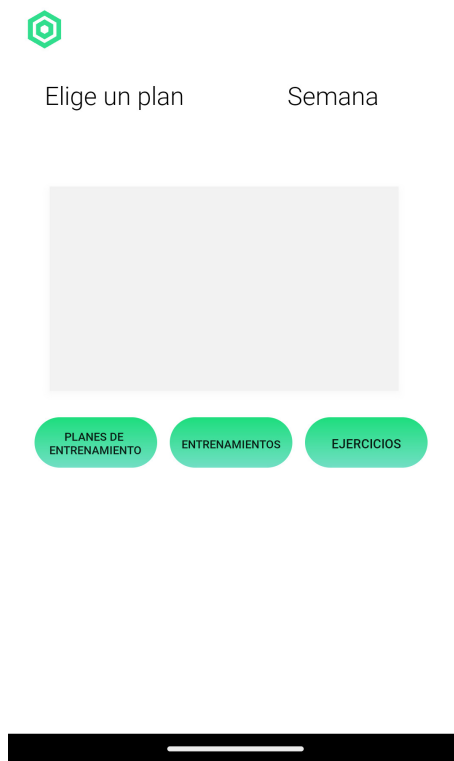


Figura 5.4: Inicio de sesión con los datos

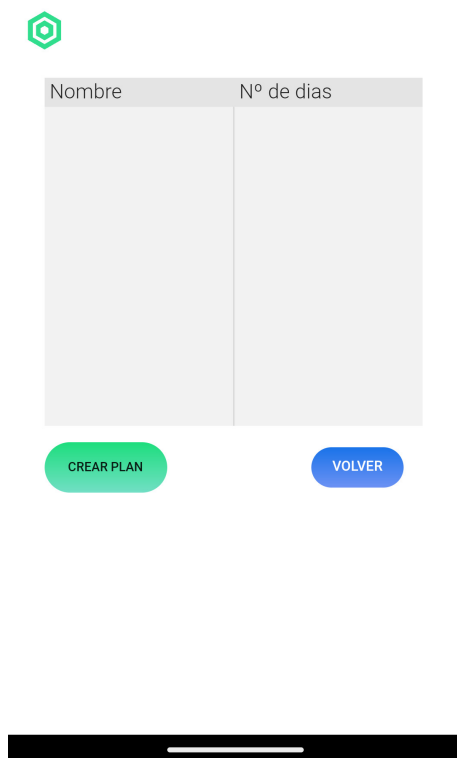


Figura 5.5: Biblioteca de planes de entrenamiento

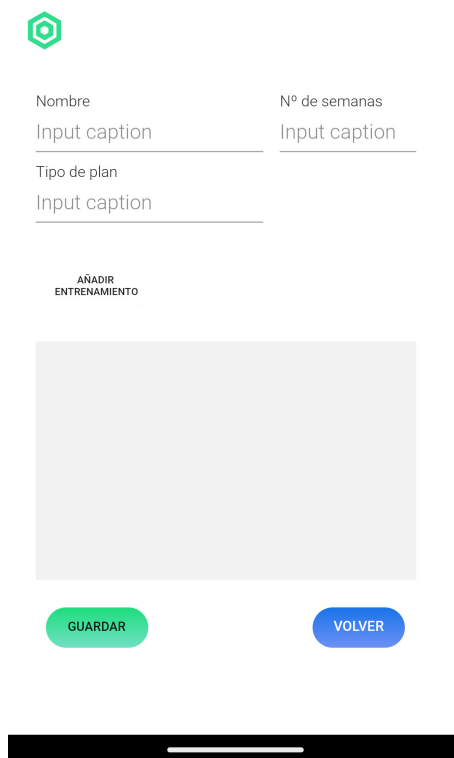


Figura 5.6: Creador de planes

Una vez aquí, Marta rellenará los campos del creador de planes y, como no tiene ningún entrenamiento almacenado en la aplicación, tendrá que crearlos. David le explica el proceso para crear entrenamientos, que es bastante parecido al de crear planes. Primero tendrá que navegar a la pantalla de crear entrenamientos; eso lo hará pulsando el botón de añadir entrenamiento y pulsando después en el botón de crear un nuevo entrenamiento. Este flujo estará representado en las figuras 5.7 y 5.8.

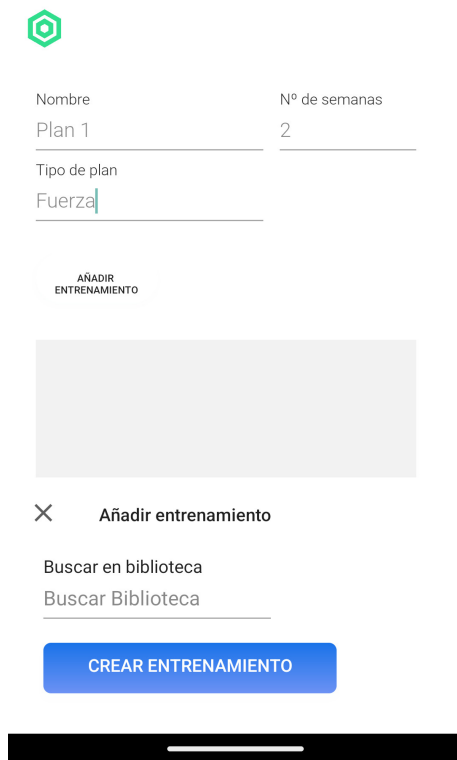


Figura 5.7: Pestaña para elegir entrenamiento a agregar

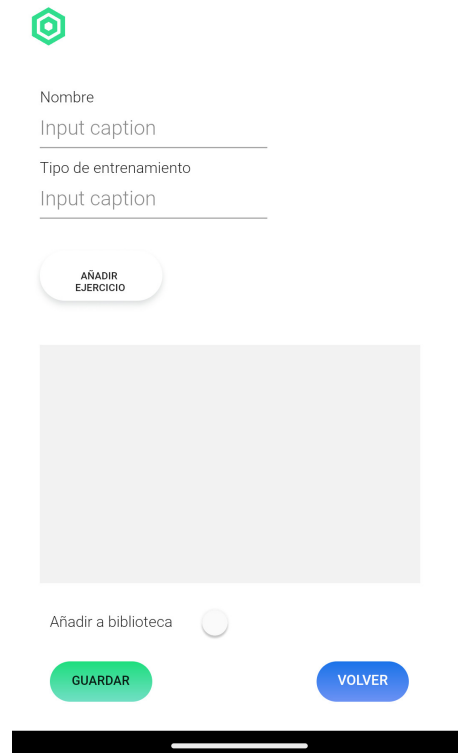


Figura 5.8: Creador de entrenamientos

En esta pantalla, Marta rellenará los campos con los datos de su entrenamiento. Para agregar los ejercicios, David le enseñará el flujo que tiene que seguir para conseguirlo. Al igual que en el punto anterior, Marta no tiene ningún ejercicio creado; por ello, usando la función incluida en esta pantalla, rellenará los datos con los ejercicios que incluya su entrenamiento. Las figuras 5.9 y 5.10 nos indicarán cómo Marta ha creado y seleccionado los ejercicios para su entrenamiento. La figura 5.11 nos mostrará cómo quedó el entrenamiento ya completado. En la figura 5.12 se mostrará cómo quedó el plan después de que Marta añadiese los entrenamientos que quería incluir en el mismo.

Nombre
Full body

Tipo de entrenamiento
fuerza

AÑADIR EJERCICIO

X Añadir ejercicio

DE BIBLIOTECA

CREAR EJERCICIO

Figura 5.9: Pestaña para elegir ejercicio a agregar

Nombre
Full body

Tipo de entrenamiento
fuerza

AÑADIR EJERCICIO

Sentadilla

X Crear Ejercicio

Nombre
Press banca

Nº de series
3

Tipo de ejercicio
fuerza

Nº de rep
6

Añadir a biblioteca

GUARDAR

Figura 5.10: Creador de ejercicios

Nombre
Full body

Tipo de entrenamiento
fuerza

AÑADIR EJERCICIO

Sentadilla
Press banca
Zancadas
Curl de biceps
Curl de triceps
Rueda abdominal

Añadir a biblioteca

GUARDAR

VOLVER

Figura 5.11: Entrenamiento a agregar

Nombre
Plan 1

Nº de semanas
2

Tipo de plan
Fuerza

AÑADIR ENTRENAMIENTO

Full body
Full body 2

GUARDAR

VOLVER

Figura 5.12: Plan con todos los datos añadidos

Una vez que Marta agregó el plan a su biblioteca, David le explicó que para poder llevar un seguimiento del progreso tenía que activar el plan. Esto lo haría desde la biblioteca de planes, tal y como se indica en la figura 5.13. De esta manera, en la pestaña de inicio de entrenamientos, podría seleccionar el entrenamiento que quiere realizar, como se muestra en la figura 5.14. Cuando vaya al gimnasio a hacer el entrenamiento, solamente tendrá que pulsar sobre el entrenamiento que le toque y completar los campos de la nueva pantalla. Una vez guarde el entrenamiento, quedará registrado en la entrada del calendario correspondiente al día en que lo guarde.

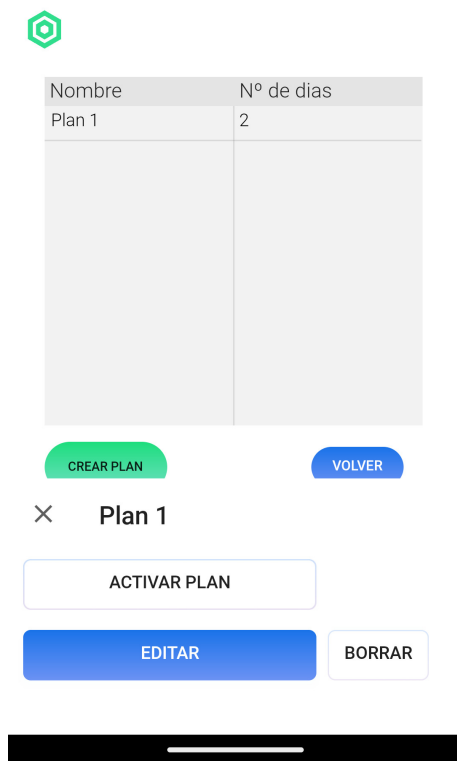


Figura 5.13: Activación del plan

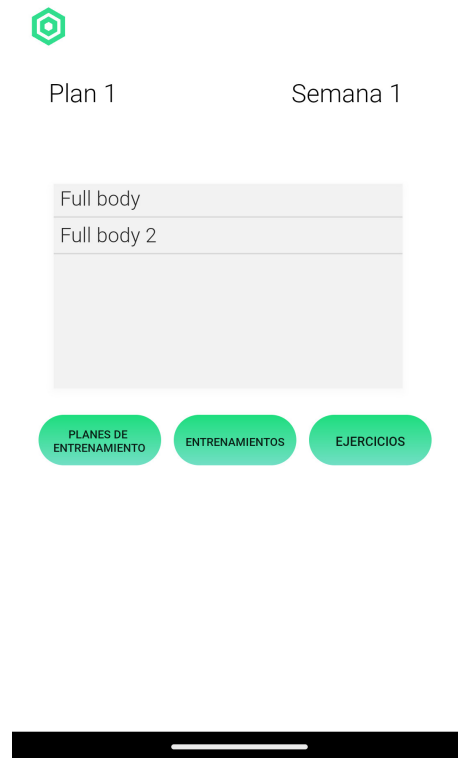


Figura 5.14: Pantalla de entrenamientos con el plan activado

Tras aprender a gestionar sus entrenamientos, Marta le preguntó a David cómo podría usar la aplicación para controlar las calorías que consume y así conseguir sus objetivos de peso. David le explicó que la misma aplicación incluye una sección para llevar un seguimiento de las comidas diarias. Lo primero que le indicó fue cómo establecer un objetivo calórico. Marta quería subir de peso corporal, así que, usando el apartado correspondiente de la aplicación y rellenando los datos, estableció el objetivo que tendría que seguir las próximas semanas. En la figura 5.15 podremos ver cómo se llevaría a cabo esta acción en la aplicación.



Figura 5.15: Selector de objetivo calórico

Una vez establecido el objetivo, David enseñó a Marta a añadir la comida que acababan de preparar al diario, para que empezara a contar para sus calorías diarias. Como aún no había agregado ninguna comida a la biblioteca, tuvo que crear una desde el principio usando el creador de comidas. En la figura 5.16 veremos el selector con el cual nos dirigiremos al creador de comidas, referenciado en la figura 5.17.



Figura 5.16: Selector de comida a añadir

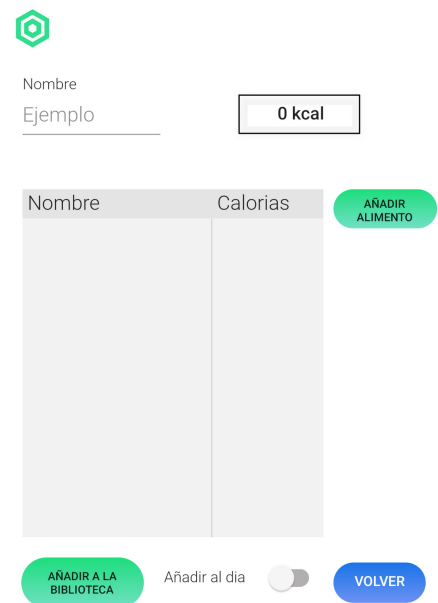


Figura 5.17: Creador de comidas

En esta pantalla, Marta rellenó la lista de ingredientes como se muestra en la figura 5.18. Una vez tuvo todos los datos completados, quedando como se ilustra en la figura 5.19, agregó la comida al día. Una vez hecho esto, la comida estaría añadida a la lista de comidas diarias de la página principal de nutrición. Además, un gráfico mostrará a Marta las calorías que le faltan para cumplir el objetivo diario tal y como se muestra en la figura 5.20.

Nombre
tortilla de patata

0 kcal

Nombre	Calorias

AÑADIR ALIMENTO

✕ Añadir alimento

Nombre Calorias
huevo 300

GUARDAR

Figura 5.18: Creador de alimentos

Nombre
tortilla de patata

600 kcal

Nombre	Calorias
huevo	300
Patatas	300

AÑADIR ALIMENTO

AÑADIR A LA BIBLIOTECA Añadir al día VOLVER

Figura 5.19: Comida a añadir.



Figura 5.20: Pantalla de nutrición actualizada

Por último, Marta le pidió a David que le explicase cómo podía añadir sus medidas corporales a la aplicación. Éste le dijo que la aplicación contaba con un apartado específico para ello, cuya pantalla será la que se muestra en la figura 5.21. Aquí podría crear una nueva entrada para el día de hoy, tal y como se ilustra en la figura 5.22. En ella, Marta rellenó todos los campos con sus datos personales y así guardó la entrada en la base de datos de la aplicación.

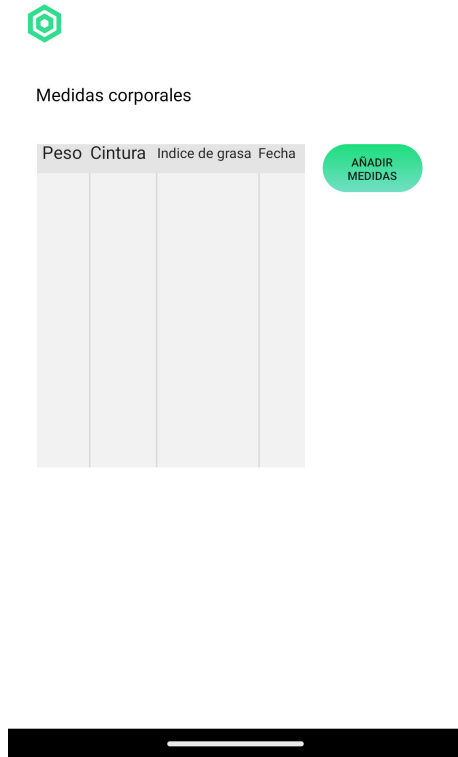


Figura 5.21: Biblioteca de medidas corporales



Figura 5.22: Creador de medidas corporales

CAPÍTULO 6

Conclusiones

En este apartado final, analizaremos si los objetivos que nos planteamos al principio del trabajo en el apartado 1.2 han sido completados. También relacionaremos el trabajo realizado con los estudios que hemos cursado durante la carrera. Por último describiremos los planes para el futuro de esta aplicación.

Al comienzo del proyecto surgieron numerosos problemas a la hora de desarrollarlo debido a la carencia de conocimientos sobre las tecnologías empleadas que disponíamos. Conforme avanzo el desarrollo, debido a la práctica y al estudio de las mismas, conseguimos familiarizarnos con ellas y así superamos las dificultades que fueron surgiendo.

El objetivo principal de este trabajo era la creación de una aplicación que nos permitiera organizar nuestra vida deportiva, ya fuera gestionando nuestros entrenamientos o llevando un seguimiento de nuestra evolución. Las funciones principales de la aplicación que teníamos pensadas, como por ejemplo crear un plan de entrenamiento, diseñar ejercicios o establecer comidas, las hemos conseguido desarrollar en su totalidad. La solución cuenta con la capacidad de crear y almacenar tanto planes como entrenamientos y ejercicios en el ámbito deportivo y la capacidad de diseñar y almacenar comidas en el ámbito nutricional.

Por otro lado, otro de los objetivos que nos propusimos fue que la aplicación contara con un calendario en el que pudiéramos registrar los entrenamientos que realizamos cada día o las comidas que consumimos. Finalmente, conseguimos añadir esta funcionalidad a nuestra aplicación.

Por último, queríamos que nuestra aplicación contara con una forma de llevar un seguimiento de nuestro peso y medidas corporales, algo vital para los deportistas que quieren identificar la razón por la cual no alcanzan sus objetivos. Para esto, logramos incorporar esta función en nuestra aplicación, incluyendo hasta una fórmula para calcular el índice de grasa corporal del usuario. También añadimos una sección en la que observar el progreso que hemos conseguido a lo largo de los meses, mostrando una gráfica que nos permitirá verlo claramente.

Tras este breve resumen, podemos concluir que sí, todos los objetivos que nos propusimos fueron cumplidos. A nivel personal, este proyecto me ha ayudado a entender mejor cómo gestionar la vida deportiva, proporcionándome nuevos conocimientos que me serán de gran utilidad en el futuro. También he adquirido habilidades en una amplia gama de tecnologías punteras que serán fundamentales en mi carrera profesional. A nivel profesional, este proyecto me ha brindado una visión de cómo es el desarrollo de una aplicación real, permitiéndome aprender cómo se gestionan los problemas que surgen durante el desarrollo y cómo superarlos.

6.1 Relación con los estudios cursados

A lo largo de estos años hemos cursado una gran cantidad de asignaturas, todas ellas destinadas a convertirnos en los profesionales del sector que somos hoy en día. Gracias a los conocimientos que hemos adquirido a través de estas asignaturas, hemos podido afrontar este trabajo y, aunque todas han aportado su granito de arena, hay varias que me gustaría destacar en esta sección.

- **Interfaces persona computador:** Los conocimientos adquiridos en esta asignatura respecto al desarrollo de interfaces y los estándares de calidad que necesitan cumplir han resultado invaluable en el desarrollo de este trabajo.
- **Base de datos y sistemas de información:** En esta asignatura aprendimos a crear y a administrar bases de datos, algo que ha resultado de gran utilidad para nuestro trabajo.
- **Gestión de proyectos:** Gracias a las enseñanzas que aprendimos en esta asignatura hemos conseguido organizar adecuadamente el proyecto a lo largo del tiempo para tener un desarrollo fluido y con las menores complicaciones posibles.
- **Desarrollo web:** En esta asignatura aprendimos algunos de los lenguajes más comunes en el desarrollo web como HTML y CSS, también aprendimos cómo pueden interactuar cliente y servidor en una aplicación web. Estos conocimientos han sido requeridos a la hora de desarrollar nuestra aplicación.
- **Ciberseguridad en dispositivos móviles:** Gracias a esta asignatura aprendimos cómo funcionan y cómo están estructuradas las aplicaciones Android, esto ha permitido desarrollar una aplicación más completa y segura para el usuario.

No me cabe duda de que aún podríamos nombrar numerosas asignaturas más, pero a nuestro parecer, estas serían las que más nos han ayudado en el desarrollo de este proyecto.

6.2 Trabajo Futuro

Aunque nuestra aplicación cumple con los requisitos que nos propusimos al comenzar este trabajo, esta no será ni mucho menos su versión final. Nuestro objetivo final es convertir esta aplicación en una más completa, que no se limite solo a un tipo de deporte; por eso mismo, el equipo tiene numerosas ideas sobre cómo ampliar y mejorar la misma.

- **Publicación en Google Play:** El primer objetivo que tenemos planteado a corto plazo es la publicación de la aplicación en Google Play; de esta forma, podremos llegar a más gente a la que ayudar a gestionar su vida deportiva.
- **Crear versión multiplataforma:** En su primera versión, la aplicación solo estará disponible para Android, pero debido a la facilidad con la que React Native permite el desarrollo multiplataforma, el equipo se compromete a crear una versión que sea soportada en más sistemas operativos.
- **Añadir más funciones de accesibilidad:** Otra función que nos gustaría agregar a la aplicación es la posibilidad de mayor personalización de la interfaz; funciones como cambiar el tamaño de la letra para las personas con problemas visuales o incluso algún modo para daltónicos que se ajuste a sus necesidades es algo que nos encantaría abordar en un futuro próximo.

- **Incluir más deportes en el apartado deportivo:** Actualmente, la aplicación solo cuenta con la creación de ejercicios de fuerza o estiramientos; a medio plazo, añadiremos soporte para otros deportes, como el atletismo o la natación, por poner algún ejemplo.

Como bien podemos observar, el futuro del desarrollo de esta aplicación está asegurado. Todas estas mejoras servirán de guía para el equipo, que junto con las opiniones de los usuarios harán que la aplicación siga progresando a lo largo del tiempo.

Bibliografía

- [1] *El deporte tras el confinamiento, ¿un negocio al alza?* Fecha de acceso: 03 AGO 2020. 2020. URL: <https://elpais.com/economia/estar-donde-estes/2020-08-03/el-deporte-tras-el-confinamiento-un-negocio-al-alza.html> (vid. pág. 1).
- [2] *¿Qué es la metodología ágil?* Fecha de acceso: 19 JUL 2022. 2022. URL: <https://www.redhat.com/es/devops/what-is-agile-methodology> (vid. pág. 2).
- [3] Chiyana Simões. *MoSCoW. ¿Qué es y cómo priorizar en el desarrollo de tu aplicación?* Fecha de acceso: 14 JUL 2020. 2020. URL: <https://www.itdo.com/blog/moscow-que-es-y-como-priorizar-en-el-desarrollo-de-tu-aplicacion/> (vid. pág. 2).
- [4] IBM. *Arquitectura de tres niveles*. URL: <https://www.ibm.com/es-es/topics/three-tier-architecture#:~:text=La%20arquitectura%20de%20tres%20niveles%20es%20una%20arquitectura%20de%20aplicaci%C3%B3n,datos%2C%20donde%20se%20almacenan%20y> (vid. págs. 2, 13).
- [5] Pandora FMS team. *NoSQL*. 2022. URL: <https://pandorafms.com/blog/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una/#:~:text=Ventajas%20de%20una%20base%20de,los%20modelos%20de%20Entidad%20Relaci%C3%B3n>. (vid. pág. 27).
- [6] *Android Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides)*. 2017 (vid. pág. 35).
- [7] *Fullstack React Native, The Complete Guide to React Native*. 2017 (vid. pág. 36).
- [8] Google. *Calidad básica de las app*. URL: <https://developer.android.com/docs/quality-guidelines/core-app-quality?hl=es-419> (vid. pág. 47).
- [9] *Calcular Índice de grasa*. 2023. URL: <https://www.calculator.net/body-fat-calculator.html> (vid. pág. 50).

APÉNDICE A

Objetivos de desarrollo sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Hace 8 años, el 25 de diciembre de 2015, se adoptaron un conjunto de objetivos y pautas para proteger el planeta, asegurar un futuro próspero para todos, eliminar la pobreza, etc.. Cada objetivo cuenta con metas específicas que esperan poder lograrse en los próximos 15 años.

Tabla A.1: Objetivos de desarrollo sostenible

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.				✓
ODS 2. Hambre cero.				✓
ODS 3. Salud y bienestar.	✓			
ODS 4. Educación de calidad.				✓
ODS 5. Igualdad de género.				✓
ODS 6. Agua limpia y saneamiento.				✓
ODS 7. Energía asequible y no contaminante.				✓
ODS 8. Trabajo decente y crecimiento económico.				✓
ODS 9. Industria, innovación e infraestructura.				✓
ODS 10. Reducción de las desigualdades.		✓		
ODS 11. Ciudades y comunidades sostenibles.				✓
ODS 12. Producción y consumo responsable.		✓		
ODS 13. Acción por el clima.				✓
ODS 14. Vida submarina.				✓
ODS 15. Vida de ecosistemas terrestres.				✓
ODS 16. Paz, justicia e instituciones sólidas.				✓
ODS 17. Alianzas para lograr objetivos.				✓

En la tabla A.1, referenciamos todos los objetivos que son relevantes para nuestra solución. A continuación, explicaremos cómo están relacionados con ella.

- **ODS 3. Salud y bienestar:** Este objetivo es el que más se adecúa a nuestra aplicación, al tratarse de una solución para organizar la vida deportiva y nutricional de

las personas. Al permitir llevar un registro detallado de los hábitos alimenticios y el progreso en entrenamientos, nuestra aplicación contribuye directamente a este objetivo.

- **ODS 10. Reducción de las desigualdades:** Desde un primer momento, pensamos que nuestra aplicación debería ser capaz de ayudar a todo tipo de personas, independientemente de su origen socioeconómico, género o nivel de habilidad física. Es por esto que diseñamos una interfaz amigable y ofrecemos un gran número de herramientas para llevar un estilo de vida más saludable. Por todo esto, aspiramos a contribuir a nivelar las oportunidades y reducir la desigualdad en el ámbito del deporte y la salud.
- **ODS 12. Producción y consumo responsable:** Creemos que, gracias a que nuestra aplicación cuenta con la capacidad de guardar recetas y llevar un registro de lo que los usuarios consumen, los usuarios serán capaces de controlar mejor lo que consumen, siendo más responsables a la hora de comprar y organizar sus comidas. Esto hace que nuestra solución contribuya en cierta medida con este objetivo.

APÉNDICE B

Casos de uso

Usuario no registrado

Tabla B.1: CU-01, Registro en la aplicación

Nombre	Registrar usuario.
Objetivo	Crear una nueva cuenta de usuario empleando un correo electrónico.
Descripción	El usuario creará una nueva cuenta en la aplicación proporcionando un correo electrónico, un nombre de usuario y una contraseña.
Pre-condición	El usuario no debe de estar ya registrado en la aplicación.
Secuencia	<ol style="list-style-type: none">1. Presionar el botón Registrarse.2. Introducir los datos y pulsar el botón registrarse.3. La aplicación valida los datos y da de alta al usuario en la aplicación.4. Se redirige al usuario a la página de inicio de sesión para que pueda iniciar sesión.
Post-condición	La aplicación registra al usuario.
Excepciones	Los datos son erróneos, la aplicación alerta al usuario para que los revise.

Tabla B.2: CU-02, Inicio de sesión

Nombre	Iniciar sesión.
Objetivo	Iniciar la sesión para poder usar la aplicación.
Descripción	El usuario iniciará sesión en la aplicación proporcionando sus datos.
Pre-condición	El usuario no debe de tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Introducir los datos en los campos proporcionados y pulsar botón iniciar sesión. 3. La aplicación valida los datos y redirige al usuario a la página principal.
Post-condición	La aplicación inicia la sesión del usuario.
Excepciones	Los datos son erróneos, la aplicación alerta al usuario para que los revise.

Tabla B.3: CU-03, Olvidar contraseña

Nombre	Recuperar contraseña.
Objetivo	Recuperar la contraseña del usuario.
Descripción	El usuario cambiará su contraseña de inicio de sesión.
Pre-condición	El usuario no debe de tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. Iniciar la aplicación. 2. Pulsar botón olvidé la contraseña. 3. Introducir el correo electrónico en el campo proporcionado y pulsar el botón de enviar correo. 4. La aplicación valida la dirección de correo electrónico y manda el correo. 5. La aplicación redirige al usuario a la página de inicio de sesión.
Post-condición	La aplicación registrará el cambio de contraseña.
Excepciones	El correo no coincide con ningún usuario registrado.

Usuario Registrado

Tabla B.4: CU-04, Cerrar Sesión

Nombre	Cerrar Sesión.
Objetivo	Cerrar la sesión del usuario.
Descripción	El usuario cerrara la sesión actualmente abierta en la aplicación
Pre-condición	El usuario debe tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. En la página de inicio abrir el submenú de dirección y pulsar el botón perfil personal. 2. La aplicación redirigirá a la página donde el usuario pulsara sobre el botón cerrar sesión. 3. La aplicación cerrara la sesión del usuario y le redirigirá a la pagina de iniciar sesión.
Post-condición	La aplicación borrará la comida de la base de datos.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.5: CU-05,Añadir medida corporal

Nombre	Añadir medida corporal.
Objetivo	Añadir medidas corporales a la base de datos.
Descripción	El usuario añadirá una entrada de medidas corporales a la base de datos.
Pre-condición	El usuario debe tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. Abrir el menú desplegable de la aplicación y pulsar sobre el apartado medidas corporales. 2. Pulsar el botón añadir medidas. 3. Introducir los datos en los campos proporcionados y pulsar el botón de guardar.
Post-condición	La aplicación archivará las nuevas medidas corporales.
Excepciones	Los datos introducidos no son válidos, la aplicación alertará al usuario.

Tabla B.6: CU-06, Modificar medida corporal

Nombre	Modificar medida corporal.
Objetivo	Modificar medidas corporales de la base de datos.
Descripción	El usuario modificará una entrada de medidas corporales de la base de datos.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe de tener alguna medida corporal en la base de datos.
Secuencia	<ol style="list-style-type: none"> 1. Abrir el menú desplegable de la aplicación y pulsar sobre el apartado medidas corporales. 2. Pulsar sobre la medida corporal a editar. 3. Editar las medidas tal y como el usuario desee y pulsar el botón de guardar. 4. La aplicación comprobará los datos y actualizará la medida en la base de datos. El submenú de edición se cerrará una vez se guarde.
Post-condición	La aplicación actualizará la entrada en la base de datos.
Excepciones	Los datos introducidos no son válidos, la aplicación alertará al usuario.

Tabla B.7: CU-07,Borrar medida corporal

Nombre	Borrar medida corporal
Objetivo	Borrar medidas corporales de la base de datos.
Descripción	El usuario borrará una entrada de medidas corporales de la base de datos.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe de tener alguna medida corporal en la base de datos.
Secuencia	<ol style="list-style-type: none"> 1. Abrir el menú desplegable de la aplicación y pulsar sobre el apartado medidas corporales. 2. Pulsar sobre la medida corporal a borrar. 3. Pulsar el botón de borrar. 4. La aplicación comprobará los datos y borrará la medida en la base de datos. El submenú de edición se cerrará una vez se guarde.
Post-condición	La aplicación actualizará la entrada en la base de datos.
Excepciones	

Tabla B.8: CU-08,Visualizar estadísticas

Nombre	Visualizar estadísticas
Objetivo	Ver estadísticas del usuario.
Descripción	El usuario seleccionará qué datos desea ver en los gráficos.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe de tener algún entrenamiento o medida registrado en la base de datos.
Secuencia	<ol style="list-style-type: none"> 1. Abrir el menú desplegable de la aplicación y pulsar sobre el apartado estadísticas. 2. Seleccionar los datos que desea ver en el gráfico.
Post-condición	La aplicación mostrará un gráfico con los datos proporcionados.
Excepciones	

Tabla B.9: CU-09, Modificar datos de la cuenta

Nombre	Modificar datos de la cuenta
Objetivo	Modificar los datos del usuario.
Descripción	El usuario modificará los datos de su cuenta de la aplicación.
Pre-condición	El usuario debe tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. Abrir el menú desplegable de la aplicación y pulsar sobre el apartado de perfil personal. 2. Elegir el apartado que desea cambiar de su cuenta pulsando sobre el botón correspondiente. 3. Cambiar los datos empleando los campos proporcionados por la aplicación y pulsando el botón guardar. 4. La aplicación valida los cambios y redirige al usuario a la página principal del perfil personal.
Post-condición	La aplicación registrará los cambios de datos proporcionados por el usuario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.10: CU-10, Iniciar entrenamiento

Nombre	Iniciar un entrenamiento del plan.
Objetivo	Realizar un entrenamiento del plan.
Descripción	El usuario realizará un entrenamiento en la aplicación.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe tener un plan de entrenamiento activo en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. Abrir el menú desplegable de la aplicación y pulsar sobre el botón de entrenamientos. 2. Seleccionar entrenamiento de la lista de entrenamientos y pulsar el botón de empezar entrenamiento. 3. La aplicación redirigirá a la pantalla donde el usuario rellenará los campos proporcionados con los datos de su entrenamiento y cuando acabe pulsará el botón de guardar. 4. La aplicación validará los datos registrando el entrenamiento en la aplicación y redirigirá al usuario a la pestaña anterior.
Post-condición	La aplicación registrará el entrenamiento en el plan de entrenamientos del usuario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrara un aviso.

Tabla B.11: CU-011, Crear plan de entrenamiento

Nombre	Crear plan de entrenamiento.
Objetivo	Crear un plan de entrenamiento.
Descripción	El usuario creará un plan de entrenamiento.
Pre-condición	El usuario debe tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de entrenamientos, pulsar el botón de planes de entrenamiento. 2. La aplicación redirige al usuario a la biblioteca de planes, pulsará el botón de añadir plan y será redirigido a la pantalla de creación de planes. 3. El usuario rellenará los campos proporcionados y agregará los entrenamientos que desea pudiendo modificarlos a su gusto, después pulsará el botón de guardar. 4. La aplicación validará los datos y redirigirá a la pantalla anterior.
Post-condición	La aplicación registrará el plan de entrenamientos en la base de datos del usuario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.12: CU-012, Modificar plan de entrenamiento

Nombre	Modificar plan de entrenamiento.
Objetivo	Modificar un plan de entrenamiento.
Descripción	El usuario modificará un plan de entrenamiento.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe tener un plan de entrenamientos creado.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de entrenamientos, pulsar el botón de planes de entrenamiento. 2. La aplicación redirige al usuario a la biblioteca de planes, pulsará el plan que quiera editar de la lista y se le redirigirá a una página donde editarlo. 3. El usuario rellenará los campos proporcionados y modificará los entrenamientos que desee pudiendo modificarlos a su gusto, después pulsará el botón de guardar. 4. La aplicación validará los datos y redirigirá a la pantalla anterior.
Post-condición	La aplicación actualizará el plan de entrenamientos en la base de datos del usuario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.13: CU-013, Borrar plan de entrenamiento

Nombre	Borrar plan de entrenamiento.
Objetivo	Borrar un plan de entrenamiento.
Descripción	El usuario borrará un plan de entrenamiento.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe tener un plan de entrenamientos creado
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de entrenamientos, pulsar el botón de planes de entrenamiento. 2. La aplicación redirige al usuario a la biblioteca de planes, pulsará el plan que quiera borrar de la lista y se le abrirá un submenú con la opción para borrar el plan. 3. Tras pulsar el botón, la aplicación actualizará la base de datos eliminando el seleccionado.
Post-condición	La aplicación borrará el plan de entrenamientos en la base de datos del usuario.
Excepciones	

Tabla B.14: CU-014, Crear entrenamiento

Nombre	Crear entrenamiento.
Objetivo	Creación de un entrenamiento.
Descripción	El usuario creará un entrenamiento.
Pre-condición	El usuario debe tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de entrenamientos, pulsar el botón de entrenamiento. 2. La aplicación redirige al usuario a la biblioteca de entrenamientos, pulsará el botón de crear entrenamiento. 3. El usuario rellenará los campos proporcionados y agregará los ejercicios que crea convenientes, pulsará el botón de guardar. 4. La aplicación validará los datos y redirigirá a la pagina anterior.
Post-condición	La aplicación registrará el entrenamiento en la base de datos del usuario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.15: CU-015, Modificar entrenamiento

Nombre	Modificar entrenamiento.
Objetivo	Modificación de un entrenamiento.
Descripción	El usuario modificará un entrenamiento.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe tener un entrenamiento creado.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de entrenamientos, pulsar el botón de entrenamientos. 2. La aplicación redirigirá a la biblioteca de entrenamientos donde el usuario pulsará el entrenamiento a modificar. 3. El usuario modificará los campos proporcionados y agregará los ejercicios que crea convenientes, pulsará el botón de guardar. 4. La aplicación validará los datos y redirigirá a la página anterior.
Post-condición	La aplicación registrará los cambios del entrenamiento en la base de datos del usuario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.16: CU-016, Borrar entrenamiento

Nombre	Borrar entrenamiento.
Objetivo	Borrado de un entrenamiento.
Descripción	El usuario borrará un entrenamiento.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe tener un entrenamiento creado.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de entrenamientos, pulsar el botón de entrenamientos. 2. La aplicación redirigirá a la biblioteca de entrenamientos donde el usuario pulsará el entrenamiento que desea borrar. 3. El usuario pulsará el botón de borrar en el submenú que se abrirá y la aplicación borrará de la base de datos este entrenamiento.
Post-condición	La aplicación borrará el entrenamiento en la base de datos del usuario.
Excepciones	

Tabla B.17: CU-017, Crear ejercicio

Nombre	Crear ejercicio.
Objetivo	Creación de un ejercicio.
Descripción	El usuario creará un ejercicio.
Pre-condición	El usuario debe tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de entrenamientos, pulsar el botón de ejercicios. 2. La aplicación redirigirá a la biblioteca de ejercicios donde el usuario pulsará el botón de añadir ejercicio. 3. El usuario completará los campos proporcionados con los datos y pulsará el botón de guardar. 4. La aplicación validará los datos y registrará el ejercicio en la biblioteca, tras esto, redirigirá al usuario a la página anterior.
Post-condición	La aplicación registrará el ejercicio en la base de datos del usuario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.18: CU-018, Modificar ejercicio

Nombre	Modificar ejercicio.
Objetivo	Modificación de un ejercicio.
Descripción	El usuario modificará un ejercicio.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe tener al menos un ejercicio creado.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de entrenamientos, pulsar el botón de ejercicios. 2. La aplicación redirigirá a la biblioteca de ejercicios donde el usuario pulsará el ejercicio que desea modificar. 3. El usuario completará los campos proporcionados con los datos y pulsará el botón de guardar. 4. La aplicación validará los datos y registrará los cambios en la base de datos.
Post-condición	La aplicación modificará el ejercicio en la base de datos del usuario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.19: CU-019, Borrar ejercicio

Nombre	Borrar ejercicio.
Objetivo	Borrar un ejercicio.
Descripción	El usuario borrará un ejercicio.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe tener al menos un ejercicio creado.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de entrenamientos, pulsar el botón de ejercicios. 2. La aplicación redirigirá a la biblioteca de ejercicios donde el usuario pulsará el ejercicio que desea borrar. 3. El usuario pulsará el botón de borrar. 4. La aplicación borrará el ejercicio de la base de datos.
Post-condición	La aplicación borrará el ejercicio de la base de datos del usuario.
Excepciones	

Tabla B.20: CU-020, Cambiar objetivo calórico

Nombre	Cambiar objetivo calórico.
Objetivo	Cambiar el objetivo calórico del usuario.
Descripción	El usuario modificará su objetivo calórico.
Pre-condición	El usuario debe tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. Abrir el menú desplegable de la aplicación y pulsar sobre el botón de nutrición. 2. La aplicación direccionará al usuario a la página principal de nutrición, pulsará el botón de cambiar objetivo. 3. El usuario rellenará los campos proporcionados por la aplicación y seleccionará su objetivo, pulsará el botón de guardar. 4. La aplicación validará los datos calculando el objetivo, actualizará la entrada en la base de datos y cerrará la pestaña emergente donde el usuario introdujo los datos.
Post-condición	La aplicación registrará el cambio de objetivo en la cuenta del usuario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.21: CU-021, Añadir comida

Nombre	Añadir comida.
Objetivo	Añadir una comida.
Descripción	El usuario añadirá una comida a su biblioteca.
Pre-condición	El usuario debe tener su sesión activa en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de nutrición, pulsar el botón de añadir comida. 2. La aplicación redirigirá a la página donde el usuario completará los campos proporcionados con la información de la comida que desea añadir, pulsará el botón de añadir comida. 3. La aplicación validará los datos y añadirá la comida a la base de datos. También, si el usuario lo ha seleccionado, se agregará al día. Tras esto se le redirigirá a la página anterior.
Post-condición	La aplicación registrará la comida en la base de datos del usuario y, si el usuario lo ha indicado, añadirá la comida al día y restará las calorías de la misma a las calorías que le falten para completar su objetivo diario.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.22: CU-022, Modificar comida

Nombre	Modificar comida.
Objetivo	Modificar una comida.
Descripción	El usuario modificará una comida a su biblioteca.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe de tener una comida creada en su base de datos.
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de nutrición, pulsar el botón de comida. 2. La aplicación redirigirá a la página donde el usuario elegirá la comida que desea modificar pulsando sobre ella y luego sobre el botón de editar. 3. El usuario modificará los valores que considere de la comida y pulsará el botón de guardar. 4. La aplicación validará los datos y agregará la comida a la biblioteca. También redirigirá al usuario a la pagina anterior.
Post-condición	La aplicación modificará los datos de la comida en la base de datos.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.

Tabla B.23: CU-023, Borrar comida

Nombre	Borrar comida.
Objetivo	Borrar una comida.
Descripción	El usuario borrará una comida de su biblioteca.
Pre-condición	El usuario debe tener su sesión activa en la aplicación. El usuario debe de tener una comida creada en su base de datos
Secuencia	<ol style="list-style-type: none"> 1. En la página principal de nutrición, pulsar el botón de comida. 2. La aplicación redirigirá a la página donde el usuario elegirá la comida que desea borrar pulsando sobre ella y luego sobre el botón de borrar. 3. La aplicación borrará la comida de la biblioteca y cerrará el submenú.
Post-condición	La aplicación borrará la comida de la base de datos.
Excepciones	Si el usuario proporciona datos que no son válidos, la aplicación mostrará un aviso.