



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Aplicación móvil para la definición y seguimiento de rutinas
de entrenamiento

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Soler Martín, Adrián

Tutor/a: Canós Cerdá, José Hilario

CURSO ACADÉMICO: 2022/2023

Resumen

El objetivo de este trabajo ha sido desarrollar una aplicación móvil para el ejercicio físico. La aplicación desarrollada permite a los usuarios crear y realizar rutinas de ejercicio, tanto en línea como sin conexión a internet. Los usuarios tienen la opción de crear rutinas públicas o privadas, y también pueden explorar y guardar rutinas públicas creadas por otros usuarios para realizarlas en cualquier momento, así como, comentar y calificar las rutinas.

La aplicación también ofrece la función de crear grupos de ejercicio, donde los usuarios pueden publicar rutinas y chatear con otros miembros del grupo. Cada usuario tiene un perfil personalizado donde puede consultar sus propias rutinas, ver un calendario con las rutinas realizadas cada día y planificar su ejercicio. Además, los usuarios pueden aumentar su nivel a medida que realizan ejercicio de manera regular.

Palabras clave: Aplicación móvil, ejercicio físico, Android, metodología ágil.

Abstract

The objective of this work has been to develop a mobile application for physical exercise. The developed application allows users to create and perform exercise routines, both online and offline. Users have the option to create public or private routines and can also browse and save public routines created by other users to perform them at any time, as well as comment and rate the routines.

The application also offers the function to create exercise groups, where users can post routines and chat with other group members. Each user has a custom profile where they can view their own routines, see a calendar with the routines performed each day and plan their exercise. In addition, users can increase their level as they exercise on a regular basis.

Keywords: Mobile application, training, Android, agile methodology.

Resum

L'objectiu d'aquest treball ha sigut desenvolupar una aplicació mòbil per a l'exercici físic. L'aplicació desenvolupada permet als usuaris crear i realitzar rutines d'exercici, tant en línia com sense connexió a internet. Els usuaris tenen l'opció de crear rutines públiques o privades, i també poden explorar i guardar rutines públiques creades per altres usuaris per a realitzar-les en qualsevol moment, així com, comentar i qualificar les rutines.

L'aplicació també ofereix la funció de crear grups d'exercici, on els usuaris poden publicar rutines i xatejar amb altres membres del grup. Cada usuari té un perfil personalitzat on pot consultar les seues pròpies rutines, veure un calendar amb les rutines realitzades cada dia i planificar el seu exercici. A més, els usuaris poden augmentar el seu nivell a mesura que realitzen exercici de manera regular.

Paraules clau: Aplicació mòbil, exercici físic, Android, metodologia àgil.

Tabla de contenidos

1.	Introducción.....	6
1.1.	Motivación	6
1.2.	Objetivos	7
1.3.	Impacto Esperado	7
1.4.	Estructura.....	7
2.	Estado del arte.....	9
2.1.	Crítica al estado del arte.....	15
2.2.	Propuesta.....	16
3.	Análisis del problema.....	18
3.1.	Especificación de Requisitos	18
3.1.1.	Glosario.....	18
3.1.2.	Visión	19
3.1.3.	Casos de Uso	23
3.2.	Identificación y análisis de soluciones posibles	28
	Aplicaciones específicas para cada plataforma.....	28
	Aplicación web.....	28
3.3.	Solución propuesta	29
3.4.	Plan de Trabajo.....	29
3.4.1.	Metodología	29
3.4.2.	Aplicación de la metodología	31
3.5.	Presupuesto	33
4.	Diseño de la solución.....	34
4.1.	Arquitectura del Sistema.....	34
4.2.	Tecnología Utilizada	35
4.2.1.	Herramientas	35
4.3.	Diseño Detallado.....	38
4.3.1.	Clases de la aplicación.....	38
5.	Desarrollo de la solución propuesta	42
5.1.	Primer Sprint	42
5.2.	Segundo Sprint	47
5.3.	Tercer Sprint	48
6.	Pruebas.....	50
6.1.	Pruebas unitarias	50
6.2.	Pruebas de interfaz de usuario (UI)	51



6.3. Análisis estático del código.....	53
7. Conclusiones.....	55
7.1. Relación con los ODS.....	57
7.2. Trabajos futuros	57
7.3. Relación del trabajo desarrollado con los estudios cursados.....	59
8. Referencias	60
Anexo A.....	62
Anexo B.....	67

1. Introducción

1.1. Motivación

En la actualidad, el sedentarismo se ha convertido en uno de los principales problemas de salud pública a nivel mundial. El ejercicio físico regular, por otro lado, ha demostrado ser beneficioso para la salud tanto física como mental. Sin embargo, muchas personas encuentran dificultades para mantener una rutina de ejercicio constante debido a diferentes factores, como la falta de motivación o la falta de tiempo.

Además, durante el período de confinamiento estricto, muchas personas experimentaron limitaciones significativas en cuanto a la práctica de ejercicio físico fuera de sus hogares. Durante esos primeros meses críticos, los gimnasios y otras instalaciones de entrenamiento permanecieron cerrados, y solo más tarde se permitió el acceso con ciertas restricciones, como el uso obligatorio de mascarillas. Estas limitaciones adicionales dificultaron aún más el mantenimiento de una rutina regular de ejercicio físico.

Por otra parte, organizaciones de salud reconocidas, como la Organización Mundial de la Salud (OMS) recomienda realizar al menos 150 minutos de actividad física aeróbica de intensidad moderada a vigorosa cada semana. También se sugiere complementar la actividad aeróbica con ejercicios de fortalecimiento muscular que involucren los principales grupos musculares, al menos dos veces por semana. Es importante mencionar que estas pautas son generales, y que es recomendable adaptarlas a las capacidades individuales y encontrar un equilibrio entre la actividad física, el descanso y la recuperación para mantener una rutina sostenible y segura a largo plazo. (Organización Mundial de la Salud, 2021)

Otro aspecto importante que tiene hacer ejercicio, no solo a nivel físico sino también a nivel mental, es que según un estudio de la OMS está demostrado que realizar ejercicio reduce el riesgo de depresión gracias a la generación de endorfinas, conocidas como las “hormonas de la felicidad”, que pueden mejorar el estado de ánimo y reducir los síntomas depresivos. También ayuda a aliviar la ansiedad, ya que la actividad física aumenta la producción de neurotransmisores, como la serotonina. Además, también se nombran otros efectos positivos como mejorar la autoestima y la confianza, potenciar la función cognitiva y promover el sueño saludable. (Organización Mundial de la Salud, 2021)

Asimismo, realizar ejercicio puede ser un elemento social, ya que no solo se limita a actividades individuales, sino que también puede ser una actividad compartida y participativa que fomenta la interacción social y fortalece las relaciones personales. Además, realizar ejercicio en grupo no solo agrega diversión y motivación, sino que también brinda la oportunidad de compartir experiencias, establecer relaciones y crear un sentido de comunidad.

Todos estos motivos nos han llevado a desarrollar una aplicación para facilitar a los usuarios que puedan realizar ejercicio de una forma rápida, divertida, saludable y con la posibilidad de realizarlo en grupo.

1.2. Objetivos

Los objetivos que se pretenden conseguir con este trabajo son los siguientes:

- Animar a la sociedad a practicar más ejercicio.
- Facilitar la práctica de ejercicio.
- Promover unos hábitos de vida saludables.
- Fomentar el intercambio de conocimiento acerca de la actividad física entre los usuarios.
- Permitir al usuario realizar un seguimiento del ejercicio que realiza en su día a día.
- Permitir a los usuarios que puedan planificar sus entrenamientos.
- Permitir que la aplicación funcione con o sin conexión a internet.

1.3. Impacto Esperado

Con el desarrollo de este producto, esperamos poder fomentar un estilo de vida saludable. Al permitir a los usuarios encontrar rutinas de ejercicios de una forma rápida y sencilla, facilitamos que se pueda acceder a realizar ejercicio en cualquier momento. Esto es esencial para que los usuarios principiantes no se sientan abrumados la primera vez que accedan a la aplicación y para animar a más personas a practicar ejercicio.

Al ofrecer al usuario la posibilidad de realizar un seguimiento del ejercicio realizado y planificar un calendario con los entrenamientos que se realizarán cada día, le brindamos una herramienta que le ayudará a mantenerse motivado y organizado en su práctica de ejercicio.

Además, esperamos que la aplicación ayude a que se puedan crear grupos de entrenamiento para que los usuarios practiquen ejercicio con amigos, familiares, etc. De esta forma, puede conseguirse que la experiencia a la hora de realizar ejercicio sea más divertida y amena, especialmente para aquellos que están empezando. Esperamos lograr este impacto al permitir dentro de la aplicación la creación de grupos de ejercicio a los cuales otros usuarios puedan unirse y compartir rutinas y comunicarse mediante un chat.

1.4. Estructura

En este apartado vamos a detallar la estructura del documento haciendo hincapié en el contenido que se puede encontrar en cada capítulo.

En el capítulo 2, denominado “**Estado del arte**”, se lleva a cabo un estudio de mercado sobre las diversas aplicaciones existentes en la actualidad que ofrecen soluciones a nuestro problema. Se realiza una crítica al estado actual del mercado y, finalmente, se presenta una propuesta para abordar dicho problema.

En el capítulo 3, titulado “**Análisis del problema**”, se lleva a cabo un análisis exhaustivo de los requisitos y se identifican oportunidades de innovación. Se presentan los modelos de datos que se utilizarán en el desarrollo de la aplicación, junto con una identificación y análisis de posibles soluciones al problema. Se evalúan los pros y contras de cada solución para determinar la más adecuada, seguida de la presentación de un plan de trabajo detallado y un presupuesto que estima los costos del proyecto.

En el capítulo 4, titulado “**Diseño de la solución**”, se presenta la arquitectura del sistema y el diseño de la base de datos. Además, se destacan las herramientas, tecnologías y *frameworks*¹ utilizados durante el desarrollo de la aplicación.

En el capítulo 5, denominado “**Desarrollo de la solución propuesta**”, se describe en detalle el desarrollo de la aplicación, abordando los problemas y dificultades encontrados, así como las decisiones tomadas durante el proceso. Se detallan los hitos alcanzados, las etapas de implementación y se analizan los desafíos técnicos y soluciones adoptadas.

En el capítulo 6, titulado “**Pruebas**”, se describen detalladamente las pruebas utilizadas para verificar el funcionamiento de la aplicación y detectar la mayor cantidad de errores posible.

En el capítulo 7, denominado “**Conclusiones**”, se presentarán las conclusiones a las que se ha llegado tras el desarrollo del proyecto. En este apartado, se realizará un resumen de los principales hallazgos, logros y desafíos encontrados durante todo el proceso. Se analizarán los resultados obtenidos en relación con los objetivos planteados, se evaluará el impacto de la solución propuesta y se destacarán lecciones aprendidas. Además, se ofrecerán recomendaciones para futuros proyectos similares.

En el capítulo 8, titulado “**Referencias**”, se presentan las fuentes bibliográficas y recursos utilizados para obtener información relevante durante el desarrollo del trabajo. Se incluyen referencias a libros, artículos, estudios, sitios web u otras fuentes consultadas.

¹ Un *framework* es un conjunto de herramientas y estructuras predefinidas que facilitan el desarrollo de aplicaciones al proporcionar un esqueleto y funcionalidades comunes.

2. Estado del arte

En este apartado, se llevará a cabo un estudio de las soluciones que existen en la actualidad, centrándose en las aplicaciones más utilizadas y destacando sus características principales. Dado que hay una amplia variedad de aplicaciones disponibles, se realizará un análisis selectivo para abordar aquellas que tienen una mayor adopción y reconocimiento en el mercado. El objetivo es proporcionar una visión general de las soluciones existentes y resaltar sus aspectos más relevantes para comprender el panorama actual de las aplicaciones en el ámbito del ejercicio físico.

Nike Training Club Ejercicio

La primera aplicación que analizaremos es Nike Training Club Ejercicio². Esta aplicación está desarrollada por Nike, una de las marcas de deporte más reconocidas en la actualidad, cuenta con más de 10 millones de descargas en Google Play Store y una valoración de 4.2 estrellas sobre 5.

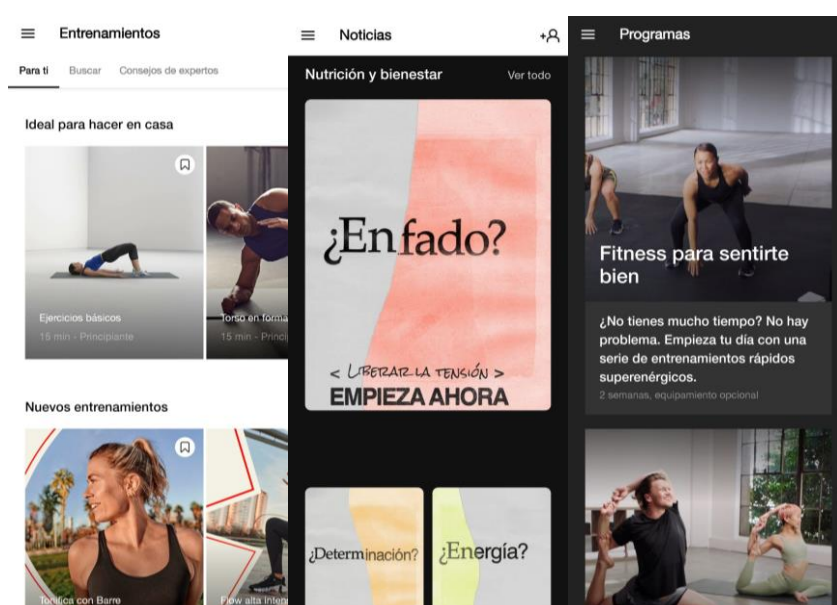


Figura 1. Imágenes Nike Training Club Ejercicio

A continuación, destacamos algunas de sus características más relevantes:

- Los usuarios pueden acceder a rutinas en formato de video diseñadas por entrenadores para adaptarse a diferentes niveles de condición física y preferencias de entrenamiento.
- Cuenta con un apartado de noticias donde se comparte información relevante sobre ejercicio, nutrición y salud, brindando a los usuarios recursos adicionales y consejos útiles.
- Se ofrece un historial completo de los entrenamientos realizados por el usuario, lo que permite realizar un seguimiento de su progreso a lo largo del tiempo.
- Cuenta con un sistema de logros que motiva a los usuarios a alcanzar metas y objetivos.
- La integración con Google Fit permite obtener datos de actividad física registrados en otras aplicaciones o dispositivos, lo que brinda una visión más completa de la actividad física realizada por el usuario.

² <https://bit.ly/3pjk1aC>

- La opción de enviar la pantalla a otros dispositivos brinda flexibilidad al permitir a los usuarios seguir las rutinas en una pantalla más grande, como un televisor o una tableta.
- Durante los entrenamientos, la aplicación ofrece instrucciones y guía a través de un entrenador virtual por voz, que proporciona indicaciones sobre los ejercicios a realizar, el tiempo, y otros detalles relevantes.
- La interfaz de la aplicación sigue el diseño característico de los productos de Nike, y se destaca por ser intuitiva y fácil de usar, lo que facilita su utilización tanto para nuevos usuarios como para aquellos familiarizados con otros productos de la marca.

La aplicación presenta las siguientes limitaciones:

- No ofrece la posibilidad de crear rutinas personalizadas según las preferencias y necesidades individuales de los usuarios. Esto limita la capacidad de adaptar los entrenamientos a objetivos específicos o condiciones físicas particulares.
- No permite la creación de ejercicios personalizados, lo que restringe la capacidad de agregar ejercicios únicos o específicos que no estén incluidos en las rutinas preestablecidas de la aplicación.
- Carece de la función para crear planes de entrenamiento personalizados, lo que impide que los usuarios organicen y programen sus entrenamientos de acuerdo con sus propias metas y disponibilidad de tiempo.

Adidas Training: Workout HIIT

La aplicación Adidas Training: Workout HIIT³, desarrollada por Adidas, otra reconocida marca deportiva, cuenta con más de 10 millones de descargas en Google Play Store y una valoración de 4.5 estrellas.

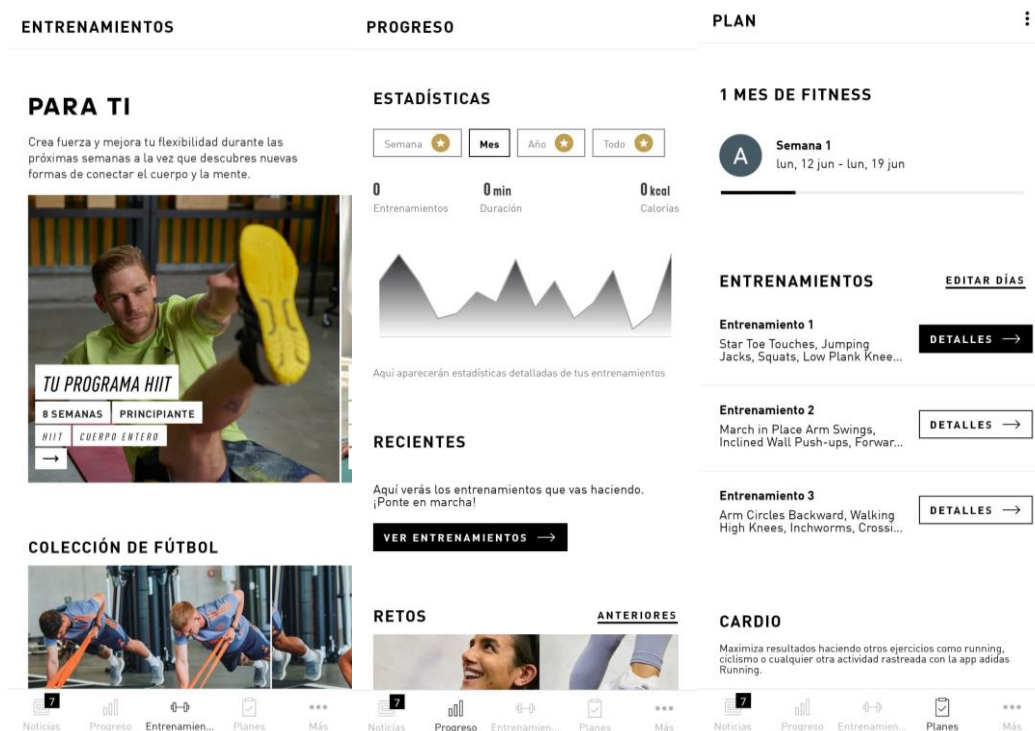


Figura 2. Imágenes Adidas Training: Workout HIIT

³ <https://bit.ly/3Pt0FdD>

Sus características principales son las siguientes:

- Al iniciar la aplicación y crear una cuenta, se solicita información personalizada para recomendar el plan de entrenamiento que más adecuado según el estado físico y los intereses del usuario.
- Ofrece una amplia variedad de rutinas de entrenamiento predefinidas diseñadas por entrenadores profesionales, abarcando diferentes niveles de condición física y objetivos. Además, se pueden seleccionar los días que se desea entrenar al configurar un plan de entrenamiento personalizado.
- Cuenta con un apartado de noticias donde los usuarios pueden crear publicaciones visibles para otros usuarios. También permite consultar las publicaciones de amigos de Facebook y contactos del teléfono, fomentando la interacción social y el intercambio de experiencias.
- Permite realizar un seguimiento de la actividad física, proporcionando estadísticas y métricas para evaluar el progreso del usuario en su entrenamiento.
- La aplicación ofrece retos y consejos relacionados con la salud y la nutrición, brindando una experiencia integral para el bienestar general.
- Dispone de un entrenador por voz que guía al usuario durante los entrenamientos, proporcionando instrucciones y motivación adicional.
- Cuenta con integración con Google Fit.
- La interfaz de la aplicación sigue el estilo característico de los productos de Adidas, siendo fácil de utilizar para los nuevos usuarios y manteniendo una estética coherente con la marca.

La aplicación tiene las siguientes limitaciones:

- No permite crear rutinas de entrenamiento personalizadas según las preferencias y necesidades del usuario.
- No permite la creación de ejercicios personalizados, limitando las opciones de personalización del entrenamiento.
- Los planes de entrenamiento predefinidos no permiten seleccionar los ejercicios específicos que el usuario desea realizar, lo que puede limitar la adaptabilidad de los programas a las preferencias individuales.
- Algunas funciones y características están restringidas y requieren una suscripción premium para acceder a ellas. Por ejemplo, ciertos planes de entrenamiento pueden estar bloqueados y no estar disponibles de forma gratuita, así como algunas estadísticas avanzadas o características adicionales.



Strong Workout Tracker Gym Log

La aplicación Strong Workout Tracker Gym Log⁴ desarrollada por Strong Fitness, destaca por su amplio conjunto de funciones y características, lo que la convierte en una opción muy completa.

Aunque cuenta con 1 millón de descargas, ha recibido una alta valoración de 4.7 estrellas en Google Play Store. Esta aplicación particularmente relevante para nuestro proyecto, ya que comparte similitudes con la que estamos desarrollando.

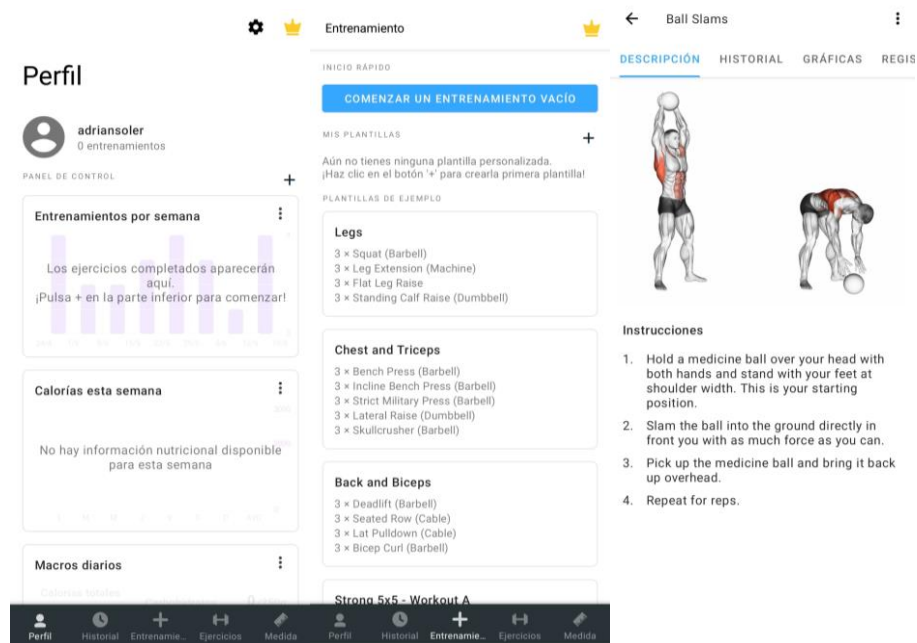


Figura 3. Imágenes Strong Workout Tracker Gym Log

Algunas de sus características principales son:

- Ofrece una amplia variedad de rutinas de ejercicio predefinidas y brinda la posibilidad de crear rutinas personalizadas según las preferencias y necesidades del usuario.
- Proporciona una lista extensa de ejercicios predeterminados y permite al usuario crear ejercicios personalizados para adaptarse a su entrenamiento específico.
- Durante la realización de una rutina, el usuario tiene la flexibilidad de añadir ejercicios adicionales y ajustar el tiempo de descanso en tiempo real, brindando mayor control y adaptabilidad.
- Permite realizar un seguimiento de la actividad física, registrando y almacenando el historial de entrenamientos realizados.
- Proporciona estadísticas y análisis de los entrenamientos y ejercicios.
- Tiene integración con Google Fit.
- Su interfaz intuitiva y de diseño nativo permite a los usuarios iniciar rápidamente una rutina de ejercicio. Los elementos en pantalla son fácilmente reconocibles, lo que facilita la navegación y comprensión de las funcionalidades de la aplicación.

⁴ <https://bit.ly/43QfmMe>

Las limitaciones son las siguientes:

- Algunas características y funcionalidades avanzadas están reservadas para los usuarios que adquieren la versión premium de la aplicación, la cual se puede obtener mediante una suscripción mensual, anual o una compra única. Los precios de las suscripciones son de 5,49€ al mes, 32,99€ al año y 99,99€ como pago único.
- Con la versión gratuita de la aplicación, existe un límite de tres rutinas personalizadas que se pueden crear y almacenar.
- No cuenta con la capacidad de planificar entrenamientos en un calendario o programar sesiones de ejercicio de manera específica.

Google Fit: Registro de Actividad

La aplicación Google Fit: Registro de Actividad⁵, desarrollada por Google, destaca por ser una de las más populares en el ámbito del ejercicio. Cuenta con más de 100 millones de descargas y una valoración de 4 estrellas.

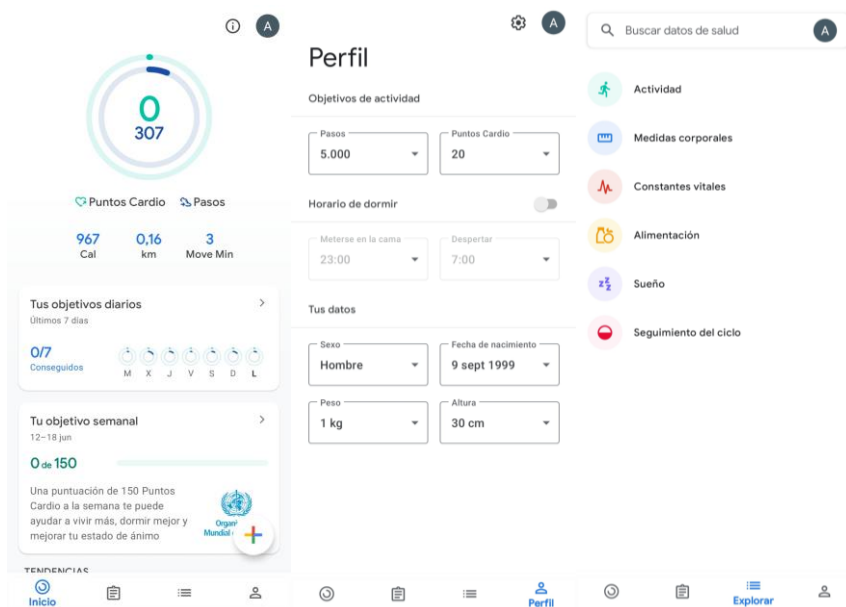


Figura 4. Imágenes Google Fit

Sus características principales son:

- Registra automáticamente las actividades físicas del usuario y también brinda la opción de crear actividades personalizadas.
- Mantiene un historial detallado de las actividades realizadas, permitiendo al usuario acceder a estadísticas específicas de cada actividad.
- Permite establecer objetivos de actividad personalizados, como pasos diarios o calorías quemadas, y proporciona notificaciones y motivación para alcanzar esos objetivos.
- Incluye un apartado de noticias que ofrece información relevante sobre recomendaciones de salud y ejercicio, provenientes de fuentes confiables la OMS.
- Su interfaz sigue los principios del diseño *Material Design*⁶, ofreciendo una experiencia intuitiva y sencilla para los usuarios.

⁵ <https://bit.ly/44d1KdR>

Las limitaciones son las siguientes:

- Su enfoque principal se centra en recopilar información y estadísticas sobre la actividad física del usuario, en lugar de proporcionar herramientas para la creación de rutinas de ejercicio.
- No cuenta con funcionalidades para planificar y programar entrenamientos personalizados en la aplicación.

Además de las aplicaciones mencionadas, existen otras opciones en el mercado que permiten realizar ejercicio, aunque pueden tener funcionalidades más limitadas en comparación con las mencionadas anteriormente.

⁶ Es un sistema de diseño visual desarrollado por Google para crear interfaces coherentes y atractivas en diversas plataformas y dispositivos.

2.1. Crítica al estado del arte

La Tabla 1 resume las características que tienen las aplicaciones analizadas de una forma estructurada.

Tabla 1. Estudio de productos competidores

Nombre	Nike Training Club Ejercicio	Adidas Training Workout HIIT	Strong Workout Tracker Gym Log	Google Fit: Reg. de actividad
Rutinas de ejercicio	✓	✓	✓	✗
Rutinas de ejercicio personalizadas	✗	✗	✓	✗
Ejercicios predeterminados	✓	✓	✓	✗
Ejercicios personalizados	✗	✗	✓	✓
Planes de entrenamiento	✗	✓	✗	✗
Noticias	✓	✓	✗	✓
Seguimiento de la actividad física	✓	✓	✓	✓
Chat y grupos	✗	✗	✗	✗
Calendario de actividad	✗	✗	✓	✗
Integración de relojes inteligentes	✗	✗	✗	✓
Enviar pantalla a otros dispositivos	✓	✗	✗	✗
Entrenador por voz	✓	✓	✗	✗
Integración con Google Fit	✓	✓	✓	✓

Como hemos comentado en las revisiones anteriores, una limitación común de las aplicaciones existentes es la falta de opciones de personalización. La mayoría de estas aplicaciones no permiten al usuario crear rutinas o ejercicios personalizados. Esto puede generar la sensación de estancamiento y dificultar el progreso del usuario, ya que las rutinas predefinidas pueden no adaptarse a sus necesidades específicas. Es importante abordar esta limitación en nuestra aplicación para brindar a los usuarios una experiencia más flexible y adaptada a sus objetivos individuales.

Otro problema común que se encuentra en estas aplicaciones es la restricción de funciones en la versión gratuita, que requiere una suscripción de pago para acceder a todas las características y beneficios completos. Esto puede limitar la experiencia de los usuarios y dificultar su capacidad para aprovechar al máximo la aplicación. Además, algunas personas pueden optar por no suscribirse debido a restricciones económicas o simplemente porque no desean comprometerse con pagos recurrentes.

Es cierto que muchas de las aplicaciones existentes se centran en el ejercicio individual y no fomentan la creación de grupos de ejercicio. Esta falta de interacción social puede desmotivar a algunos usuarios y hacer que perciban el ejercicio como algo aburrido o monótono. Sin embargo, la creación de grupos de ejercicio puede ser una excelente manera de aumentar la motivación y hacer que la experiencia de hacer ejercicio se más divertida y enriquecedora.

2.2. Propuesta

Una vez analizadas las distintas soluciones ya existentes, proponemos desarrollar una aplicación que supere las limitaciones mencionadas. Específicamente, pretendemos implementar las siguientes funcionalidades:

- Creación de ejercicios personalizados para dar más flexibilidad a los usuarios a la hora de crear sus rutinas
- Creación de rutinas de ejercicios personalizadas para que se adapten a las necesidades del usuario.
- Que el usuario pueda configurar a su gusto cualquier aspecto de la rutina, como el orden de los ejercicios, la visibilidad de la rutina, el modo en el que se realiza cada ejercicio, el tiempo de descanso entre ejercicios, etc.
- Creación de planes de entrenamiento, donde el usuario puede configurar cuando realizar los entrenamientos y la rutina o rutinas que quiere realizar en cada entrenamiento.
- Realizar un seguimiento del ejercicio del usuario y mostrar estadísticas para visualizar información sobre los ejercicios.
- Ver las rutinas y entrenamientos realizados en un calendario de actividad.
- Creación de objetivos para que el usuario pueda tener metas y motivarse para conseguirlas.

Características diferenciadoras de nuestra propuesta:

- Que los usuarios tengan un nivel que pueden aumentar realizando rutinas, entrenamientos, consiguiendo objetivos, etc. Esto puede hacer que los usuarios se motiven para alcanzar el nivel de sus amigos o conocidos.
- Que los usuarios compartan sus rutinas y puedan utilizar rutinas de otros usuarios para realizar ejercicio.
- Crear grupos de ejercicio con un chat donde los integrantes pueden enviar mensajes y compartir rutinas y planes de entrenamiento. Esta característica puede ser interesante para entrenadores personales, ya que pueden administrar sus clientes y enviarles sus rutinas para que puedan realizarlas directamente en la aplicación.

A diferencia de las aplicaciones analizadas, nuestra propuesta se centra en fomentar la interacción entre los usuarios, permitiéndoles compartir y utilizar rutinas y planes de entrenamiento publicados por los otros usuarios. Además, promovemos la creación de grupos de ejercicio donde los usuarios puedan compartir sus progresos, rutinas, planes de entrenamiento, entre otros, formando así comunidades de ejercicio.

Al permitir a los usuarios formar grupos de ejercicio dentro de nuestra aplicación, les brindamos la oportunidad de conectarse con amigos, familiares u otros usuarios con intereses similares. Esto les permite compartir rutinas, intercambiar consejos y animarse mutuamente a realizar ejercicio. Además, al tener acceso a las rutinas de otros usuarios, pueden descubrir nuevas rutinas que pueden ayudarles a mejorar su estado físico.

La incorporación de características de grupo con chat también puede promover la interacción y el apoyo mutuo entre los miembros del grupo. Esto crea un sentido de comunidad y pertenencia, lo que puede aumentar la motivación y el compromiso de los usuarios con el ejercicio.



3. Análisis del problema

En este apartado, se llevará a cabo un análisis del problema que se pretende resolver. En primer lugar, se realizará una especificación de requisitos para identificar los objetivos, características y restricciones de nuestro sistema. Este documento servirá como guía para el desarrollo del producto. A continuación, se identificarán y analizarán las posibles soluciones. Una vez completado este análisis, se detallará la solución seleccionada y se explicará porque se considera la mejor opción para abordar el problema. Una vez que se haya establecido la solución, se describirá el plan de trabajo en detalle. Se explicará la metodología que se utilizará y como se aplicará para llevar a cabo el desarrollo del producto. Por último, se elaborará un presupuesto que nos permitirá tener una idea del coste que implicaría el desarrollo del producto en un entorno real.

3.1. Especificación de Requisitos

Vamos a dividir la especificación en tres apartados: glosario, visión y casos de uso. El glosario incluirá la descripción de los términos más relevantes que se van a utilizar en el proyecto, para así crear un consenso y que quede claro a qué nos estamos refiriendo cuando nombramos algo. En el apartado de visión, se incluirán los actores, el diagrama de contexto, el diagrama de clases, la especificación de características y los requisitos no funcionales del proyecto, para así mostrar cómo va a ser el producto que vamos a crear y qué características tiene. Por último, en el apartado de Casos de Uso, se incluirán los diagramas de Casos de Uso y una descripción de cada uno de ellos para mostrar cómo se realizaría cada acción dentro del sistema y quién realizaría dicha acción.

3.1.1. Glosario

A continuación, vamos a definir los términos básicos que se van a utilizar a lo largo del desarrollo.

Término	Significado
Usuario	El usuario es la persona que utiliza la aplicación.
Rutina	Una rutina es un conjunto definido de ejercicios físicos que se realizan en un orden concreto.
Ejercicio	Un ejercicio es una actividad física que mejora y mantiene la aptitud física.
Grupo	Un grupo en un colectivo de usuarios que pueden compartir rutinas entre ellos y mensajes.
Mensaje	Un mensaje es un texto enviado por el grupo.
Comentario	Es un texto que se asocia a una rutina en la que el usuario puede dar su opinión sobre la rutina.
Plan de entrenamiento	Un plan de entrenamiento es la planificación de los entrenamientos a lo largo del tiempo.
Entrenamiento	Es la definición de las rutinas que se van a realizar ese día.
Registro diario	Almacena las rutinas que ha realizado el usuario en un día.
Objetivo	Es un objetivo que el usuario quiere alcanzar.

3.1.2. Visión

Objetivos

Los objetivos de esta sección son los siguientes:

- Mostrar el diagrama de contexto, que ilustra las interacciones y relaciones del sistema con entidades externas.
- Describir los actores del sistema, es decir, las entidades o roles que interactúan con el sistema.
- Detallar las características del sistema, es decir, las funcionalidades y capacidades que ofrece.
- Especificar los requisitos no funcionales del sistema.
- Presentar el diagrama de clases, que representa las entidades y relaciones clave del sistema.

Diagrama de contexto

En este apartado encontramos el diagrama de contexto, que proporciona una visión general del sistema y los actores involucrados. El diagrama de contexto muestra las interacciones entre el sistema y los agentes externos, resaltando los límites del sistema y la comunicación con los actores.

En el diagrama de contexto, se representa el sistema como un bloque central y se identifican los actores externos como entidades separadas fuera del sistema. Estos actores pueden ser usuarios, otros sistemas, dispositivos u otros elementos que interactúan con el sistema de alguna manera.

Es una herramienta útil para comprender las interacciones del sistema con su entorno y establecer los límites del sistema. Proporciona una visión de alto nivel y ayuda a identificar los principales agentes externos y las comunicaciones necesarias para la interacción.

A continuación, en la Figura 5, podemos observar el diagrama de contexto de nuestro sistema.

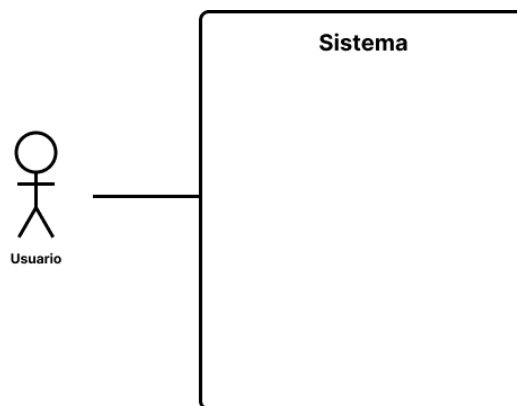


Figura 5. Diagrama de contexto

Como podemos observar en la Figura 5 contiene el siguiente actor:

- **Usuario:** Representa a las personas que utilizan la aplicación y se relacionan con ella a través de la interfaz gráfica de usuario (GUI) o funcionalidades específicas proporcionadas por la aplicación.

Características

El mapa de características es una representación visual de las funcionalidades y características que se desean implementar en una aplicación. A continuación, en la Figura 6, se presenta el mapa de características de nuestra aplicación:

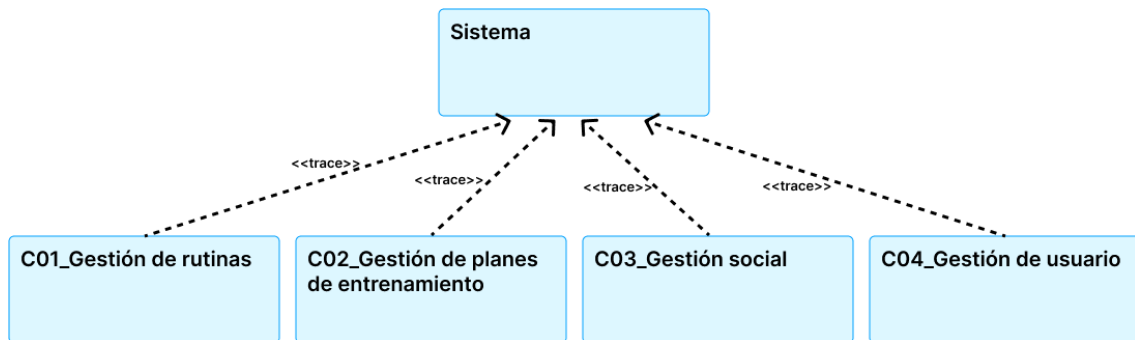


Figura 6. Mapa de características

Las características del sistema son las siguientes:

Gestión de rutinas

- Crear, editar y eliminar rutinas de ejercicios.
- Acceder a la información de una rutina.
- Crear y eliminar ejercicios personalizados.
- Realizar una rutina.
- Votar las rutinas.
- Comentar una rutina.

Gestión de planes de entrenamiento

- Crear, modificar y borrar planes de entrenamiento.
- Consultar un plan de entrenamiento.
- Crear entrenamientos.
- Consultar planes de entrenamiento publicados por otros usuarios.

Gestión social

- Crear grupo de ejercicio.
- Consultar los grupos a los que pertenece.
- Compartir rutinas.
- Compartir planes de entrenamiento.
- Enviar mensajes a todos los integrantes.
- Salir de un grupo.
- Eliminar un grupo.

Gestión de usuario

- Consultar el perfil del usuario
- Editar perfil.
- Definir objetivos.
- Consultar sus estadísticas.
- Consultar calendario entrenamiento.

Requisitos No Funcionales

Estos requisitos se centran en los aspectos de rendimiento, usabilidad, seguridad y otros atributos que impactan en la calidad global del sistema. A continuación, en la Figura 7, podemos observar los requisitos no funcionales del sistema.

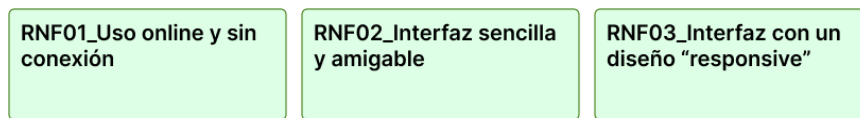


Figura 7. Requisitos no funcionales

Los requisitos no funcionales son los siguientes:

- **Uso online y sin conexión:** Se debe permitir la utilización de la aplicación con y sin conexión a internet para favorecer que se pueda utilizar en cualquier momento.
- **Interfaz sencilla y amigable:** La interfaz de la aplicación debe ser amigable y sencilla de utilizar para favorecer que los nuevos usuarios no encuentren dificultades para utilizar todas las funcionalidades de la aplicación.
- **Interfaz con un diseño *responsive*:** La interfaz de la aplicación debe ajustarse a cualquier tamaño de pantalla para permitir que se pueda usar en la mayor cantidad de dispositivos.

Diagrama de clases

El diagrama de clases es una herramienta fundamental en el diseño y desarrollo de sistemas orientados a objetos. Proporciona una representación visual de las clases, sus atributos, relaciones y métodos, permitiendo una comprensión clara de la estructura y comportamiento del sistema.

A continuación, en la Figura 8, podemos ver el diagrama de clases de nuestro sistema:

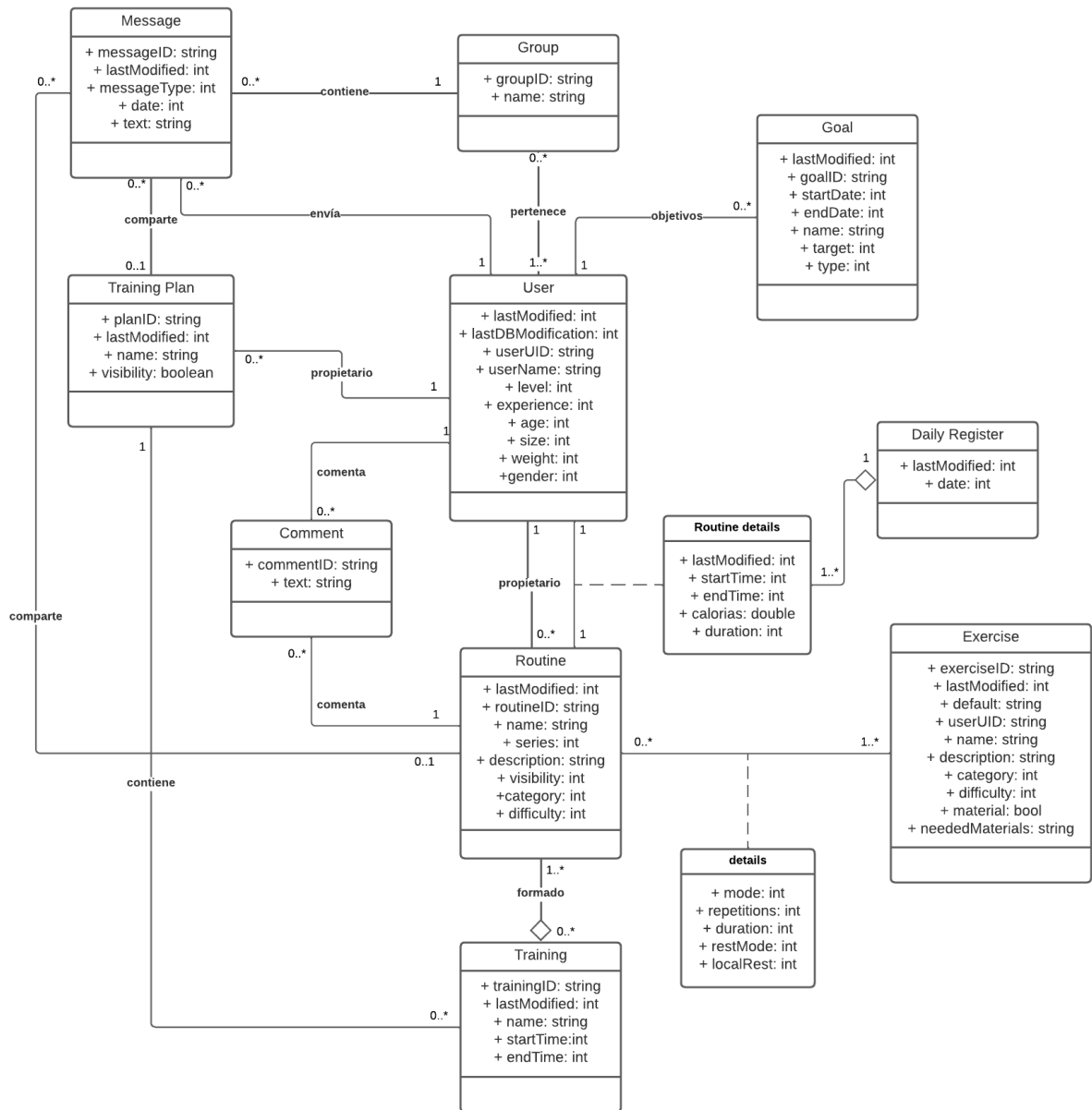


Figura 8. Diagrama de clases

3.1.3. Casos de Uso

A continuación, vamos a mostrar los casos de uso de nuestra aplicación agrupados por características. Para no extender demasiado este apartado solo se van a mostrar algunos casos de uso. El resto se puede encontrar en el Anexo A.

Diagrama: Gestión de rutinas

En la Figura 9, mostramos el diagrama de casos de uso para la característica Gestión de rutinas. Como vemos en la Figura 9, hay dos casos de uso principales, CU01_Crear rutina y CU02_Consultar rutina, que es extendido por cinco casos de uso CU03_Realizar rutina, CU04_Eliminar rutina, CU05_Editar rutina, CU06_Comentar rutina y CU07_Votar rutina.

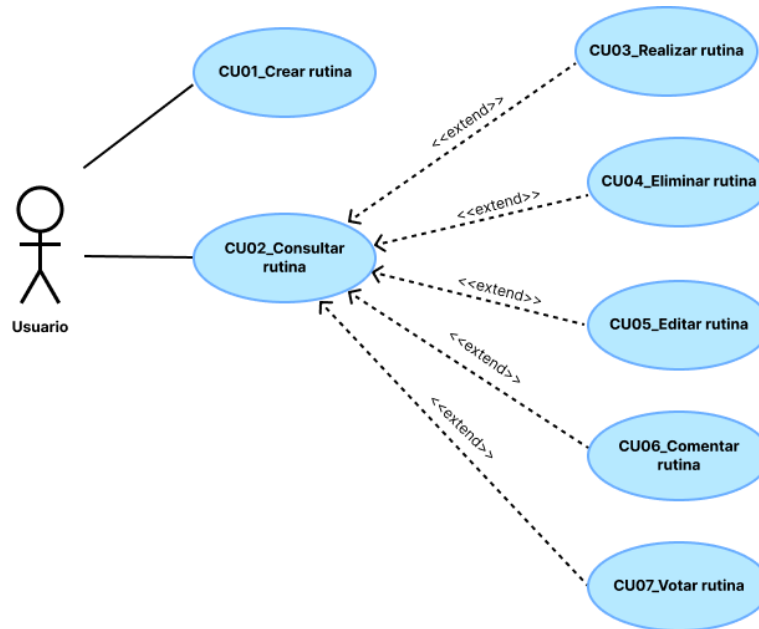


Figura 9. Gestión de rutinas

A continuación, en la Tabla 2 se describe el comportamiento del caso de uso CU01_Crear rutina. Esta funcionalidad le permite al usuario crear una rutina que puede personalizar para que se adapte a sus necesidades.

Tabla 2. CU01_Crear rutina

Descripción	El usuario puede rellenar los datos de la rutina: nombre, descripción, categoría, dificultad, visibilidad, número de series, tiempo de descanso entre series y descanso entre ejercicios.	
Precondición	El usuario ha iniciado sesión y ha accedido a la pantalla de creación de rutina.	
Secuencia normal	Paso	Acción
	1	El usuario introduce el nombre, descripción, categoría, dificultad, visibilidad y número de series de la rutina.
	2	Si el número de series es mayor que uno, el usuario puede seleccionar el tiempo de descanso entre series.
	3	El usuario puede añadir ejercicios pulsando un botón.
	4	El usuario puede ordenar los ejercicios arrastrándolos a la posición deseada.
Postcondición	La rutina se crea con los datos ingresados y los ejercicios añadidos y ordenados según la preferencia del usuario.	
Excepciones	El usuario no completa todos los campos requeridos.	
	El usuario no selecciona ningún ejercicio.	

En la Tabla 3, se describe el comportamiento del caso de uso CU02_Consultar rutina. Aquí el usuario puede consultar una rutina y realizar distintas operaciones sobre ella. Como realizar, eliminar, editar, comentar o votar la rutina que ha consultado.

Tabla 3. CU02_Consultar rutina

Descripción	El usuario puede acceder a la información de la rutina. Una vez accede, puede realizar las siguientes acciones: realizar una rutina, eliminar una rutina, editar rutina, comentar rutina y votar rutina.	
Precondición	El usuario ha iniciado sesión y está en la pantalla de consulta de rutinas.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona una rutina de la lista.
	2	Se muestra la información detallada de la rutina seleccionada.
3	El usuario puede realizar las siguientes acciones: realizar una rutina, eliminar una rutina, editar rutina, comentar rutina y votar rutina.	
Postcondición	La información de la rutina seleccionada se muestra al usuario y se realizan las acciones elegidas (realizar, eliminar, editar, comentar, votar).	
Excepciones	Si no se puede acceder a la información de la rutina, se muestra un mensaje de error.	
	Si ocurre un error al realizar alguna de las acciones (realizar, eliminar, editar, comentar, votar), se muestra un mensaje de error.	

Diagrama: Gestión de planes de entrenamiento

En la Figura 10 encontramos el diagrama de casos de uso para la característica Gestión de planes de entrenamiento. Podemos observar que hay dos casos de uso principales CU08_Crear plan de entrenamiento y CU09_Consultar plan de entrenamiento que es extendido por tres casos de uso CU10_Añadir entrenamiento, CU11_Eliminar entrenamiento y CU03_Realizar rutina.

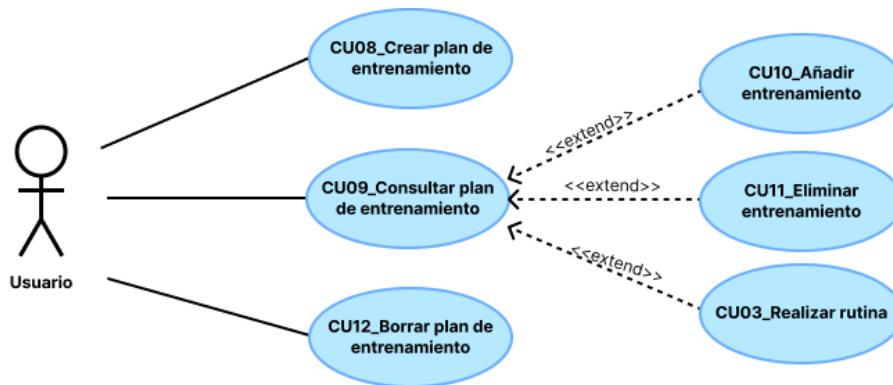


Figura 10. Gestión de planes de entrenamiento

En la Tabla 4, se describe el comportamiento del caso de uso CU08_Crear plan de entrenamiento. Esta funcionalidad permite al usuario crear un plan de entrenamiento según sus necesidades.

Tabla 4. CU08_Crear plan de entrenamiento

Descripción	El usuario puede crear un plan de entrenamiento introduciendo su nombre y estableciendo la visibilidad.	
Precondición	El usuario ha iniciado sesión y ha accedido a la lista de planes de entrenamiento.	
Secuencia normal	Paso	Acción
	1	El usuario introduce un nombre para el plan de entrenamiento.
	2	El usuario selecciona la visibilidad del plan de entrenamiento (pública o privada).
Postcondición	Se crea un nuevo plan de entrenamiento con el nombre y visibilidad especificados.	
Excepciones	El usuario no ha proporcionado un nombre para el plan de entrenamiento.	

En la Tabla 5, se describe el comportamiento del caso de uso CU09_Consultar plan de entrenamiento. Aquí el usuario puede consultar un plan de entrenamiento y realizar las siguientes acciones: borrar el plan de entrenamiento y añadir, eliminar y realizar entrenamientos.

Tabla 5. CU09_Consultar plan de entrenamiento

Descripción	Permite al usuario consultar los entrenamientos en un calendario, añadir entrenamientos, eliminar entrenamientos y realizar las rutinas.	
Precondición	El usuario ha iniciado sesión y ha accedido al plan de entrenamiento deseado.	
Secuencia normal	Paso	Acción
	1	El usuario consulta el calendario para ver los entrenamientos programados.
	2	El usuario puede añadir nuevos entrenamientos al plan seleccionando una fecha y hora específicas.
	3	El usuario puede eliminar entrenamientos existentes del plan.
	4	El usuario puede seleccionar un entrenamiento para realizar la rutina asociada.
	5	El usuario puede borrar el plan de entrenamiento.
Postcondición	Los cambios realizados en el plan de entrenamiento (añadir, eliminar, realizar rutinas, borrar) se guardan.	
Excepciones	Sin excepciones	

Diagrama: Gestión social

En la Figura 11 encontramos el diagrama de casos de uso para la característica Gestión social. Podemos observar que hay dos casos de uso principales CU13_Crear grupo de ejercicio que extiende el caso de uso CU14_Añadir usuarios. También vemos el caso de uso CU15_Acceder al grupo que es extendido por los siguientes casos de uso CU14_Añadir usuarios, CU16_Compartir que incluye CU17_Seleccionar rutina, CU18_Compartir plan de entrenamiento que incluye CU19_Seleccionar plan de entrenamiento, CU20_Enviar mensaje, CU21_Abandonar grupo y CU22_Eliminar grupo.

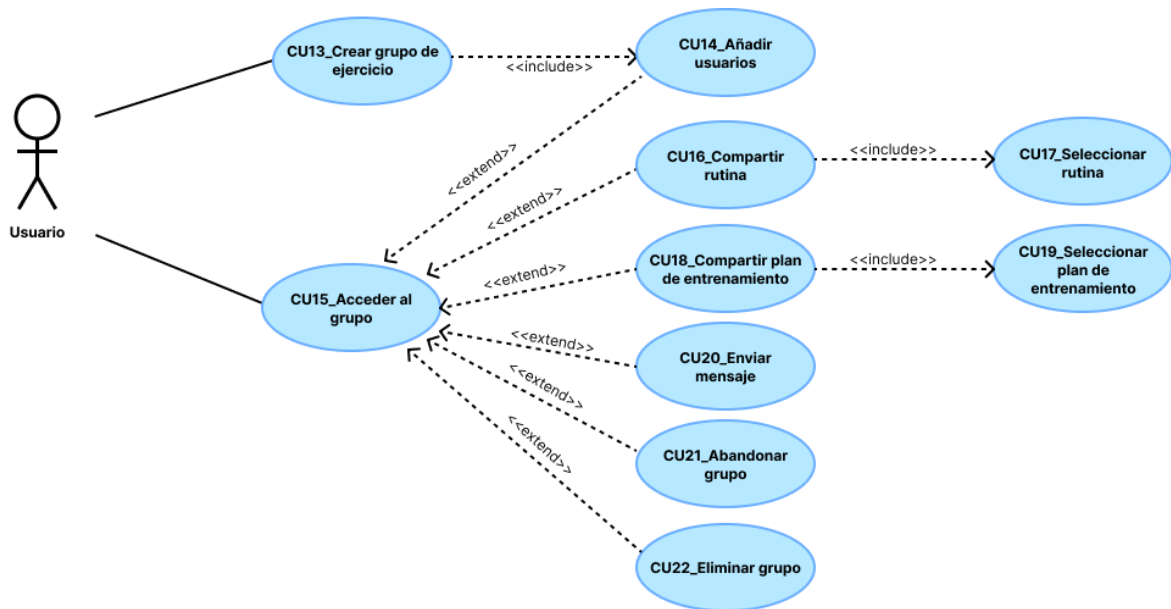


Figura 11. Gestión social

A continuación, encontramos la Tabla 6 que describe el comportamiento del caso de uso CU13_Crear grupo de ejercicio. Esta funcionalidad permite al usuario crear un grupo de ejercicio y añadir a los usuarios.

Tabla 6. CU13_Crear grupo de ejercicio

Descripción	Permite al usuario crear un grupo de ejercicio, introduciendo el nombre del grupo y añadiendo usuarios al mismo.	
Precondición	El usuario ha iniciado sesión y ha accedido a la pantalla de creación de grupos de ejercicio.	
Secuencia normal	Paso	Acción
	1	El usuario introduce un nombre para el grupo de ejercicio.
	2	El usuario selecciona y añade usuarios al grupo.
Postcondición	Se crea un nuevo grupo de ejercicio con el nombre y usuarios especificados.	
Excepciones	El usuario no ha proporcionado un nombre para el grupo de ejercicio.	

En la Tabla 7, se describe el comportamiento del caso de uso CU14_Añadir usuarios. Aquí el usuario puede seleccionar los usuarios que quiere añadir al grupo.

Tabla 7. CU14_Añadir usuarios

Descripción	Permite al usuario añadir usuarios a un grupo de ejercicio existente.	
Precondición	El usuario ha iniciado sesión y ha accedido a la pantalla de gestión de grupos de ejercicio.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona el grupo de ejercicio al que desea añadir usuarios.
	2	El usuario selecciona y añade usuarios al grupo de ejercicio.
Postcondición	Los usuarios seleccionados se añaden al grupo de ejercicio.	
Excepciones	El usuario no ha seleccionado usuarios válidos para añadir al grupo.	

Gestión de usuario

En la Figura 12 encontramos el diagrama de casos de uso para la característica Gestión de usuario. Podemos observar que tiene cuatro casos de uso principales CU23_Ver perfil de usuario que es extendido por el caso de uso CU24_Editar perfil. Y los casos de uso CU25_Seguimiento entrenamiento, CU26_Calendario de entrenamiento y CU27_Consultar objetivos.

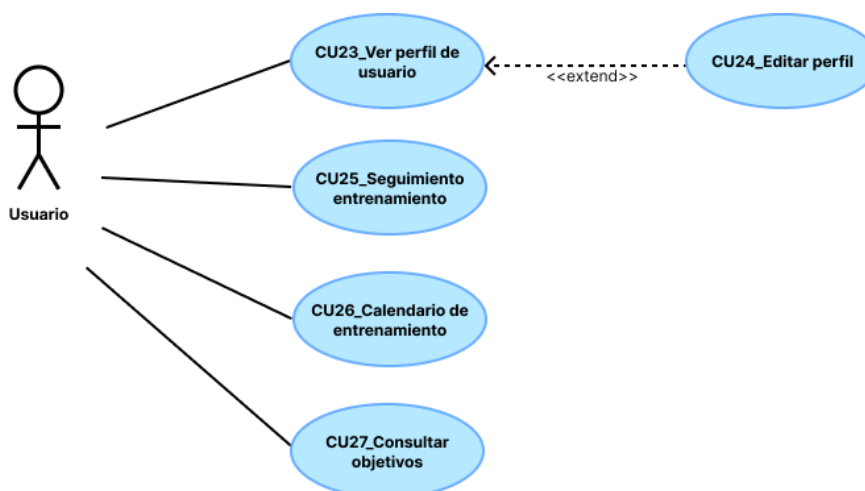


Figura 12. Gestión de usuario

En la Tabla 8 se describe el comportamiento del caso de uso CU23_Ver perfil de usuario. Esta funcionalidad permite al usuario ver la información de su perfil y seleccionar la opción para editar la información de su perfil.

Tabla 8. CU23_Ver perfil de usuario

Descripción	Permite al usuario ver la información de su perfil y editar su perfil.	
Precondición	El usuario ha iniciado sesión y está en su perfil.	
Secuencia normal	Paso	Acción
	1	El usuario visualiza la información de su perfil.
	2	El usuario selecciona la opción de editar perfil.
Postcondición	La información del perfil se actualiza con los cambios realizados por el usuario.	
Excepciones	El usuario no ha iniciado sesión.	

En la Tabla 9 se describe el comportamiento del caso de uso CU24_Editar perfil. Aquí los usuarios pueden modificar la información de su perfil. Aquí los usuarios pueden introducir su edad, peso, altura, sexo y nombre de usuario. Gracias a esta información se puede aproximar mejor los resultados que se obtienen al realizar los entrenamientos a la realidad.

Tabla 9. CU24_Editar perfil

Descripción	Permite al usuario editar su perfil, incluyendo cambios en la edad, peso, altura, sexo y nombre de usuario.	
Precondición	El usuario ha iniciado sesión y está en su perfil.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción de editar perfil.
	2	El sistema muestra un formulario de edición de perfil con los campos existentes.
	3	El usuario realiza los cambios deseados en los campos del formulario.
	4	El usuario guarda los cambios al pulsar el botón de guardar.
Postcondición	La información del perfil se actualiza con los cambios realizados por el usuario.	
Excepciones	El usuario no ha iniciado sesión.	

3.2. Identificación y análisis de soluciones posibles

En este apartado vamos a analizar las posibles soluciones al problema planteado explicando sus ventajas y desventajas.

Aplicaciones específicas para cada plataforma

Una posible solución sería crear aplicaciones específicas para cada tipo de plataforma, como Android, iOS, Windows, macOS, entre otras. Esta opción presenta varias ventajas significativas. En primer lugar, al desarrollar aplicaciones nativas para cada plataforma, se garantiza un rendimiento óptimo y una experiencia de usuario fluida en cada dispositivo. Además, con el uso de lenguajes multiplataforma y *frameworks* adecuados, es posible compartir la lógica de la aplicación entre las distintas plataformas, lo que simplifica el desarrollo y el mantenimiento del código.

Sin embargo, es importante tener en cuenta que esta solución también conlleva desventajas. El principal inconveniente radica en el costo y el tiempo de desarrollo asociados. Al tener que crear productos independientes para cada plataforma, se requiere una mayor inversión de recursos y un mayor esfuerzo de desarrollo. Además, el proceso de diseño de la interfaz debe adaptarse a los estándares y pautas específicas de cada plataforma, lo que implica trabajo adicional.

Aplicación web

Otra posible solución sería crear una aplicación web. Esta opción presenta ventajas significativas, como la capacidad de ser accesible desde cualquier dispositivo que cuente con un navegador web, lo que permitiría llegar a un amplio número de usuarios. Además, al tratarse de una aplicación basada en la web, no sería necesario desarrollar versiones específicas para cada plataforma, lo que simplificaría el proceso de desarrollo y mantenimiento.

Sin embargo, es importante tener en cuenta una desventaja importante de esta solución: la limitación de su funcionamiento sin conexión a internet. Implementar la capacidad de utilizar la aplicación sin conexión puede resultar más costoso y complejo, ya que requeriría de la creación de un servidor local.

En resumen, optar por una aplicación web ofrece la ventaja de la accesibilidad multiplataforma, pero se debe tener en cuenta el requerimiento de funcionamiento offline y los costos asociados a su implementación.

3.3. Solución propuesta

A continuación, se presenta la solución elegida para el desarrollo del TFG. La solución seleccionada consiste en la creación de una aplicación nativa para dispositivos Android. Esta elección se basa en la búsqueda de un rendimiento óptimo y en la capacidad de desarrollar todas las funcionalidades deseadas de manera eficiente.

Una de las características clave de esta solución es la posibilidad de implementar un sistema que permita el uso de la aplicación tanto en línea como sin conexión a internet. Esto ofrece una mayor flexibilidad para los usuarios, quienes podrán acceder y utilizar la aplicación en cualquier momento y lugar.

3.4. Plan de Trabajo

Una vez presentada la solución elegida, es importante establecer un plan de trabajo para asegurar una ejecución eficiente y ordenada del proyecto. A continuación, se detallan los aspectos clave del plan de trabajo.

3.4.1. Metodología

Para el desarrollo de la aplicación, se ha elegido utilizar la metodología ágil Scrum (Ken Schwaber & Jeff Sutherland, 2020). Scrum es ampliamente reconocida y utilizada en el ámbito del desarrollo de software debido a su enfoque iterativo e incremental, que permite una entrega continua de valor al cliente.

Esta metodología suele ser ideal para equipos de hasta 10 personas. Además, suelen ser equipos *cross-functional*⁷, por lo que todos sus miembros tienen el conocimiento necesario para realizar cualquier tarea y de forma interna deciden quien realiza qué, cuándo y cómo cada tarea.

Conceptos previos

Esta metodología utiliza los siguientes conceptos que se van a definir a continuación:

Unidad de trabajo (UT)

Es una pieza de trabajo específica y manejable. Representa un requisito funcional o una funcionalidad deseada. Debe ser lo suficientemente pequeña para sea completada en un periodo de tiempo corto, generalmente dentro de una iteración o *sprint*.

Backlog

Aquí se ordenan todas las unidades de trabajo que hay que realizar en el producto que se está desarrollando, como funcionalidades de la aplicación, creación del modelo de datos, etc.

⁷ Se dice de un equipo en el que todos los integrantes tienen conocimiento para realizar cualquier tarea.



Sprint

Son el corazón de la metodología Scrum, es donde se desarrollan las ideas y se convierten en valor para el producto. Tienen una duración fija de 2 a 4 semanas. Un *sprint* empieza nada más terminar el anterior. Además, tiene su propio *backlog*, que se suele llamar *Sprint Backlog*, en el que están las unidades de trabajo que se desarrollaran en ese *sprint*.

Durante el *sprint* no se producen cambios que puedan hacer peligrar el objetivo que se quiere alcanzar, la calidad del trabajo no disminuye, el *backlog* se puede refinar según sea necesario y el alcance del *sprint* se puede renegociar con el *Product Owner* según se va recabando más información.

Incremento

Es un paso concreto hacia el producto final. Cada incremento es una suma de los incrementos anteriores, verificando que todos funcionan correctamente al mismo tiempo.

Definición de Done

Es la descripción formal del estado que tiene que cumplir cada incremento obtenido cuando se mide la calidad requerida para el producto. Ofrece transparencia al proveer a todo el mundo del conocimiento necesario para detectar que parte del trabajo fue completada por parte del incremento.

Si un ítem del *backlog* no cumple esta definición no puede darse por terminado y no debe presentarse en la revisión del *sprint*, sino que debe devolverse al *backlog* para desarrollarlo en el futuro.

Roles

Cada equipo de trabajo, también llamado *Scrum Team*, está formado por los siguientes roles:

Desarrollador

Los desarrolladores son las personas del equipo Scrum que tienen como tarea crear cualquier aspecto funcional de cada *sprint*. Realizan las siguientes funciones: crean un plan para cada *sprint*, incrementan la calidad al ajustarse a la Definición de Done, adaptan su plan cada día para alcanzar la meta del *sprint* y reparten las responsabilidades entre todos ellos.

Product Owner (PO)

Es el responsable de maximizar el valor del producto y de dirigir efectivamente el *backlog* del producto en un equipo Scrum. Realiza las siguientes tareas: debe comunicar efectivamente el objetivo del producto, crear los ítems del producto en el *backlog*, ordenar los ítems y asegurarse de que el *backlog* es transparente, visible y entendible.

Scrum Master (SM)

Es el responsable de establecer y garantizar la correcta implementación de Scrum dentro de un equipo. Su función principal es asegurarse de que todos los miembros del equipo comprendan tanto la teoría como la práctica de Scrum.

El *Scrum Master* ayuda al equipo de muchas formas: enseñando a los miembros en autogestión y en su capacidad para realizar diferentes funciones dentro del equipo, ayudando al equipo a centrarse en crear incrementos de alto valor que sigan la Definición de Done y eliminando las causas que no permitan avanzar al equipo.

Reuniones

Por último, esta metodología recomienda realizar las siguientes reuniones durante el *sprint*.

Reunión Diaria

También llamada *Daily Scrum*. Esta reunión es realizada por los desarrolladores cada día antes de comenzar a desarrollar y suele durar 15 minutos. El propósito de esta reunión es revisar el progreso del *sprint*, adaptar el *Sprint Backlog* según las necesidades y ajustar el trabajo a realizar.

Reunión de revisión del Sprint

También llamada *Sprint Review*. Esta reunión se realiza al final de cada *sprint* y suele durar un máximo de 4 horas para *sprint* de un mes de duración. Para *sprints* más cortos, la duración es menor. El propósito de esta reunión es revisar el resultado del *sprint*, determinar futuras adaptaciones, presentar el resultado a los *stakeholders*⁸ y se discute y discutir el progreso hacia el producto final.

Reunión de Retrospectiva

También llamada *Sprint Retrospective*, esta reunión se realiza al final de cada *sprint* y suele durar como máximo 3 horas para *sprint* de un mes. Para *sprints* más cortos, la reunión es más corta. El propósito de esta reunión es revisar como se ha aplicado la metodología durante el *sprint* y se buscar problemas que se puedan resolver o mejoras que se puedan realizar para mejorar el flujo de trabajo. Todos estos cambios deben aplicarse en el siguiente *sprint*.

Además, vamos a combinar esta metodología con el enfoque Kanban para tener una mayor visibilidad del flujo de trabajo. Con esta metodología hacemos uso de un tablero Kanban en el que se sitúan todas las unidades de trabajo y van pasando por todos los estados hasta terminarlas. Vamos a usar un tablero Kanban orientado al desarrollo de software que está formado por los siguientes estados:

- Registrar UT.
- Esperar Prioridad.
- Especificar Requisitos.
- Esperar incluir en Sprint.
- Programar.
- Aplicar Pruebas de Aceptación.
- Terminada.

3.4.2. Aplicación de la metodología

Asignación de los Roles

En primer lugar, vamos a empezar hablando de cómo se van a asignar los tres roles *Scrum Master*, *Product Owner* y desarrollador. Al solo tener un integrante en el equipo los tres roles van a ser desempeñados por una sola persona.

Duración de los sprint

A continuación, vamos a detallar como se van a dividir el desarrollo del producto y la duración de los *sprints*.

⁸ Son las partes interesadas o involucradas en un proyecto o proceso, como clientes, usuarios, patrocinadores y equipos de desarrollo, que tienen un interés o influencia en el resultado final.



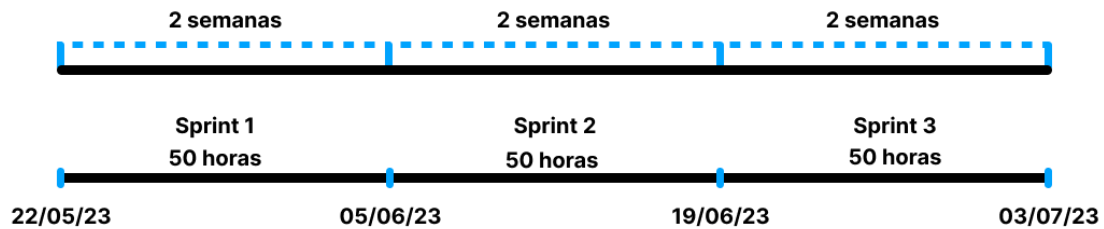


Figura 13. Organización de los sprints

Al tener una duración total de 300 horas para desarrollar el TFG, se ha decidido repartir la mitad de las horas a realizar la memoria y la otra mitad a desarrollar la aplicación.

De esta manera, nos quedan 150 horas para desarrollar la aplicación, por lo que se ha decidido realizar tres *sprints* de dos semanas de duración. Dedicando 50 horas en cada sprint. Lo que nos permite distribuir las horas en 5 horas diarias sin contar los fines de semana.

Definición de Done

Seguidamente, como vamos a usar un tablero Kanban vamos a aclarar cuál va a ser la Definición de Done en cada uno de estos estados.

Registrar UT

Este es el primer estado por el que pasaran las UT una vez han sido creadas. Aquí tenemos que verificar que no está repetida y que no se solapa con alguna otra. Además, se realizará una estimación preliminar de las Horas Ideales de Programación⁹ que nos llevará desarrollar esa UT.

Esperar Prioridad

En este estado se deberían de mantener ordenadas todas las UT, con cada nueva UT que llegue a este estado se le debe asignar un número de orden.

Este estado es en el que se mantienen las UT hasta que el *Product Owner* decide que hay que comenzar a prepararlas para incluirlas en el *sprint* en el que se desarrollarán. En este estado se deben de mantener ordenadas todas las UT, a cada nueva UT que llegue a este estado se le debe asignar un número de orden.

Especificar Requisitos

En este estado se deben elaborar los *mockups*¹⁰ de la interfaz con la suficiente información para programar la UT. También hay que establecer las Pruebas de Aceptación¹¹, PA en adelante, que puedan estar asociadas a los cambios que realice esta UT. Por último, si es necesario, se debe reestimar la UT. Ya que ahora se cuenta con más información para tener una estimación más cercana al coste de desarrollarla.

Esperar incluir en sprint

Este estado es en el que se quedan las UT preparadas hasta que se van a implementar en el *sprint*. Estas UT se moverán a Programar una vez estén asignadas a un *sprint* y este comience. Antes de pasar la UT a Programar debe comprobarse que el esfuerzo de añadir esa UT al *sprint* no es mayor a la capacidad del equipo.

⁹ Son las horas que se esperan invertir en la programación de una funcionalidad o característica.

¹⁰ Representación visual estática o interactiva de la interfaz de usuario de una aplicación o sistema.

¹¹ Evaluación realizada para determinar si un sistema o aplicación cumple con los criterios de aceptación definidos por los usuarios finales, clientes o stakeholders.

Programar

En este estado se desarrolla la lógica, la interfaz y la posible persistencia, siguiendo las pautas de organización de código acordadas por el equipo. Si es necesario, se puede refactorizar el código o aplicar patrones.

Una vez desarrollado se debe comprobar las PA establecidas y verificar que la UT pasa las pruebas correctamente. Cuando se ha comprobado que pasa las pruebas se puede pasar al estado Aplicar Pruebas de Aceptación.

Aplicar Pruebas de Aceptación

Aquí se vuelven a pasar las PA para comprobar que la UT funciona correctamente y si no se encuentran fallos se pasa al estado Terminado. En el caso de encontrar algún error la UT se devuelve al estado Programar para corregirlo.

3.5. Presupuesto

En este apartado se va a crear un presupuesto estimado que refleje los costos asociados a la realización de la aplicación en un entorno de trabajo real.

En primer lugar, hay que estimar el costo de los recursos hardware que se van a utilizar para desarrollar el producto. A continuación, iremos explicando para qué se va a utilizar cada recurso y el costo que tendría.

Para poder desarrollar la aplicación y realizar pruebas, necesitamos un ordenador de sobremesa, teclado, ratón y monitor. El costo podría rondar los 1000€. En el ordenador se instalarían todos los programas necesarios para el desarrollo. Teniendo en cuenta que se pueden utilizar tecnologías de código abierto no sería necesario gastar dinero en software de desarrollo.

Además, también hay que tener en cuenta el coste del servidor de la base de datos. Como vamos a utilizar Firebase, que es una plataforma que escala en base a la demanda de nuestra aplicación, al principio, cuando la base de usuarios sea pequeña, se podría utilizar el plan gratuito.

Respecto a las horas-hombre, hay que tener en cuenta no solo el tiempo invertido en el desarrollo, sino que también las horas dedicadas a la especificación de requisitos, el análisis de otras aplicaciones, etc. Considerando 150 horas para el desarrollo y 50 horas a las demás tareas, tendríamos un total de 200 horas de trabajo, lo que equivaldría a aproximadamente 3,5 semanas a tiempo completo.

Si el desarrollador cobra un sueldo de 1500€ y el coste del material de desarrollo nos cuesta 1000€. El costo total del proyecto en total sería de 2500€ para tener un producto listo para ser lanzado.

A continuación, se muestra en la Tabla 10 un resumen del presupuesto:

Tabla 10. Resumen del presupuesto

Recurso	Costo estimado
Hardware (ordenador, teclado, ratón, monitor)	1000€
Software de desarrollo	0€ (software de código abierto)
Servidor de base de datos (Firebase)	0€ (plan gratuito inicialmente)
Desarrollo	1500€
Total	2500€

4. Diseño de la solución

En este apartado, vamos a definir y explicar la arquitectura que adoptaremos para nuestra aplicación, así como el diseño de las clases que interactuarán durante su ejecución. También discutiremos las tecnologías que utilizaremos para su desarrollo.

4.1. Arquitectura del Sistema

En este apartado, vamos a explicar la arquitectura de nuestra aplicación y los subsistemas que permiten su funcionamiento. A continuación, en la Figura 14, mostraremos una representación gráfica de dicha arquitectura.

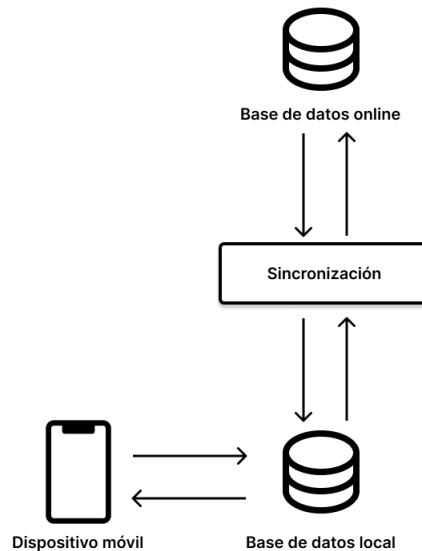


Figura 14. Arquitectura de la aplicación

Para nuestra aplicación, hemos decidido utilizar una arquitectura Cliente-Servidor con sincronización de datos. Los componentes desempeñan las siguientes funciones:

Dispositivo móvil

- Actúa como cliente.
- Proporciona al usuario la interfaz visual para interactuar con el sistema.
- Procesa las operaciones realizadas por el usuario.
- Se comunica con la base de datos local.

Base de datos local

- Almacena la información del usuario de manera local.
- Permite al usuario acceder a la información incluso sin acceso a Internet.
- Se comunica con la base de datos online para sincronizar la información local con la de la base de datos online cuando haya conexión disponible.

Sincronización

- Monitoriza los cambios realizados en la base de datos local.
- Proporciona los medios para transferir los cambios detectados desde la base de datos local a la base de datos en línea.
- Se encarga de aplicar los cambios de una manera adecuada y segura.
- En caso de conflictos durante la sincronización, como cambios simultáneos en ambos entornos que afecten a los mismos datos, el componente aplica un mecanismo de resolución de conflictos en el que selecciona los cambios más recientes.

Base de datos online

- Mantiene la información de todos los usuarios de manera centralizada.
- Permite que los usuarios acceder a su información desde cualquier dispositivo móvil.
- Facilita el intercambio de información entre usuarios al tener una base de datos centralizada.

Esta arquitectura nos permite brindar una experiencia fluida al usuario, ya que pueden acceder a su información desde cualquier dispositivo y también utilizar la aplicación en modo offline. Además, la sincronización de datos entre la base de datos local y la base de datos online garantiza que los datos estén actualizados y disponibles en todo momento.

4.2. Tecnología Utilizada

En este apartado se van a comentar las herramientas, tecnologías y *frameworks* que se van a utilizar durante el desarrollo del proyecto, el conocimiento personal y coste de aprendizaje de estas.

4.2.1. Herramientas

Respecto a las herramientas que vamos a utilizar son las siguientes:

Visual Studio Code

Como entorno de desarrollo vamos a utilizar Visual Studio Code¹². Es actualmente uno de los IDE¹³ más populares gracias a la gran cantidad de opciones que tiene. A continuación, se nombran algunas de sus características más destacadas:

- Es un IDE de código abierto, ligero y personalizable.
- Tiene una gran cantidad de extensiones.
- Permite realizar control de versiones con Git.

Esta herramienta ya la hemos utilizado anteriormente por lo que no ha sido necesario invertir tiempo en su aprendizaje. Es muy sencillo de utilizar y tiene una documentación clara.

Github

Para mantener el repositorio en la nube se ha utilizado GitHub¹⁴. Es una plataforma en línea que permite a los desarrolladores alojar, colaborar y gestionar el control de versiones de proyectos de software, proporcionando herramientas para el seguimiento de cambios, la gestión de problemas y la colaboración en equipo.

Esta herramienta ya la hemos utilizado anteriormente por lo que no ha sido necesario invertir tiempo en su aprendizaje. Es sencillo de utilizar y rápido de aprender.

¹² <https://code.visualstudio.com/>

¹³ Un IDE (Entorno de Desarrollo Integrado) es una herramienta de software que proporciona un conjunto completo de funcionalidades para el desarrollo de aplicaciones en un solo entorno de trabajo.

¹⁴ <https://github.com/>



4.2.2. Tecnologías y frameworks

Respecto a las tecnologías y *frameworks* que vamos a utilizar son los siguientes:

Firestore

Para la base de datos online vamos a utilizar Firestore¹⁵. Es un servicio de Google que ofrece distintos servicios para desarrollar aplicaciones en la nube. En nuestro caso vamos a utilizar dos de esos servicios:

Cloud Firestore

Es una base de datos NoSQL¹⁶ flexible, escalable y en la nube que permite almacenar y sincronizar datos para el desarrollo tanto del lado del cliente como del servidor.

Authentication

Proporciona servicios para autenticar a los usuarios en la aplicación, permite utilizar distintos proveedores para iniciar sesión como Google o Facebook. Pero nosotros solo vamos a utilizar el inicio con correo y contraseña.

Esta tecnología ya la hemos utilizado con anterioridad por lo que no se ha tenido que invertir tiempo en su aprendizaje, además es una tecnología sencilla y rápida de aprender.

Realm

Como base de datos local para almacenar la información del usuario en el dispositivo vamos a utilizar Realm¹⁷. Es una tecnología que tiene las siguientes características:

- Es una base de datos NoSQL orientada a objetos,
- Se integra directamente en la aplicación
- Se ejecuta en el dispositivo del usuario
- Tiene un enfoque *offline first* que significa que está diseñada para funcionar en entornos sin conexión a internet.

Esta tecnología no la habíamos utilizado con anterioridad por lo que ha sido necesario ir aprendiendo a utilizarla durante el desarrollo del proyecto. Tiene una curva de aprendizaje sencilla por lo que no nos ha requerido demasiado tiempo, además tiene una documentación muy detallada y clara.

JavaScript

Para desarrollar la lógica de la aplicación vamos a utilizar JavaScript¹⁸. Tiene las siguientes características:

- Es un lenguaje de programación asíncrono y orientado a objetos,
- Cuenta con una gran cantidad de bibliotecas y *frameworks* que facilitan su desarrollo.

Este lenguaje ya lo habíamos utilizado con anterioridad por lo que no se ha tenido que invertir tiempo en su aprendizaje, aunque tiene una curva de aprendizaje rápida.

15 <https://firebase.google.com/?hl=es>

16 Categoría de bases de datos que no utilizan el modelo de tablas relacionales. Se basan en diferentes modelos de datos, como documentos, grafos o clave-valor.

17 <https://www.mongodb.com/docs/realm/>

18 <https://mzl.la/43hthdf>

React Native

Para desarrollar la interfaz de usuario de la aplicación vamos a utilizar React Native¹⁹. Tiene las siguientes características:

- Es un *framework* de JavaScript que permite construir aplicaciones para iOS y Android compartiendo código común.
- Ofrece un desarrollo eficiente, rendimiento nativo y acceso a características del dispositivo.

Esta tecnología la hemos utilizado con anterioridad por lo que no ha sido necesario dedicar tiempo a su aprendizaje. Es un lenguaje fácil de aprender y que permite desarrollar interfaces rápidamente.

Expo

Expo²⁰ es una plataforma de desarrollo de aplicaciones móviles que simplifica el proceso de creación de aplicaciones nativas utilizando JavaScript y React Native. Tiene las siguientes características:

- Facilidad de desarrollo.
- Amplia gama de bibliotecas y componentes.
- Actualizaciones en vivo, es decir, cuando se cambia el código de la aplicación los cambios se ven reflejados en tiempo real en el dispositivo.
- Servicios en la nube.

Esta tecnología la hemos utilizado con anterioridad por lo que no ha sido necesario dedicar tiempo a su aprendizaje. Es un fácil de aprender y tiene una documentación muy clara con ejemplos paso a paso sobre cualquier aspecto de la tecnología.

Tailwind CSS

Es un framework de utilidades CSS²¹ que permite construir interfaces de usuario de manera rápida y eficiente. La principal característica de Tailwind CSS²² es su enfoque en clases CSS utilitarias. En lugar de definir estilos personalizados en archivos CSS separados, Tailwind CSS proporciona una amplia gama de clases predefinidas que se pueden aplicar directamente a los componentes de React Native.

Es una tecnología que hemos utilizado con anterioridad por lo que no ha sido necesario dedicar tiempo a su aprendizaje. Es rápida y sencilla de aprender y tiene una documentación muy clara.

¹⁹ <https://bit.ly/3O57GQQ>

²⁰ <https://docs.expo.dev/>

²¹ CSS (*Cascading Style Sheets*) es un lenguaje de diseño utilizado para definir la apariencia y el estilo visual de los elementos de una página web.

²² <https://tailwindcss.com/docs/installation>



4.3. Diseño Detallado

En este apartado se va a detallar cuales van a ser las clases que se utilizaran en la aplicación y la relación que hay entre ellas.

4.3.1. Clases de la aplicación

A continuación, en la Figura 15, se puede encontrar el diagrama de clases de nuestra aplicación:

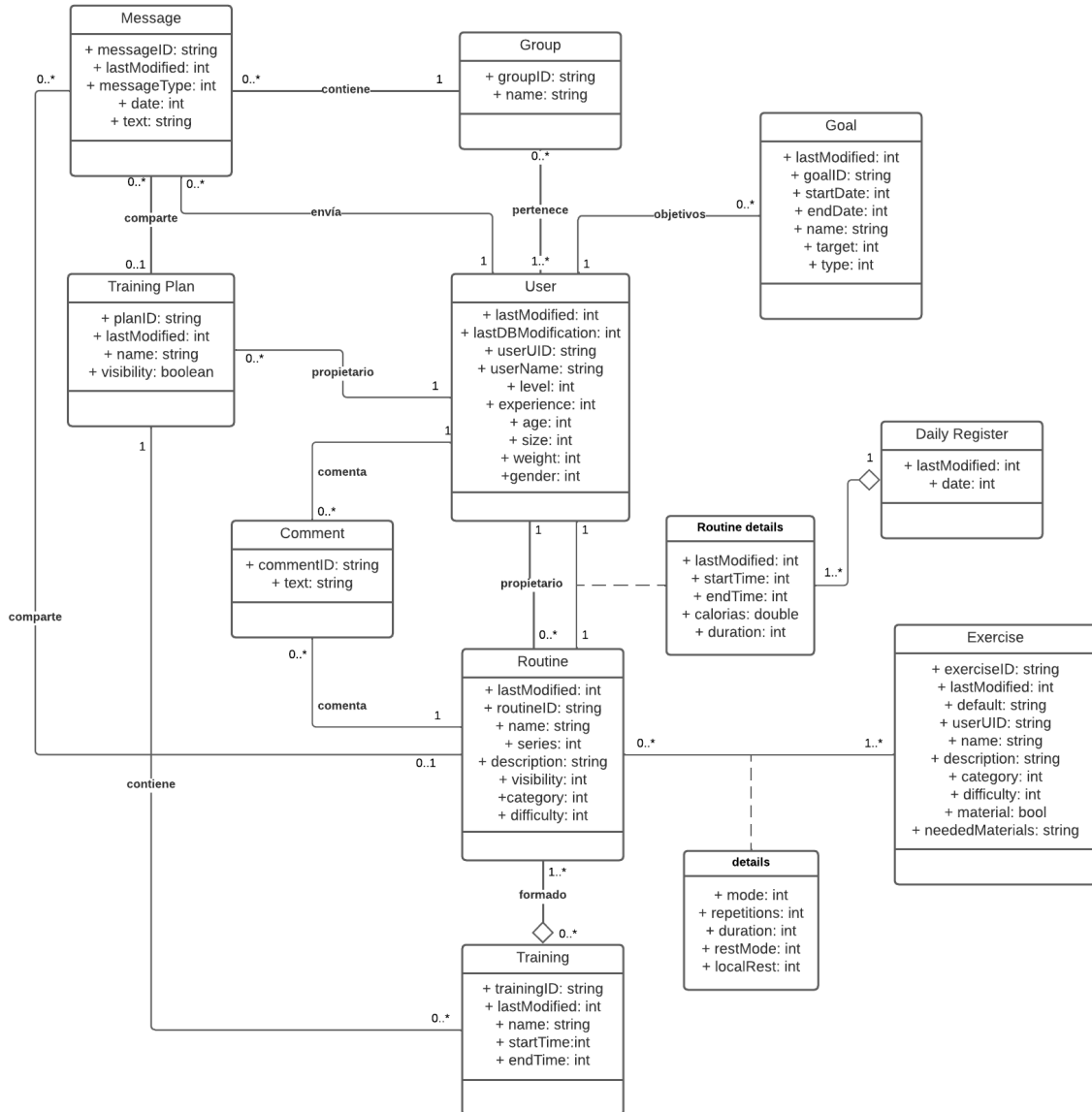


Figura 15. Diagrama de clases

Usuario

Esta es la clase central de la aplicación y contiene toda la información del usuario. Tiene los siguientes atributos:

- ***lastModified(int)***: Contiene una marca de tiempo de la fecha en la que se modificó.
- ***userID(string)***: Este es el identificador único del usuario.
- ***userName(string)***: Es el nombre del usuario.
- ***lastDBModification(int)***: Mantiene una marca de tiempo de la última vez que el usuario ha modificado la base de datos.
- ***level(int)***: Es el nivel del usuario.
- ***experience(int)***: Es la experiencia del usuario.
- ***age(int)***: Es la edad del usuario.
- ***size(int)***: Es la altura del usuario en centímetros.
- ***weight(int)***: Es el peso del usuario en kilogramos.

Tiene las siguientes relaciones:

- Un usuario puede crear cero o muchas Rutinas.
- Un usuario puede tener cero o muchos Objetivos.
- Un usuario puede estar en cero o muchos Grupos.
- Un usuario puede enviar cero o muchos Mensajes.
- Un usuario puede realizar cero o muchos Comentarios.
- Un usuario puede tener cero o muchos Planes de Entrenamiento.

Rutina

Esta clase contiene la información sobre una rutina y como realizarla. Tiene los siguientes atributos:

- ***lastModified(int)***: Contiene una marca de tiempo de la fecha en la que se modificó.
- ***routineID(string)***: Es el identificador único de la rutina.
- ***name(string)***: Es el nombre de la rutina.
- ***series(int)***: Es el número de veces que se va a realizar la rutina para que se considere completa.
- ***description(string)***: Es la descripción de la rutina.
- ***visibility(int)***: Representa si una rutina es pública o privada.
- ***category(int)***: Representa la categoría a la que pertenece la rutina.
- ***difficulty(int)***: Representa la dificultad de la rutina.

Tiene las siguientes relaciones:

- Una rutina pertenece a un Usuario.
- Una rutina puede tener cero o muchos Comentarios.
- Una rutina se puede compartir en cero o muchos Mensajes.
- Una rutina tiene uno o muchos Ejercicios.
- Una rutina puede estar incluida en cero o muchos Entrenamientos.



Ejercicio

Esta clase contiene la información sobre cómo se realiza un ejercicio. Tiene los siguientes atributos:

- ***lastModified(int)***: Contiene una marca de tiempo de la fecha en la que se modificó.
- ***exercisID(string)***: Es el identificador único del ejercicio.
- ***default(int)***: Representa si un ejercicio es predeterminado o no.
- ***name(string)***: Es el nombre del ejercicio.
- ***description(string)***: Es la descripción del ejercicio.
- ***category(int)***: Representa la categoría a la que pertenece un ejercicio.
- ***difficulty(int)***: Representa la dificultad del ejercicio.

Tiene la siguiente relación, un ejercicio puede estar en cero o muchas Rutinas.

Objetivo

Contiene la información de un objetivo. Tiene los siguientes atributos:

- ***lastModified(int)***: Contiene una marca de tiempo de la fecha en la que se modificó.
- ***goalID(string)***: Es el identificador único del objetivo.
- ***startDate(int)***: Es la fecha de inicio del objetivo.
- ***endDate(int)***: Es la fecha de finalización del objetivo.
- ***name(string)***: Es el nombre del objetivo.
- ***target(int)***: Representa el valor que se quiere conseguir.
- ***type(int)***: Representa el tipo de objetivo.

Tiene la siguiente relación, un objetivo pertenece a un Usuario.

Plan de entrenamiento

Contiene la información de un plan de entrenamiento. Tiene los siguientes atributos:

- ***lastModified(int)***: Contiene una marca de tiempo de la fecha en la que se modificó.
- ***planID(string)***: Es el identificador único del plan de entrenamiento.
- ***name(string)***: Es el nombre del plan de entrenamiento.
- ***visibility(boolean)***: Es la visibilidad del plan de entrenamiento.

Tiene las siguientes relaciones:

- Un plan de entrenamiento pertenece a un Usuario.
- Un plan de entrenamiento tiene cero o muchos Entrenamientos.
- Un plan de entrenamiento puede compartirse en cero o muchos Mensajes.

Entrenamiento

Contiene la información de un entrenamiento. Tiene los siguientes atributos:

- ***lastModified(int)***: Contiene una marca de tiempo de la fecha en la que se modificó.
- ***trainingID(string)***: Es el identificador único del entrenamiento.
- ***name(string)***: Es el nombre del entrenamiento.
- ***startTime(int)***: Es la fecha de inicio del entrenamiento.
- ***endTime(int)***: Es la fecha de finalización del entrenamiento.

Comentario

Contiene la información de un comentario. Tiene los siguientes atributos:

- **commentID(string)**: Es el identificador único del comentario.
- **text(string)**: Es el contenido del comentario.

Tiene las siguientes relaciones:

- Un comentario pertenece a un Usuario.
- Un comentario pertenece a una Rutina.

Grupo

Contiene la información de un grupo de entrenamiento. Tiene los siguientes atributos:

- **groupID(string)**: Es el identificador único del grupo.
- **name(string)**: Es el nombre del grupo.

Tiene las siguientes relaciones:

- Un grupo contiene cero o muchos Mensajes.
- Un grupo pertenece a uno o muchos Usuarios.

Mensaje

Contiene la información de un mensaje. Tiene los siguientes atributos:

- **lastModified(int)**: Contiene una marca de tiempo de la fecha en la que se modificó.
- **messageID(string)**: Es el identificador único del mensaje.
- **messageType(int)**: Representa el tipo de mensaje.
- **date(int)**: Es la fecha de envío del mensaje.
- **text(string)**: Es el texto del mensaje.

Tiene las siguientes relaciones:

- Un mensaje comparte cero o una Rutina.
- Un mensaje comparte cero o un Plan de Entrenamiento.
- Un mensaje pertenece a un Grupo.
- Un mensaje pertenece a un Usuario.



5. Desarrollo de la solución propuesta

En este apartado se va a explicar cómo se ha ido desarrollando la solución propuesta presentando los problemas y dificultades que se han ido encontrando, las decisiones que se han ido tomando y las particularidades de la solución final.

5.1. Primer Sprint

Repositorio de la solución

Antes de empezar con el desarrollo de la aplicación, es necesario preparar el repositorio en el que se van a subir las distintas versiones que se tengan como resultado de añadir funcionalidades al proyecto.

Vamos a subir el proyecto inicial a GitHub. Una vez subido ya podemos empezar a desarrollar la aplicación y cuando se complete una funcionalidad subir los cambios al repositorio utilizando los comandos de Git.

Esquema de las bases de datos

Antes de empezar a configurar las bases de datos es importante definir el esquema que vamos a utilizar y que clases y relaciones van a tener los datos. Ambas bases de datos van a utilizar el mismo esquema que es el modelo de datos que se puede observar en la Figura 16.

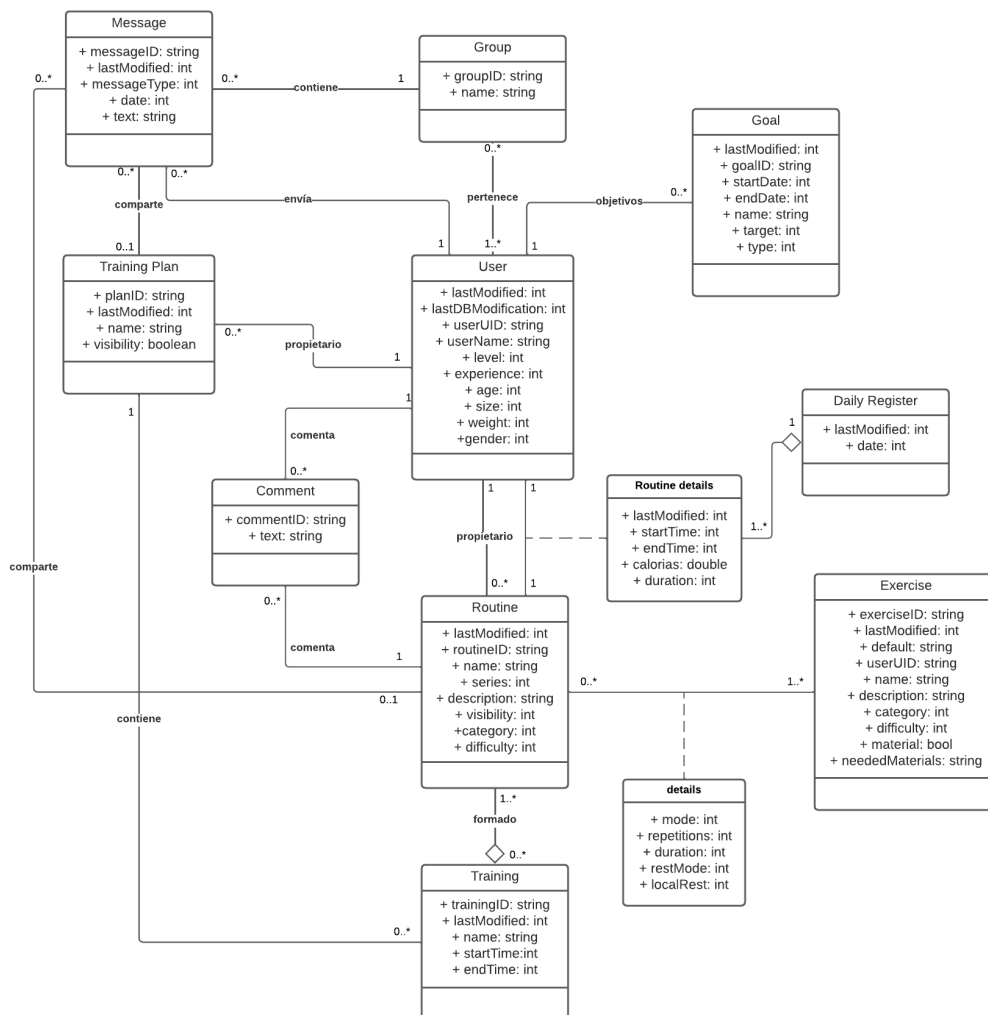


Figura 16. Modelo de datos

Base de datos online

A continuación, vamos a configurar la conexión con el servicio para la base de datos online, que como se ha comentado antes se va a utilizar Firebase. Además, entre los distintos servicios que ofrece la plataforma se van a utilizar Cloud Firestore y Authentication.

Cloud Firestore es una base de datos flexible y escalable en la nube que permite almacenar y sincronizar datos. Authentication permite registrar e iniciar la sesión de los usuarios en la aplicación para que puedan acceder y consultar los datos.

Base de datos local

Una vez configurada la base de datos online, se pasó a configurar la base de datos local. Como base de datos local se va a utilizar la tecnología Realm. En este punto se encontraron dos problemas:

El primer problema, era permitir que los usuarios pudiesen iniciar sesión en la aplicación cuando no tenían conexión a internet. Ya que, al no poder acceder al servidor, no se podía verificar la información de inicio de sesión para autenticar al usuario.

Para permitir que los usuarios pudiesen acceder a la aplicación sin conexión, se decidió:

- Para simplificar la creación de usuarios, que solo se pueda iniciar sesión o registrar a un usuario en la aplicación con conexión a Internet.
- Crear una base de datos local que almacena los perfiles de los usuarios que han iniciado sesión en el dispositivo.
- Cada usuario mantiene un campo que indica si tiene la sesión iniciada o no.

Esta solución, nos ayuda a permitir que el usuario pueda iniciar sesión sin conexión siempre que tenga la sesión iniciada. En caso contrario tendría que conectarse a internet e iniciar sesión o no podría acceder a la aplicación.

El segundo problema que encontramos fue que se podía dar la posibilidad de que, en un mismo dispositivo, se accediera a distintos usuarios y que entre ellos pudiesen ver, modificar o eliminar la información.

Para solucionar este problema, se decidió crear una base de datos para cada usuario que inicie sesión en la aplicación.

Esta solución, nos permite que los datos de cada usuario estén separados y que los demás usuarios no puedan acceder a ellos. Ya que cada usuario solo puede acceder a su base de datos.

Diseño de interfaz responsive

Como queremos que la aplicación se adapte a todo tipo de pantallas debemos diseñar las interfaces para que utilice el espacio disponible en pantalla. Para ello se hará uso de la tecnología que se ha definido anteriormente TailwindCSS.

Gestión de perfiles de usuario

A raíz de los problemas encontrados anteriormente, se decidió introducir esta funcionalidad para permitir que los usuarios pudiesen ver con que perfiles han iniciado sesión en la aplicación y seleccionar con cual acceder. Para ello se creó un selector de perfiles que muestra el nombre del usuario, el correo utilizado para acceder y si la sesión está iniciada.



Se comporta de la siguiente manera, cuando se selecciona un usuario, si la cuenta tiene la sesión iniciada accede directamente a la aplicación, en caso contrario, el usuario es dirigido a la pantalla de inicio de sesión para que ingrese la contraseña para acceder a la aplicación.

Comprobar conexión del dispositivo

A continuación, se desarrolló un componente para detectar el estado de la conexión del dispositivo y poder actuar en consecuencia. Para ello se ha utilizado la librería `NetInfo` que permite monitorizar el estado del dispositivo y actualizar el estado en cuanto el dispositivo tiene o deja de tener conexión a Internet.

Variables globales de la aplicación

Para tener variables globales que se pueden acceder en cualquier componente de la aplicación. Se ha creado un componente llamado `GlobalContext` que mantiene todas las variables que se necesitan.

Para ello se ha utilizado un *hook*²³ de React Native llamado `useContext` que permite crear un contexto que almacena y permite modificar el estado de las variables que se le pasan. Para que puedan ser accedidas los componentes deben ser hijos del `GlobalContext` por lo que hay que envolver la aplicación con este componente.

Aquí guardamos las siguientes variables:

- **IsSignedIn:** Permite consultar si el usuario ha iniciado sesión en la aplicación.
- **hasInternet:** Mantiene el estado de la conexión del dispositivo, haciendo uso del componente que comprueba la conexión.
- **isLoading:** Permite consultar si la aplicación está cargando datos.
- **profile:** Permite obtener el perfil del usuario que ha iniciado sesión.
- **isSincronizing:** Permite consultar si la aplicación está sincronizando las bases de datos.

Autenticado y registro de usuarios

Una vez se desarrollaron los componentes para obtener si el dispositivo tiene conexión a internet y el estado de la aplicación, se desarrolló la funcionalidad para que los usuarios pudiesen iniciar sesión y registrarse en la aplicación.

Para iniciar sesión se creó un formulario donde el usuario introduce su correo y contraseña. Estos campos se validan utilizando las siguientes expresiones regulares para comprobar que el texto introducido es válido. Si el usuario no tiene conexión a internet cuando realiza la petición se le muestra un mensaje informándole de la necesidad de tener conexión para poder iniciar sesión.

Para el registro de los usuarios se creó un formulario donde el usuario introduce su nombre de usuario, correo, contraseña y se le pide que repita la contraseña. Los tres primeros campos se validan usando expresiones regulares y el último comprobando si las dos contraseñas son iguales, para asegurarnos que el usuario no se ha equivocado introduciendo la contraseña y que la recuerda. Además, se introdujo un botón para mostrar o ocultar la contraseña. Al igual que con el inicio de sesión, también se muestra un mensaje informándole de la necesidad de tener conexión para registrarse.

²³ Es una función especial en React y React Native que permite a los componentes funcionales utilizar características de React, como el estado, el ciclo de vida y el contexto, sin necesidad de utilizar una clase.

Sincronización de las bases de datos

Para poder sincronizar los datos correctamente había que tener en cuenta varios casos que se podían producir. Para que sea más sencillo de explicar se ha realizado un diagrama de flujo, que se puede ver en la Figura 17.

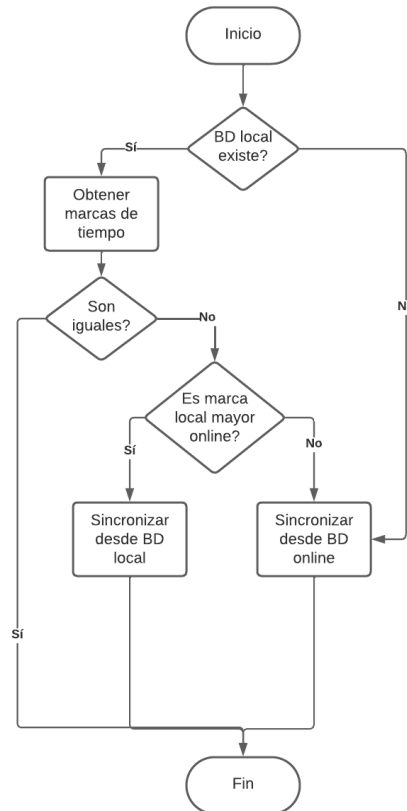


Figura 17. Diagrama de flujo

En primer lugar, se comprueba que en el dispositivo existe la base de datos del usuario; en el caso de que no exista, quiere decir que la copia más actual es la del servidor, por lo que se obtiene la información del usuario y se copia en la base de datos local.

Si la base de datos existe, se compara la marca de tiempo de las dos bases de datos. En el caso en el que las dos marcas de tiempo sean iguales no es necesario sincronizar los datos, ya que las dos bases de datos son iguales.

En el caso en que la marca de tiempo de la base de datos local sea más actual, se deben obtener los datos locales y los de la base de datos online. La sincronización se realiza, como se puede ver en la Figura 18.

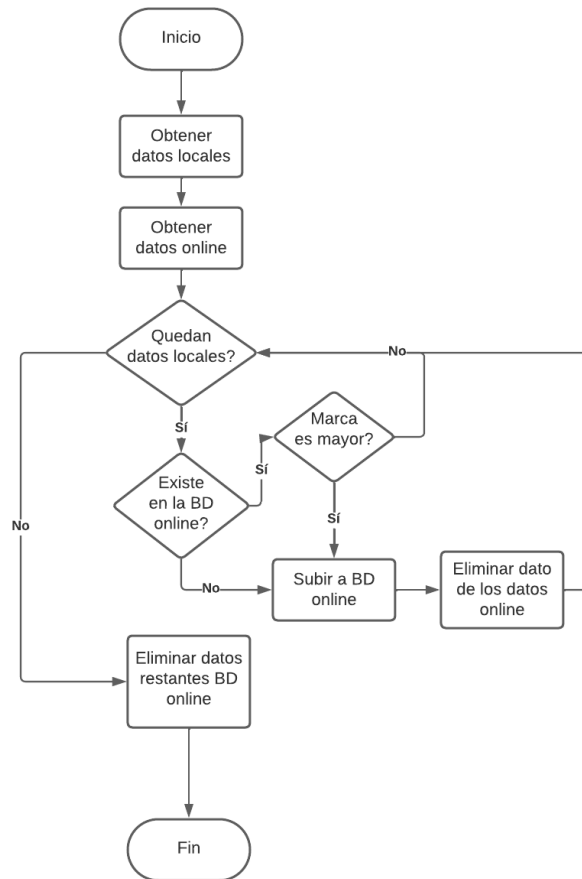


Figura 18. Diagrama de flujo

Como se puede ver en la Figura 18, primero se obtienen los datos de la base de datos local y online y se almacenan en dos vectores que vamos a llamar “*datosLocales*” y “*datosOnline*”.

Si el vector “*datosLocales*” no está vacío, se recorre cada objeto buscando si existe en el vector “*datosOnline*”. Si existe se comparan las marcas de tiempo para saber si es más reciente, en ese caso, se sube a la base de datos online y se elimina del vector “*datosOnline*”, para que la búsqueda de los siguientes datos sea más rápida. En caso contrario se pasa al siguiente objeto. Si el objeto no existe se sube a la base de datos online y se pasa al siguiente dato.

Cuando ya se han recorrido todos los objetos del vector “*datosLocales*”, se recorren los datos restantes en el vector “*datosOnline*” y se eliminan de la base de datos online. Todos estos objetos que quedan en el vector son datos que han sido eliminado en la base de datos local ya que no se encontraban en el vector “*datosLocales*”.

Este proceso se realiza para cada tipo de dato: rutina, ejercicio, plan de entrenamiento, etc. Cuando la base de datos online es más reciente, se realiza el mismo proceso recorriendo el vector “*datosOnline*” y buscando en los “*datosLocales*”.

Creación de ejercicios

A continuación, se desarrolló la funcionalidad para crear ejercicios personalizados, para ello el usuario introduce el nombre del ejercicio, la descripción, la categoría, la dificultad, si necesita material y en ese caso el tipo de material necesario. El ejercicio se guarda y se añade a la lista de ejercicios.

Creación de rutinas

Para desarrollar esta funcionalidad se tuvo en cuenta las necesidades que pueden tener los usuarios a la de crear una rutina. Como la creación de ejercicios personalizados y que puedan configurar al máximo como se realiza la rutina.

En primer lugar, se generaron ejercicios predeterminados que los usuarios puedan usar para crear sus rutinas. Estos ejercicios se generaron haciendo uso de la herramienta ChatGPT y se obtuvieron 150 ejercicios variados que permiten trabajar los músculos principales del cuerpo.

A continuación, se desarrolló la funcionalidad para crear rutinas que permitía configurar la rutina a las necesidades del usuario. Permittedle seleccionar el número de series a realizar, el tiempo de descanso entre ejercicios, el tiempo de descanso entre series, que ejercicios realizar, etc.

Para añadir un ejercicio a la rutina, se despliega un modal que permite seleccionar el ejercicio que se quiere añadir o crear un ejercicio nuevo. En la lista aparecen tanto los ejercicios predeterminados como los creados por el usuario.

Para facilitar que se puedan organizar los ejercicios rápidamente se utilizó una lista que permite arrastrar un ejercicio a la posición deseada y visualizar en que orden se van a realizar.

Mientras se estaba desarrollando la funcionalidad surgió un problema: como se podían crear rutinas con ejercicios que se podían eliminar no era una buena idea almacenar solo el identificador del ejercicio, ya que en caso de ser eliminado de la base de datos la rutina ya no podía realizarse correctamente. Por ello se decidió almacenar directamente una copia de los ejercicios para que en caso de que el ejercicio sea eliminado la rutina se pueda seguir realizando sin ningún problema.

Perfil de usuario

Como última UT se desarrolló la funcionalidad del perfil que permite consultar los datos del usuario, como su nombre de usuario, sexo, edad, peso y altura. Además, se añadió un botón que permite cerrar sesión en la aplicación.

Conclusiones

El *sprint* se ha desarrollado correctamente. Hubo que realizar un ajuste del alcance del proyecto al detectar que no se podían completar todas las UT antes de la finalización del *sprint*. Estas UT se movieron al siguiente *sprint* para realizar su desarrollo.

5.2. Segundo Sprint

Gestión de objetivos

Esta funcionalidad permite al usuario crear objetivos introduciendo un nombre, la fecha de inicio, la fecha de finalización, el tipo de objetivo y la meta que quiere alcanzar. Para calcular el avance del usuario se utiliza el registro diario que permite obtener las rutinas realizadas, la duración y las calorías consumidas. Con estos datos podemos mostrar al usuario si está avanzando en sus objetivos.

Modificación, borrado y copia de rutinas

Para permitir que los usuarios puedan gestionar sus rutinas se desarrolló la funcionalidad para que los usuarios pudiesen modificar, borrar y copiar rutinas. Para poder modificar y eliminar una rutina el usuario debe ser el creador de la rutina, esto evita que cualquier usuario pueda modificar datos que no son suyos.



Cualquier usuario puede realizar una copia de la rutina y modificarla a su gusto sin afectar al creador de la rutina.

Realización de rutinas

Durante este *sprint* se creó la funcionalidad para permitir que el usuario pueda realizar rutinas. Aquí, cada vez que el usuario termina una rutina se almacena la información de la rutina, como las calorías, la hora de inicio, la hora de finalización, la duración, etc.

Para obtener las calorías que consume el usuario se utilizó la fórmula de Harris-Benedict, para calcular la Tasa Metabólica Basal (Mifflin et al., 1990), TMB en adelante.

Las ecuaciones utilizadas son las siguientes:

Género	Ecuación
Hombres	$TMB = (10 \times \text{peso en kg}) + (6.25 \times \text{altura en cm}) - (5 \times \text{edad en años}) + 5.$
Mujeres	$TMB = (10 \times \text{peso en kg}) + (6.25 \times \text{altura en cm}) - (5 \times \text{edad en años}) - 161.$

Estas ecuaciones nos permiten aproximar las calorías que el usuario necesita para realizar sus funciones básicas en reposo. Con esta información y la duración e intensidad del ejercicio, podemos calcular de forma aproximada las calorías que consume el usuario.

Creación de planes entrenamiento

También se desarrolló una funcionalidad para crear planes de entrenamiento donde los usuarios pueden organizar sus entrenamientos. Se utilizó un calendario con agenda para poder visualizar cuando son los entrenamientos de una forma sencilla.

Para que las rutinas de los entrenamientos no fueran inaccesibles si se eliminaban del dispositivo, se decidió realizar una copia de toda la rutina en el entrenamiento. Una vez el entrenamiento ya ha terminado se guarda solamente el nombre de la rutina para ahorrar espacio en el dispositivo.

Modificación y borrado de planes de entrenamiento

Al igual que con las rutinas, para poder modificar y eliminar un plan de entrenamiento el usuario tiene que ser el creador de este.

Conclusiones

En este *sprint* sí que se alcanzaron los objetivos marcados al inicio del *sprint*. Además, al conocer más las tecnologías, se realizó una reestimación de las UT. Estas reestimaciones se acercaron más al coste real de desarrollar cada funcionalidad.

5.3. Tercer Sprint

Calendario de actividad

Para que los usuarios puedan ver los entrenamientos o rutinas realizadas y los entrenamientos programados, se creó un calendario de actividad donde se puede consultar esta información en formato calendario y en formato agenda.

Grupos de entrenamiento

Esta función permite crear grupos de entrenamiento, introduciendo un nombre para el grupo y seleccionando los usuarios que se quieren añadir, de esta manera los usuarios pueden comunicarse dentro de la aplicación.

Además, dentro del chat los usuarios pueden compartir rutinas y planes de entrenamiento. En este punto detectamos que se podían eliminar las rutinas y planes de entrenamiento y que dejaran de estar disponibles si se habían compartido. Por esta razón se decidió realizar una copia del objeto completo para que se pueda acceder, aunque haya sido eliminado el objeto original.

Seguimiento de la actividad del usuario

Otra funcionalidad, que se ha desarrollado durante este *sprint* es la de permitir que el usuario pueda consultar estadísticas de los entrenamientos y rutinas que ha realizado.

Estas estadísticas se generan a partir de los registros guardados al realizar cada rutina. De esta manera se pueden calcular distintos tipos de estadísticas que pueden ser útiles para el usuario.

Comentar rutina

También se ha desarrollado una funcionalidad que permite realizar comentarios en una rutina. Estos comentarios solo están disponibles cuando hay conexión a internet. Se ha decidido hacerlo así para no almacenar esta información en el dispositivo, ya que sin conexión no es una información relevante para el usuario y se estaría utilizando el almacenamiento del dispositivo sin necesidad.



6. Pruebas

Durante el desarrollo de nuestra aplicación, hemos realizado pruebas para garantizar su calidad y funcionalidad. A continuación, vamos a presentar algunas de las pruebas que hemos realizado.

6.1. Pruebas unitarias

A lo largo del proceso, se han llevado a cabo numerosas pruebas unitarias, pero en este apartado solo vamos a presentar un par de ejemplos representativos. Estas pruebas nos han permitido identificar y solucionar rápidamente cualquier problema potencial, brindando confianza en la robustez de nuestro código.

Para realizar estas pruebas hemos utilizado la librería Jest²⁴, un marco de pruebas unitarias para JavaScript que proporciona una sintaxis sencilla y potentes herramientas para escribir y ejecutar pruebas de manera eficiente en proyectos de desarrollo de software.

Por ejemplo, vamos a mostrar una prueba realizada para un fragmento de código que permite eliminar un ejercicio de una rutina.

```
describe('removeExercise', () => {
  test('should remove an exercise from the exercises array', () => {
    const exercise1 = { id: 1, name: 'Exercise 1' };
    const exercise2 = { id: 2, name: 'Exercise 2' };
    const exercise3 = { id: 3, name: 'Exercise 3' };
    const exercises = [exercise1, exercise2, exercise3];
    const result = removeExercise(exercise2, exercises);
    expect(result).toEqual([exercise1, exercise3]);
  });
});
```

Figura 19. Prueba unitaria eliminar ejercicio

Como se puede ver en la Figura 19, para declarar una prueba se utiliza la función “describe” que toma como argumento el nombre de la prueba y la función que va a ejecutar en el test. Dentro de esta función se pueden declarar varias pruebas.

En este caso tenemos una prueba que a su vez recibe el nombre del test y la función que se ejecuta en ese test. Aquí definimos tres ejercicios y los incluimos en un array de ejercicios. Luego llamamos al método que queremos analizar, en este caso, “removeExercise” y almacenamos el valor que devuelve en la variable “result”. Por último, comprobamos que el valor “result”, es igual a un array con el ejercicio 1 y el ejercicio 3. De esta manera podemos validar que el método funciona correctamente y descartar que tenga errores.

²⁴ <https://jestjs.io/es-ES/docs/getting-started>

Otro ejemplo, puede ser las pruebas realizadas sobre un fragmento de código que permite añadir ejercicios a una rutina.

```
describe('addExercise', () => {
  test('should add an exercise to the empty exercises array', () => {
    const exercise = { id: 1, name: 'Exercise 1' };
    const exercises = [];
    const result = addExercise(exercise, exercises);
    expect(result).toEqual([exercise]);
  });

  test('should add a new exercise when it does not exist in the exercises array', () => {
    const exercise1 = { id: 1, name: 'Exercise 1' };
    const exercise2 = { id: 2, name: 'Exercise 2' };
    const exercises = [exercise1];
    const result = addExercise(exercise2, exercises);
    expect(result).toEqual([exercise1, exercise2]);
  });

  test('should update an existing exercise in the exercises array', () => {
    const exercise1 = { id: 1, name: 'Exercise 1' };
    const exercise2 = { id: 2, name: 'Exercise 2' };
    const exercises = [exercise1, exercise2];
    const updatedExercise = { id: 2, name: 'Updated Exercise 2' };
    const result = addExercise(updatedExercise, exercises);
    expect(result).toEqual([exercise1, updatedExercise]);
  });
});
```

Figura 20. Prueba unitaria añadir ejercicio

Como se puede ver en la Figura 20, hemos realizado tres pruebas sobre este método.

- La primera prueba comprueba que se puede añadir un ejercicio a un array vacío.
- La segunda prueba comprueba que se puede añadir un nuevo ejercicio que no existe en el array.
- La tercera prueba comprueba que se actualiza el ejercicio si ya existe en el array.

Con estas pruebas comprobamos todos los casos posibles para este método.

6.2. Pruebas de interfaz de usuario (UI)

Para garantizar una experiencia fluida y atractiva para nuestros usuarios, hemos realizado pruebas de interfaz de usuario. A continuación, presentaremos algunos ejemplos. Estas pruebas nos han permitido evaluar la funcionalidad de nuestra interfaz en diferentes escenarios. A través de estas pruebas, nos aseguramos de que la aplicación se vea y se comporte como se espera, cumpliendo con los estándares de diseño establecidos y ofreciendo una experiencia intuitiva y agradable para nuestros usuarios.

Para realizar estas pruebas hemos utilizado Jest y la biblioteca [@testing-library/react-native](https://bit.ly/43W7TKH)²⁵, que proporciona utilidades y herramientas optimizadas para realizar pruebas de componentes de React Native de manera efectiva y mantenible.

²⁵ <https://bit.ly/43W7TKH>

Por ejemplo, para probar un componente que implementa la funcionalidad para actuar como un `RadioButton` hemos realizado las siguientes pruebas sobre la interfaz para garantizar su correcto funcionamiento.

```
test('renders the label correctly', () => {
  const { getByText } = render(
    <RadioButton data={mockData} index={mockIndex} setIndex={mockSetIndex} label={mockLabel} />
  );
  const labelElement = getByText(mockLabel);
  expect(labelElement).toBeDefined();
});
```

Figura 21. Prueba UI Renderiza label correctamente

Como se puede ver en la Figura 21, hemos implementado un método que comprueba si el componente renderiza correctamente el valor de “label”. Para ello se llama al método “render” de `@testing-library/react-native` y dentro de este método se le pasa el componente. Después se busca por texto un componente que incluya el valor de “label”. Por último, se comprueba que existe.

Otra prueba de este mismo componente es la siguiente.

```
test("should call setIndex when clicking on an item", () => {
  const data = ["Option 1", "Option 2", "Option 3"];
  const label = "Select an option";
  let index = 0;
  const setIndex = jest.fn().mockImplementation((newValue) => {
    index = newValue;
  });

  const { getAllByTestId } = render(
    <RadioButton data={data} index={index} setIndex={setIndex} label={label} />
  );

  const itemElements = getAllByTestId("radial-button-item");

  fireEvent(itemElements[1], 'press');

  expect(setIndex).toHaveBeenCalled();
});
```

Figura 22. Prueba UI Llama setIndex cuando se clica un ítem

En la Figura 22, se ha implementado una prueba para comprobar que, al pulsar en un ítem, se llama correctamente al método “setIndex”. La prueba implica renderizar previamente el componente mediante el método “render” y localizar los elementos etiquetados como “radial-button-item”. Luego, se emula el evento de pulsación (“press”) sobre el segundo ítem mediante el método “fireEvent”. Por último, se verifica que el método “setIndex” ha sido llamado exitosamente, asegurando así el correcto funcionamiento y comportamiento del código en relación con la selección de ítems.

6.3. Análisis estático del código

En este apartado, se explora la aplicación del análisis estático en el contexto del desarrollo de aplicaciones móviles utilizando React Native. Se presenta la herramienta SonarLint²⁶ como un recurso efectivo para analizar y evaluar la calidad del código en tiempo real, identificando patrones de diseño, buenas prácticas y posibles vulnerabilidades. El análisis estático se ha integrado de manera continua en el flujo de trabajo del proyecto, proporcionando una mejora significativa en la calidad del código.

En la primera ejecución al terminar el sprint obtuvimos los siguientes resultados al ejecutar un escaneo más profundo usando SonarQube²⁷.

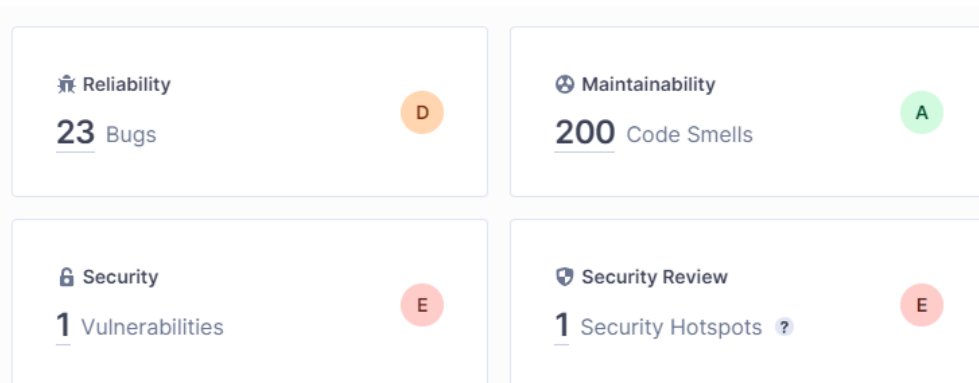


Figura 23. Resultados SonarQube

Como se puede ver en la Figura 23, el escaneo encontró lo siguiente:

- 23 *bugs*²⁸. La mayoría de estos *bugs* eran provocados por utilizar la función “map” y no usar el valor que retornaba. Para solucionarlo se cambió a la función “forEach”.
- 200 *code smells*²⁹. Estos *code smells* eran provocados por importaciones de librerías que no se utilizaban, variables que no se utilizaban, código muerto, etc.
- 1 vulnerabilidad en la seguridad de la aplicación. Esta vulnerabilidad sucedía porque la clave API para acceder a Firebase está en un archivo de la aplicación. Al no planear publicar la aplicación se marcó como falso positivo, pero en una aplicación que se quiera publicar se debería obtener de un servidor con autenticación o con algún método más seguro.
- 1 fallo de seguridad. Este último fallo de seguridad esta provocado por la forma en la que se almacena la contraseña del usuario en la base de datos para los perfiles. Nuevamente se marcó como falso positivo.

²⁶ <https://www.sonarsource.com/products/sonarlint/>

²⁷ <https://www.sonarsource.com/products/sonarqube/>

²⁸ Error o defecto en un programa de software que causa un comportamiento inesperado o incorrecto en su funcionamiento.

²⁹ Indicio o señal de que existe un problema potencial en el diseño o la implementación de un código fuente, lo que puede conducir a un código de baja calidad, dificultades de mantenimiento o posibles errores.

Todos estos problemas se pudieron resolver como se puede ver en la Figura 24.



Figura 24. Resultado SonarQube

7. Conclusiones

En conclusión, el desarrollo de esta aplicación nos ha brindado una comprensión más profunda de los desafíos y oportunidades en el proceso de desarrollo. A través de pruebas y la superación de obstáculos técnicos, hemos logrado implementar las funcionalidades requeridas con éxito.

El primer objetivo que nos marcamos era animar a la sociedad a practicar más ejercicio y creemos que esta aplicación puede ayudar a conseguirlo. Gracias a la gran cantidad de opciones que ofrece los usuarios pueden decidir cuándo y cómo realizar ejercicio. Además, al desarrollar una aplicación para realizar ejercicio también estamos promoviendo que los usuarios tengan unos hábitos de vida saludables.

Otro objetivo que nos marcamos era facilitar la práctica de ejercicio, y creemos que la aplicación cumple este objetivo, ya que ha sido diseñada para que los usuarios puedan realizar ejercicio fácilmente y que de una forma rápida encuentren una rutina que se adapte a sus necesidades.

Al permitir que los usuarios puedan crear grupos de entrenamiento hemos conseguido que se fomente el intercambio de conocimiento entre ellos, permitiendo que compartan rutinas y planes de entrenamiento. De esta manera, los usuarios pueden beneficiarse de la experiencia y conocimientos de otros usuarios y aplicarlo en su actividad física.

También hemos conseguido que los usuarios puedan consultar su progreso diario a través del calendario de actividad y las estadísticas. El calendario de actividad permite registrar y visualizar sus sesiones de ejercicio. Las estadísticas les proporciona datos que pueden ayudarles a conocer el estado en el que se encuentran y mejorar su actividad física.

A través de los planes de entrenamiento hemos logrado que los usuarios puedan planificar sus rutinas, lo que les permite organizar y visualizar sus sesiones en un calendario con una agenda integrada. Esto permite que se organicen y puedan mantener un seguimiento de su progreso, consiguiendo una experiencia de entrenamiento más efectiva y satisfactoria.

Uno de los objetivos principales era permitir que la aplicación funcionase tanto con conexión a Internet como sin ella, lo que permite que se pueda utilizar en cualquier momento y lugar. Esto era vital para no limitar al usuario a realizar ejercicio solo cuando tenía conexión a Internet.

Por último, en la Tabla 11 podemos observar una comparativa de nuestra aplicación con las analizadas al principio de la memoria, en la que se puede ver que características tiene cada aplicación.



Tabla 11. Estudio de competidores

Características de las plataformas					
Nombre	Nike Training Club Ejercicio	Adidas Training Workout HIIT	Strong Workout Tracker Gym Log	Google Fit: Reg. de actividad	FitMe
Rutinas de ejercicio	✓	✓	✓	✗	✓
Rutinas de ejercicio personalizadas	✗	✗	✓	✗	✓
Ejercicios predeterminados	✓	✓	✓	✗	✓
Ejercicios personalizados	✗	✗	✓	✓	✓
Noticias	✓	✓	✗	✓	✗
Seguimiento de la actividad física	✓	✓	✓	✓	✓
Chat y grupos	✗	✗	✗	✗	✓
Calendario de actividad	✗	✗	✓	✗	✓
Integración de relojes inteligentes	✗	✗	✗	✓	✗
Enviar pantalla a otros dispositivos	✓	✗	✗	✗	✗
Entrenador por voz	✓	✓	✗	✗	✗
Integración con Google Fit	✓	✓	✓	✓	✗
Planes de entrenamiento	✗	✗	✗	✗	✓

7.1. Relación con los ODS

En la Tabla 12 podemos ver el Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Tabla 12. Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS)

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

Nuestra aplicación se alinea directamente con el Objetivo de Desarrollo Sostenible número 3: “Garantizar una vida sana y promover el bienestar para todos en todas las edades”. Al proporcionar rutinas de ejercicio que los usuarios pueden adaptar a sus necesidades individuales, la aplicación promueve un estilo de vida activo y saludable. Al fomentar la práctica regular de ejercicio físico, contribuye a reducir los riesgos de enfermedades como la obesidad, la diabetes y las enfermedades cardiovasculares. Además, al ofrecer descripciones sobre cómo realizar los ejercicios correctamente, la aplicación ayuda a los usuarios a realizar los ejercicios correctamente y evitar posibles lesiones, promoviendo así un enfoque seguro y efectivo para mantenerse en forma.

7.2. Trabajos futuros

A continuación, vamos a definir otras funcionalidades que nos hubiera gustado incluir en la aplicación pero que por falta de tiempo no se han podido incluir y que pueden hacer que la aplicación tenga mucho más valor para el usuario.

Integración con Google Fit

Como se puede ver en la Tabla 11 todas las aplicaciones que hemos analizado tienen esta funcionalidad, por lo que puede ser interesante para nuestra aplicación. Esta funcionalidad permitiría a los usuarios obtener la información almacenada en Google Fit y usarla en nuestra aplicación, así como guardarla en Google Fit para combinarla con información de otras aplicaciones.



Integración con relojes inteligentes

Con la gran popularidad de los relojes inteligentes esta funcionalidad sería un plus para nuestra aplicación ya que la mayoría de las aplicaciones no la incluyen. Esta funcionalidad nos permitiría obtener información en tiempo real del usuario como su ritmo cardíaco, calorías consumidas, tipo de actividad que está realizando, etc. Es una funcionalidad que otorgaría un valor añadido a la aplicación y permitiría diferenciarla de otras opciones disponibles.

Generación de rutinas con Inteligencia Artificial

Con el aumento del uso de tecnologías de inteligencia artificial en el último año podría ser interesante desarrollar una funcionalidad que permita a los usuarios generar rutinas de ejercicio personalizadas haciendo uso de esta tecnología. De esta forma el usuario puede generar una rutina que se adapte a sus necesidades, ya que la IA analizaría los datos del usuario, como los días de entrenamiento, estado de forma, edad, peso, altura, etc.

Detección del usuario durante la realización del ejercicio

Esta funcionalidad permitiría guiar al usuario durante la realización de los ejercicios haciendo uso de la cámara del dispositivo y dando instrucciones al usuario por voz a través de los altavoces del dispositivo. Para esta funcionalidad también sería necesario utilizar inteligencia artificial para detectar al usuario y su postura durante la realización de los ejercicios, detectando si está realizando bien o no el ejercicio y dándole indicaciones para que el usuario realice correctamente el ejercicio. Esta funcionalidad puede ayudar a los usuarios con poca experiencia a reducir lesiones por realizar mal los ejercicios.

Mejoras en el chat de la aplicación

También se podrían realizar mejoras en el chat de la aplicación como eliminar, editar y reaccionar a mensajes, mejoras a la hora de gestionar los grupos, más opciones para los usuarios como destacar un mensaje, realizar votaciones, etc.

Incluir imágenes y videos en los ejercicios

Otro aspecto que mejoraría la aplicación sería añadir imágenes y videos a los ejercicios para que sea más sencillo entender cómo realizarlo correctamente. Actualmente tan solo tenemos una descripción de cómo se realiza el ejercicio que en algunas ocasiones puede no ser suficiente para realizarlo correctamente.

Añadir entrenador por voz

Esta funcionalidad puede ser interesante para poder guiar a los usuarios a través de la rutina sin que necesariamente tengan que estar pendiente del móvil para saber que ejercicio es el siguiente y como realizarlo. Además, si se combina con las instrucciones de detección del usuario puede ser ideal para que actúe como entrenador personal dando instrucciones que ayuden al usuario a sacar el máximo a cada ejercicio.

7.3. Relación del trabajo desarrollado con los estudios cursados

Todo lo que hemos aplicado en este proyecto está relacionado con los estudios cursados. Además, al haber cursado la rama de Ingeniería del Software, nos hemos formado en muchos aspectos relacionados con el desarrollo de software.

Un aspecto muy importante es como se trabaja en la actualidad en un proyecto, la metodología que hemos aplicado se ha trabajado en las asignaturas de Proceso de Software (PSW) y Proyecto de Ingeniería de Software (PIN), en estas asignaturas se desarrollaron proyectos utilizando la metodología ágil Scrum. Donde hemos aprendido la importancia que tiene la metodología ágil en los proyectos que se desarrollan en la actualidad.

Otro aspecto importante a la hora de desarrollar una aplicación es definir cómo va a ser el sistema y que funcionalidades va a tener. Todo esto lo hemos aprendido en las asignaturas de Calidad de software (CSO) y Análisis y especificación de requisitos (AER).

Respecto a la forma de validar una aplicación y verificar si tiene fallos hemos aplicado lo aprendido en las asignaturas de CSO, Diseño de software (DDS), Análisis, validación y depuración de software (AVD), Ingeniería de Software (ISW) y Mantenimiento y evolución de software (MES). En estas asignaturas también aprendimos técnicas de código limpio, patrones de diseño, arquitecturas software, entre otras muchas cosas.



8. Referencias

- adidas Training: Workout HIIT - Aplicaciones en Google Play.* (s. f.). Recuperado 6 de agosto de 2023, de <https://play.google.com/store/apps/details?id=com.runtastic.android.results.lite&hl=es>
- Code Quality Tool & Secure Analysis with SonarQube.* (s. f.). Recuperado 6 de agosto de 2023, de <https://www.sonarsource.com/products/sonarqube/>
- Empezando · Jest.* (s. f.). Recuperado 6 de agosto de 2023, de <https://jestjs.io/es-ES/docs/getting-started>
- Expo Documentation.* (s. f.). Expo Documentation. Recuperado 6 de agosto de 2023, de <https://docs.expo.dev/>
- Firebase | Google's Mobile and Web App Development Platform.* (s. f.). Firebase. Recuperado 6 de agosto de 2023, de <https://firebase.google.com/?hl=es-419>
- Getting Started | React Native Testing Library.* (s. f.). Recuperado 6 de agosto de 2023, de <https://callstack.github.io/react-native-testing-library/docs/getting-started>
- GitHub: Let's build from here.* (s. f.). GitHub. Recuperado 6 de agosto de 2023, de <https://github.com/>
- Google Fit: Reg. de actividad - Aplicaciones en Google Play.* (s. f.). Recuperado 6 de agosto de 2023, de <https://play.google.com/store/apps/details?id=com.google.android.apps.fitness&hl=es>
- Installation—Tailwind CSS.* (s. f.). Recuperado 6 de agosto de 2023, de <https://tailwindcss.com/docs/installation>
- Introduction · React Native.* (2023, junio 21). <https://reactnative.dev/docs/getting-started>

JavaScript | MDN. (2023, julio 8). <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Ken Schwaber & Jeff Sutherland. (2020). *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*.
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>

Lint IDE Tool & Real-Time Software for Code | Sonar. (s. f.). Recuperado 6 de agosto de 2023, de <https://www.sonarsource.com/products/sonarlint/>

Mifflin, M., St Jeor, S., Hill, L., Scott, B., Daugherty, S., & Koh, Y. (1990). A new predictive equation for resting energy expenditure in healthy individuals. *The American Journal of Clinical Nutrition*, 51(2), 241-247.
<https://doi.org/10.1093/ajcn/51.2.241>

Nike Training Club—Ejercicio—Apps en Google Play. (s. f.). Recuperado 6 de agosto de 2023, de
https://play.google.com/store/apps/details?id=com.nike.ntc&hl=es_419

Organización Mundial de la Salud. (2021). *Directrices de la OMS sobre actividad física y comportamientos sedentarios*. Organización Mundial de la Salud.
<https://apps.who.int/iris/handle/10665/349729>

Strong Workout Tracker Gym Log—Aplicaciones en Google Play. (s. f.). Recuperado 6 de agosto de 2023, de
<https://play.google.com/store/apps/details?id=io.strongapp.strong&hl=es>

Visual Studio Code—Code Editing. Redefined. (s. f.). Recuperado 6 de agosto de 2023, de <https://code.visualstudio.com/>

Welcome to the Realm Docs—Realm. (s. f.). Recuperado 6 de agosto de 2023, de <https://www.mongodb.com/docs/realm/>



Anexo A

Casos de uso

CU03 Realizar rutina

Descripción	La pantalla permite al usuario realizar la rutina seleccionada en el caso de uso "CU02_Consultar rutina". Una vez que inicia la rutina, puede terminarla, pausarla y realizar los ejercicios correspondientes.	
Precondición	El usuario ha iniciado sesión, ha seleccionado una rutina en el caso de uso "CU02_Consultar rutina" y se encuentra en la pantalla de inicio de la rutina.	
Secuencia normal	Paso	Acción
	1	Se muestra el primer ejercicio de la rutina.
	2	El usuario realiza el ejercicio.
	3	El sistema registra el progreso del ejercicio y muestra el siguiente ejercicio.
	4	El usuario continúa realizando los ejercicios hasta que finalice la rutina.
	5	El usuario puede pausar la rutina en cualquier momento.
	6	Si el usuario pausa la rutina, se guarda el progreso y se muestra la opción de continuar o terminar la rutina.
7	El usuario puede finalizar la rutina cuando haya completado todos los ejercicios.	
Postcondición	Se registra el progreso de la rutina realizada por el usuario.	
Excepciones	Si no se puede acceder a la información de la rutina, se muestra un mensaje de error.	
	Si ocurre un error al registrar el progreso de la rutina o al realizar los ejercicios, se muestra un mensaje de error.	

CU04 Eliminar rutina

Descripción	El usuario puede eliminar la rutina seleccionada en el caso de uso "CU02_Consultar rutina".	
Precondición	El usuario ha iniciado sesión y ha seleccionado una rutina en el caso de uso "CU02_Consultar rutina".	
Secuencia normal	Paso	Acción
	1	El usuario confirma la eliminación de la rutina.
Postcondición	La rutina seleccionada se elimina del sistema.	
Excepciones	Sin excepciones	

CU05 Editar rutina

Descripción	El usuario puede editar la información de la rutina seleccionada en el caso de uso "CU02_Consultar rutina".	
Precondición	El usuario ha iniciado sesión, ha seleccionado una rutina en el caso de uso "CU02_Consultar rutina" y se encuentra en la pantalla de edición de la rutina.	
Secuencia normal	Paso	Acción
	1	El usuario puede modificar el nombre, descripción, categoría, dificultad, visibilidad, número de series, tiempo de descanso entre series y descanso entre ejercicios de la rutina.
	2	El usuario puede añadir o eliminar ejercicios de la rutina.
	3	El usuario puede reordenar los ejercicios arrastrándolos a la posición deseada.
4	El sistema registra el progreso del ejercicio y muestra el siguiente ejercicio.	
Postcondición	Los cambios realizados se guardan en la rutina seleccionada.	
Excepciones	El usuario no completa todos los campos requeridos.	
	El usuario no selecciona ningún ejercicio.	

CU06 Comentar rutina

Descripción	El usuario puede realizar un comentario en la rutina seleccionada en el caso de uso "CU02_Consultar rutina".	
Precondición	El usuario ha iniciado sesión y ha seleccionado una rutina en el caso de uso "CU02_Consultar rutina".	
Secuencia normal	Paso	Acción
	1	El usuario escribe su comentario en un campo de texto designado.
2	El usuario envía el comentario.	
Postcondición	El comentario se guarda y se muestra asociado a la rutina seleccionada.	
Excepciones	El usuario no ha proporcionado ningún comentario antes de enviar.	

CU07 Votar rutina

Descripción	El usuario puede votar una rutina de la lista de rutinas pulsando el botón de "me gusta".	
Precondición	El usuario ha iniciado sesión y ha accedido a la lista de rutinas.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la rutina en la lista de rutinas.
2	El usuario pulsa el botón de "me gusta" asociado a la rutina.	
Postcondición	Se registra el voto del usuario para la rutina seleccionada.	
Excepciones	Sin excepciones	

CU10 Añadir entrenamiento

Descripción	Permite al usuario añadir un entrenamiento al plan de entrenamiento seleccionado en el caso de uso "CU09_Consultar plan de entrenamiento".	
Precondición	El usuario ha iniciado sesión y ha accedido al plan de entrenamiento deseado.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona una rutina existente para el entrenamiento.
2	El usuario introduce la fecha, la hora de inicio y la hora de fin para el entrenamiento.	
Postcondición	Se añade un nuevo entrenamiento al plan de entrenamiento con la rutina, fecha y horas especificadas.	
Excepciones	El usuario no ha proporcionado una fecha, hora de inicio o hora de fin válidas para el entrenamiento.	

CU11 Eliminar rutina

Descripción	Permite al usuario eliminar un entrenamiento del plan de entrenamiento seleccionado en el caso de uso "CU09_Consultar plan de entrenamiento".	
Precondición	El usuario ha iniciado sesión y ha accedido al plan de entrenamiento deseado.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa el botón de eliminar asociado al entrenamiento.
Postcondición	El entrenamiento seleccionado se elimina del plan de entrenamiento.	
Excepciones	Sin excepciones	

CU12 Borrar plan de entrenamiento

Descripción	Permite al usuario eliminar un plan de entrenamiento pulsando el botón de eliminar.	
Precondición	El usuario ha iniciado sesión y ha accedido a la lista de planes de entrenamiento.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa el botón de eliminar asociado al plan de entrenamiento.
Postcondición	El plan de entrenamiento seleccionado se elimina del sistema.	
Excepciones	Sin excepciones	

CU15 Acceder al grupo

Descripción	Permite al usuario acceder al chat de un grupo de ejercicio y compartir rutinas, planes de entrenamiento y enviar mensajes.	
Precondición	El usuario ha iniciado sesión y ha accedido al grupo de ejercicio deseado.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona el grupo de ejercicio al que desea acceder.
	2	El usuario accede al chat del grupo.
	3	El usuario puede compartir rutinas y planes de entrenamiento en el chat.
	4	El usuario puede enviar mensajes.
Postcondición	El usuario está activo en el chat del grupo y puede interactuar con los demás miembros.	
Excepciones	Sin excepciones.	

CU16 Compartir rutina

Descripción	Permite al usuario compartir una rutina en el grupo de ejercicio al que ha accedido.	
Precondición	El usuario ha iniciado sesión y ha accedido al grupo de ejercicio deseado.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción de compartir en el grupo.
	2	El usuario selecciona la rutina que desea compartir.
Postcondición	La rutina seleccionada se comparte en el grupo de ejercicio.	
Excepciones	El usuario no ha seleccionado una rutina.	

CU17 Seleccionar rutina

Descripción	Permite al usuario seleccionar una rutina.	
Precondición	El usuario ha iniciado sesión, ha accedido al grupo de ejercicio deseado y ha seleccionado compartir rutina.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la rutina que desea compartir.
Postcondición	La rutina seleccionada se comparte en el grupo de ejercicio.	
Excepciones	El usuario no ha seleccionado una rutina.	

CU18 Compartir plan de entrenamiento

Descripción	Permite al usuario compartir un plan de entrenamiento en el grupo de ejercicio al que ha accedido.	
Precondición	El usuario ha iniciado sesión y ha accedido al grupo de ejercicio deseado.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción de compartir en el grupo.
	2	El usuario selecciona el plan de entrenamiento que desea compartir.
Postcondición	El plan de entrenamiento seleccionado se comparte en el grupo de ejercicio.	
Excepciones	El usuario no ha seleccionado un plan de entrenamiento.	

CU19 Seleccionar plan de entrenamiento

Descripción	Permite al usuario seleccionar un plan de entrenamiento.	
Precondición	El usuario ha iniciado sesión, ha accedido al grupo de ejercicio deseado y ha seleccionado compartir plan de entrenamiento.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona el plan de entrenamiento que desea compartir.
Postcondición	El plan de entrenamiento seleccionado se comparte en el grupo de ejercicio.	
Excepciones	El usuario no ha seleccionado una rutina.	

CU20 Enviar mensaje

Descripción	Permite al usuario enviar un mensaje al grupo al que ha accedido.	
Precondición	El usuario ha iniciado sesión y ha accedido al grupo de ejercicio deseado.	
Secuencia normal	Paso	Acción
	1	El usuario redacta un mensaje en el área de texto.
	2	El usuario selecciona la opción de enviar el mensaje al grupo.
Postcondición	El mensaje se envía al grupo de ejercicio y es visible para los miembros.	
Excepciones	El mensaje enviado está vacío.	

CU21 Abandonar grupo

Descripción	Permite al usuario abandonar el grupo al que ha accedido.	
Precondición	El usuario ha iniciado sesión y ha accedido al grupo de ejercicio deseado.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción de abandonar el grupo.
	2	El sistema muestra una confirmación para confirmar la acción.
	3	El usuario confirma que desea abandonar el grupo.
Postcondición	El usuario abandona el grupo y ya no forma parte de sus miembros.	
Excepciones	Sin excepciones.	

CU22 Eliminar grupo

Descripción	Permite al usuario eliminar el grupo de la lista de grupos sin dejar de formar parte de este.	
Precondición	El usuario ha iniciado sesión y tiene acceso a la lista de grupos.	
Secuencia normal	Paso	Acción
	1	El usuario selecciona el grupo que desea eliminar.
	2	El usuario selecciona la opción de eliminar el grupo.
	3	El sistema muestra una confirmación para confirmar la acción.
	4	El usuario confirma que desea eliminar el grupo.
Postcondición	El grupo se elimina de la lista de grupos, pero el usuario aún forma parte de él.	
Excepciones	El grupo seleccionado no existe o no es válido.	



CU25 Seguimiento entrenamiento

Descripción	Permite al usuario ver sus estadísticas de entrenamiento a lo largo del tiempo.	
Precondición	El usuario ha iniciado sesión y tiene registros de entrenamiento almacenados.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la sección de seguimiento de entrenamiento.
	2	El sistema muestra las estadísticas del usuario, como el progreso, rendimiento, tiempos, repeticiones, etc., a lo largo del tiempo.
	3	El usuario puede seleccionar un periodo específico para ver estadísticas detalladas.
	4	El usuario puede visualizar gráficos o tablas que representen su progreso.
Postcondición	El usuario tiene acceso a sus estadísticas de entrenamiento y puede analizar su progreso.	
Excepciones	El usuario no ha iniciado sesión.	
	No existen registros de entrenamiento para el usuario.	

CU26 Calendario de entrenamiento

Descripción	Permite al usuario ver en un calendario los entrenamientos y rutinas realizadas y programadas.	
Precondición	El usuario ha iniciado sesión y tiene registros de entrenamiento almacenados.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la sección del calendario.
	2	El sistema muestra un calendario con los días del mes actual.
	3	Los entrenamientos realizados se resaltan o se muestran de alguna manera en el calendario.
	4	Los entrenamientos programados se muestran en las fechas correspondientes.
Postcondición	El usuario puede visualizar los entrenamientos realizados y programados en el calendario.	
Excepciones	El usuario no ha iniciado sesión.	
	No existen registros de entrenamiento para el usuario.	

CU27 Consultar objetivos

Descripción	Permite al usuario consultar sus objetivos, crear nuevos objetivos y eliminarlos.	
Precondición	El usuario ha iniciado sesión y accede a la sección de objetivos.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la sección de objetivos.
	2	El sistema muestra los objetivos actuales del usuario.
	3	El usuario puede crear un nuevo objetivo proporcionando los detalles requeridos.
	4	El usuario puede eliminar un objetivo existente seleccionando la opción de eliminar.
Postcondición	El usuario puede consultar sus objetivos, crear nuevos objetivos y eliminarlos.	
Excepciones	El usuario no ha iniciado sesión.	
	No existen registros de entrenamiento para el usuario.	

Anexo B

Implantación

En este apartado, se crea la versión de producción de la aplicación para que pueda ser distribuida en cualquier dispositivo Android. Para ello hay que compilar la aplicación haciendo uso de la librería “eas-cli”.

Para instalar esta librería se utiliza el siguiente comando:

```
npm install -g eas-cli
```

Para poder determinar la configuración de cada versión es necesario generar un archivo eas.json. En nuestro caso el contenido del archivo final fue el siguiente.

```
{
  "cli": {
    "version": ">= 3.15.0"
  },
  "build": {
    "development": {
      "distribution": "internal",
      "android": {
        "gradleCommand": ":app:assembleDebug"
      },
      "ios": {
        "buildConfiguration": "Debug"
      }
    },
    "preview": {
      "android": {
        "buildType": "apk"
      }
    },
    "production": {}
  },
  "submit": {
    "production": {}
  }
}
```

Figura 25. Contenido archivo eas.json

Como se puede ver en la Figura 25, en el archivo tenemos distintas versiones para nuestra aplicación:

- **Development:** Esta es la versión de desarrollo y necesita de un servidor de desarrollo local para poder funcionar. Es la versión que hemos estado utilizando para desarrollar la aplicación.
- **Preview:** Esta es la versión que se enviaría a los stakeholders para que puedan ver el funcionamiento de la aplicación. Se ha definido la configuración “**buildType: apk**” para que genere un archivo de instalación APK que pueda instalarse en dispositivos Android.
- **Production:** Es la versión que recibirían los usuarios de la aplicación y que sería publicada en las tiendas de cada plataforma.

También se pueden subir directamente las versiones a las distintas tiendas de cada plataforma con el comando:

```
eas submit
```

Para compilar la aplicación se utiliza el siguiente comando:

```
eas build --profile preview --platform android
```

Una vez se completa la compilación se obtiene un archivo que contiene la versión de desarrollo y la versión de producción. Esta segunda versión esta lista para instalarse en cualquier dispositivo Android.

Además, cuando la aplicación esta compilada el código Javascript se convierte a código nativo del dispositivo y se optimiza para que el rendimiento sea el esperado en una aplicación nativa. De esta manera se puede ver un incremento en el rendimiento de esta versión comparado con el que se obtiene con la versión de desarrollo.