
This is the **published version** of the bachelor thesis:

Mateos Cañadas, Ana; Talló Sendra, Marc, tut. Aplicación para el registro y seguimiento de actividades deportivas en gimnasios. 2025. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/308782>

under the terms of the  license



This is the **published version** of the bachelor thesis:

Mateos Cañadas, Ana; Talló Sendra, Marc, tut. Aplicación para el registro y seguimiento de actividades deportivas en gimnasios. 2025. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/308782>

under the terms of the  license

Aplicación para el registro y seguimiento de actividades deportivas en gimnasios

Ana Mateos Cañadas

Resumen– Actualmente, el interés por el ejercicio físico y la práctica de actividades deportivas, especialmente en gimnasios, está en auge, lo que impulsa la necesidad de herramientas que ayuden a los usuarios a gestionar sus entrenamientos, llevar un registro de sus rutinas y mantenerse motivados. Aunque existen aplicaciones similares, muchos usuarios expresan la necesidad de una herramienta más intuitiva y que ofrezca una gestión personalizada para cada usuario. Esta aplicación permitirá a los usuarios crear sus rutinas de entrenamiento, registrar los detalles de los ejercicios realizados y visualizar su progreso. Además, incluirá un amplio catálogo de ejercicios con descripciones e imágenes, y ofrecerá la opción de rutinas supervisadas por entrenadores profesionales. Esta herramienta busca mejorar la gestión del entrenamiento personal, motivar a los usuarios y facilitar el alcance de sus metas deportivas.

Palabras clave– Aplicación, Gimnasio, Registro de entrenamientos, React Native

Abstract– Currently, the interest in physical exercise and the practice of sports activities, especially in gyms, is on the rise, driving the need for tools that help users manage their workouts, track their routines, and stay motivated. Although similar applications exist, many users express the need for a more intuitive tool that offers personalized management for each user. This application will allow users to create their workout routines, record the details of the exercises performed and visualize their progress. It will also include an extensive catalog of exercises with descriptions and images, and will offer the option of routines supervised by professional trainers. This tool seeks to improve the management of personal training, motivate users and facilitate the achievement of their sports goals.

Keywords– App, Gym, Training registration, React Native

1 INTRODUCCIÓN

El ejercicio físico y la práctica de actividades deportivas se han convertido en aspectos fundamentales para el bienestar y la salud de las personas, y cada vez son más los que optan por este estilo de vida más saludable y activo. Aproximadamente el 16,5 % de la población española va al gimnasio, según “Informe OBS: La industria del deporte y el fitness” [1]. El sector del deporte y el fitness representa el 3,3 % del PIB español (en Europa esta cifra está entre el 1,5 % y el 2 %).

A pesar de esta creciente popularidad de los gimnasios, muchos usuarios carecen de las herramientas necesarias. Esta falta de recursos puede llevar a la desmotivación y/o una gestión ineficaz de las rutinas, dificultando el alcance de sus objetivos.

Por este motivo y por mi experiencia personal como usuaria de aplicaciones similares surge la idea de realizar este trabajo. Me entusiasma la idea de poder ayudar a las personas a gestionar y llevar un control detallado de sus rutinas de entrenamiento y alcanzar sus objetivos personales de manera más sencilla bajo la supervisión de profesionales cualificados, sin costes adicionales y con la información clara y accesible. Además, este trabajo me permite combinar mis dos pasiones: el deporte y la informática.

Este proyecto, por tanto, se enfoca en el desarrollo de una aplicación móvil destinada a usuarios de gimnasios, con el objetivo de ofrecer una solución que permita personalizar, registrar y monitorear el progreso de los entrenamientos de manera detallada. Para ello, se han utilizado tecnologías en tendencia, como React Native y Expo para el desarrollo del frontend, junto con Node.js, Express y MongoDB para la implementación del backend.

-
- E-mail de contacto: anamateosca@gmail.com
 - Mención realizada: Ingeniería del Software
 - Trabajo tutorizado por: Marc Talló (Ciencias de la Computación)
 - Curso 2024/25

2 OBJETIVOS

Este apartado detalla los objetivos principales y específicos planteados para el desarrollo del proyecto.

Objetivos principales:

- Ofrecer una solución para las personas que desean un control detallado de sus avances en el gimnasio.
- Permitir a los usuarios registrar sus sesiones de entrenamiento.
- Proporcionar un catálogo detallado de ejercicios para que los usuarios sepan la correcta ejecución.
- Incorporar la posibilidad de que los entrenamientos y/o planificaciones sean supervisados por entrenadores profesionales.
- Personalización de planes de entrenamiento ajustados a metas concretas.

Objetivos específicos:

- Aprender y aplicar técnicas de desarrollo móvil utilizando frameworks y lenguajes de programación, como React Native.
- Diseñar e implementar una API RESTful para gestionar la comunicación entre cliente y servidor.
- Diseñar y desarrollar interfaces de usuario intuitivas y atractivas.
- Gestionar el ciclo de vida del desarrollo del software.

3 ESTADO DEL ARTE

En el contexto actual, el interés por las actividades deportivas ha impulsado considerablemente este tipo de aplicaciones. Existen diversas aplicaciones en el mercado bastante competitivas y completas, como Jefit, Strong, Pro-Fit o Freeletics. Estas aplicaciones ofrecen funcionalidades como la creación de entrenamiento, la sugerencia de rutinas según los objetivos del usuario, el seguimiento en tiempo real de sesiones de entrenamiento, visualizar un registro histórico de los entrenamientos previos y el seguimiento de estadísticas relacionadas con la salud.

También se han tenido en consideración aplicaciones como Wikiloc y Strava, que, aunque son plataformas especializadas en actividades al aire libre, incluyen ciertas características que resultan útiles para este proyecto. Estas aplicaciones destacan por cómo se comportan como redes sociales, permitiendo agregar a otros usuarios y visualizar el progreso de sus contactos.

Sin embargo, estas aplicaciones suelen presentar varios inconvenientes:

- Pagos excesivos y acceso restringido a funciones.
- Falta de personalización.
- Interfaces poco intuitivas y complejas.

- Nula de supervisión de entrenadores profesionales.

A pesar de las muchas aplicaciones en el mercado, todavía hay margen para mejorar. Este proyecto está pensado para usuarios que simplemente quieren llevar un control de sus avances en el gimnasio de forma sencilla y con total libertad y personalización. Además, para hacer que el proyecto sea sostenible, se optará para un futuro un modelo de monetización freemium [2]. Y lo más importante, incorpora la supervisión de entrenadores profesionales.

4 METODOLOGÍA

Para el correcto desarrollo de este proyecto se ha utilizado una combinación de las metodologías de **desarrollo incremental** y **Kanban** [3]. Esta combinación es ideal para proyectos individuales. Se ha seleccionado **Jira Software**, herramienta para la gestión de proyectos que facilita la planificación y seguimiento de tareas. Permitiendo organizar de forma visual el progreso.

La metodología incremental permite dividir el trabajo en módulos y construir cada uno de manera independiente, integrándolo con las partes anteriores una vez que esté terminado. Esto facilitará un control progresivo del proyecto, permitiendo hacer ajustes a medida que se completa cada módulo y asegurando un desarrollo más flexible y eficiente.

Kanban permite gestionar las tareas de manera visual mediante el uso de un tablero con columnas, permitiendo representar de forma clara el estado de cada tarea y el progreso global. Este tablero divide las tareas en etapas como “Por hacer”, “En curso” y “Listo”.

5 PLANIFICACIÓN

La planificación de este proyecto se ha estructurado en varias fases de desarrollo incluyendo el diseño, desarrollo I y desarrollo II. A continuación, se detallan estas fases:

Fase de diseño: Se comenzó con el diseño del prototipo, para poder pasar a implementar el esquema general de la interfaz y navegar entre los módulos principales de la aplicación. Posteriormente, se implementó el primer módulo de *Ejercicios* y sus detalles. Por último, se configuró la autenticación de usuarios, junto con la implementación de las pantallas de *Inicio de Sesión*, *Registro* y *Perfil*.

Fase de desarrollo: En esta fase se desarrollaron las funcionalidades principales que dan cuerpo a la aplicación. Primero se completó el módulo de *Entrenamientos*, seguido del módulo *Comenzar* (Sesión de entrenamiento) y, finalmente, se implementó el módulo de *Planificaciones*.

Fase de desarrollo II: Se añadieron mejoras como la funcionalidad de compartir planificaciones y entrenamientos dentro del módulo de *Rutinas*, así como la incorporación de filtros de búsqueda en el módulo *Ejercicios*. Posteriormente, se desarrollaron los últimos módulos restantes: *Historial* y *Panel de Administración*. Finalmente, se mejoró la interfaz para hacerla más intuitiva y sin errores.

A continuación, se presenta el cronograma de la planificación final:



Fig. 1: Cronograma Jira Software

Como se mencionó en el apartado de Metodología, el trabajo se ha dividido en módulos (Epics) dentro de Jira Software. Estas Epics contienen subtareas, que se presentan de forma más detallada en el apéndice A.1.

6 REQUISITOS

Para garantizar que la aplicación cumpla con los objetivos establecidos, se definieron una serie de requisitos funcionales, no funcionales y técnicos. Estos se clasificaron siguiendo la metodología MoSCoW [4], priorizándolos en Must, Should y Could.

No se escatimó en la definición de requisitos, ya que esto no solo ayuda a hacer la aplicación más competitiva, sino que también permite identificar aspectos que podrían ser interesantes para un futuro. Como resultado, se implementaron todos los requisitos Must, mientras que algunos Should y Could se han reservado para futuras ampliaciones. El detalle completo de los requisitos se encuentra en el apéndice A.2.

7 PROTOTIPO

Antes de comenzar con el desarrollo, se diseñó un prototipo utilizando la herramienta **Figma**, que permite visualizar y probar la estructura de la interfaz de usuario, así como definir la navegación entre módulos y transiciones de pantalla mediante la funcionalidad de Flows.

Además, se llevaron a cabo pruebas con usuarios para simular la navegación real dentro de la aplicación. Esto permitió detectar mejoras en la disposición de los elementos y garantizar una buena experiencia de usuario.

A continuación se muestran el prototipo final:

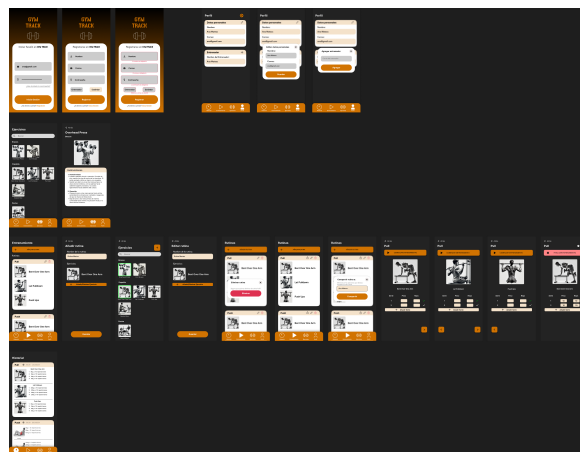


Fig. 2: Prototipo

8 DIAGRAMA DE CASOS DE USO

Para la creación de diagramas de casos de uso de la aplicación, se ha utilizado **PlantUML**, una herramienta de programación para crear diagramas a partir de texto simple.

El siguiente diagrama muestra las interacciones principales de cada tipo de usuario con la aplicación, permitiendo entender las funcionalidades disponibles para cada rol. La aplicación cuenta con tres roles: Administrador, Entrenador y Usuario Estándar. Los dos últimos comparten casi todas las funcionalidades, con la única diferencia de que el Entrenador tiene la capacidad adicional de compartir rutinas.

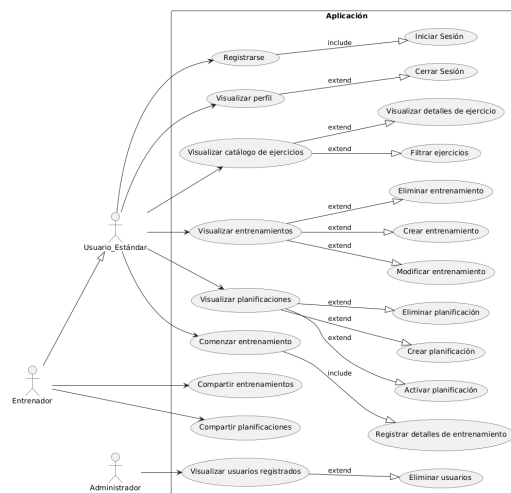


Fig. 3: Diagrama de casos de uso

9 DESARROLLO

Para preparar el entorno de desarrollo del proyecto, se configuró el equipo con las herramientas y tecnologías necesarias para soportar tanto el frontend como el backend de la aplicación. A continuación, se describen los pasos previos al desarrollo:

Instalación de Node.js y npm: Node.js es esencial para ejecutar el código del lado del servidor, y npm (Node

Package Manager) permite la gestión de dependencias del proyecto. Esta combinación facilita la instalación de librerías y frameworks necesarios como Express, Mongoose, Cloudinary, Expo...

Instalación de Git: Git se utilizó para el control de versiones del código. Al integrar Git con **GitHub**, fue posible gestionar dos repositorios separados para el backend y frontend, permitiendo almacenar y controlar el código fuente para un seguimiento detallado de los cambios.

Instalación de Visual Studio Code (VSCode): Se utilizó como IDE (entorno de desarrollo integrado) para la edición y desarrollo del código de forma eficiente.

El apartado de desarrollo se estructura en dos subapartados: Backend y Frontend.

9.1. Backend

El desarrollo del backend de la aplicación se ha estructurado para garantizar una gestión eficiente de datos y una integración segura con el frontend.

9.1.1. Plataforma y framework

El proyecto utiliza **Node.js** [5] como plataforma para la ejecución del código del lado del servidor, junto con **Express**, un framework diseñado para la creación de una API robusta y eficiente.

9.1.2. Base de Datos

La base de datos seleccionada es **MongoDB**, una solución NoSQL que permite un almacenamiento flexible y escalable.

9.1.3. Librerías

Para interactuar con MongoDB desde Node.js, se ha utilizado **Mongoose**, una librería que proporciona un esquema de datos y herramientas para la validación, manipulación y consulta de datos.

Para mejorar la seguridad y la configuración del backend (MongoDB Atlas, Cloudinary y Firebase) se ha utilizado la librería **dotenv**, que permite la gestión de variables de entorno. Esto es fundamental para evitar exponer datos sensibles directamente en el código fuente.

9.1.4. Cloud

La base de datos está alojada en **MongoDB Atlas**, un servicio en la nube que facilita la administración, escalabilidad y seguridad de los datos.

Para almacenar y gestionar las imágenes de los ejercicios, se ha utilizado **Cloudinary**, un servicio de gestión de imágenes en la nube que facilita la carga, optimización y acceso a imágenes de manera rápida y eficiente.

Se utiliza **Firebase Admin** para la administración de usuarios desde el backend, permitiendo la eliminación de cuentas cuando sea necesario.

9.1.5. Despliegue

El servidor está alojado en **Render** [6], una plataforma que mediante el acceso al repositorio GitHub del proyecto permite desplegar y gestionar aplicaciones de manera sencilla.

9.2. Frontend

El desarrollo del frontend se ha llevado a cabo utilizando diversas herramientas y librerías para asegurar una estructura sólida, eficiente y organizada.

9.2.1. Herramientas y entorno

El proyecto se desarrolla en **React Native** [7], un framework para el desarrollo de aplicaciones, proporcionando controles de interfaz de usuario nativa y acceso completo a la plataforma nativa.

Para facilitar la configuración, construcción y despliegue se utiliza **Expo**, framework que facilita la creación, construcción y despliegue de la aplicación en React Native.

9.2.2. Entorno de pruebas

Expo Go se usa para ejecutar la aplicación directamente en dispositivos móviles para pruebas en tiempo real.

9.2.3. Librerías

Se ha usado **Expo Router** [8] para la navegación entre pantallas, lo que permite transiciones fluidas y estructuradas.

Para la comunicación entre el frontend y la API, se utiliza **Axios**, una librería que facilita la realización de solicitudes HTTP al backend. Axios permite realizar llamadas a los endpoints de manera rápida y sencilla.

Para permitir la persistencia de datos de manera offline se ha usado **AsyncStorage**, que proporciona almacenamiento local en el dispositivo, permitiendo guardar información relevante de los usuarios.

Para la integración de iconos en la interfaz de usuario se utiliza la librería **Tabler Icons** [10]. Esta librería ofrece una amplia gama de iconos vectoriales personalizables, optimizados para React Native.

9.2.4. Gestión de usuarios

Para la gestión de usuarios y autenticación, se ha utilizado **Firebase Authentication** [11], un servicio externo que proporciona una solución segura para el registro, inicio de sesión y manejo de credenciales. Aunque la integración de este servicio se realiza desde el frontend, Firebase Authentication actúa como una extensión del backend al manejar la lógica de autenticación y almacenar de forma segura los correos electrónicos y contraseñas de los usuarios.

10 RESULTADO

Pantalla de Inicio de Sesión y Registro: Permite a los usuarios registrarse o iniciar sesión en la aplicación mediante Firebase Authentication. El usuario debe ingresar un nombre único, correo, contraseña e indicar su rol. Esta pantalla inicial asegura que los usuarios puedan acceder a sus datos de manera segura.

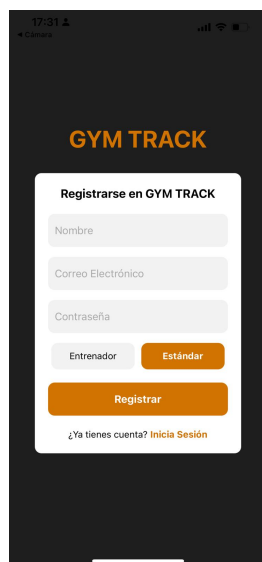


Fig. 4: Pantalla Registro

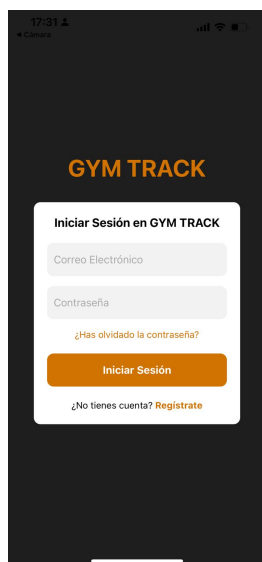


Fig. 5: Pantalla Inicio

Pantalla Ejercicios: Muestra un catálogo de un total de 17 ejercicios. El usuario tiene la opción de filtrar por nombre o por categoría (grupo muscular). Al seleccionar un ejercicio, se abre una pantalla de detalles que muestra la información completa del ejercicio, permitiendo a los usuarios comprender mejor cómo realizarlo correctamente.

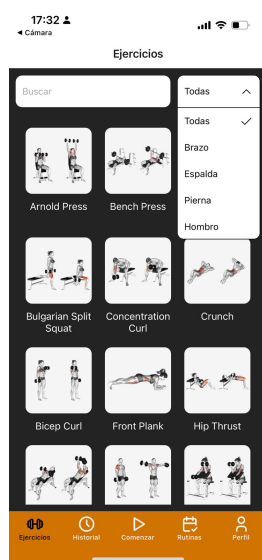


Fig. 6: Pantalla Ejercicios



Fig. 7: Pantalla Detalles

Pantalla de Perfil: Permite a los usuarios visualizar sus datos personales. Además, incluye una opción para cerrar sesión, asegurando una desconexión segura de la aplicación.

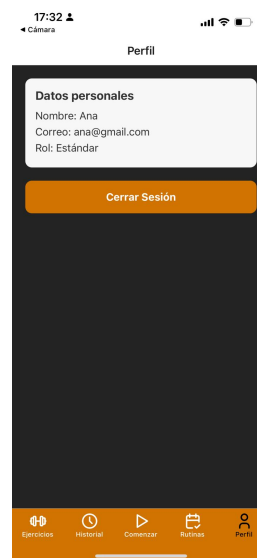


Fig. 8: Pantalla Perfil

Pantalla Rutinas: Esta pantalla principal incluye un tab que permite al usuario alternar entre dos subpantallas: **Gestión de entrenamientos** y **Gestión de planificaciones**, facilitando la navegación y organización de estas funcionalidades.

Gestión de entrenamientos: El usuario puede visualizar el listado de los entrenamientos. En esta pantalla el usuario puede:

- **Crear un entrenamiento:** Seleccionando el botón “Añadir entrenamiento”, será redirigido a la pantalla de creación de entrenamientos, donde especificará el nombre, descripción y ejercicios con sus respectivas series, repeticiones y peso.
- **Actualizar un entrenamiento:** Seleccionando el ícono del lápiz, el usuario es redirigido a la pantalla de creación con los datos preestablecidos del entrenamiento seleccionado.
- **Eliminar un entrenamiento:** Seleccionando el ícono de la papelera, se mostrará un pop-up donde podrá confirmar la eliminación.
- **Compartir un entrenamiento:** Si el usuario tiene el rol de *Entrenador*, tendrá disponible el icono de compartir. Seleccionándolo, se mostrará un pop-up donde introducirá el nombre del usuario destino.

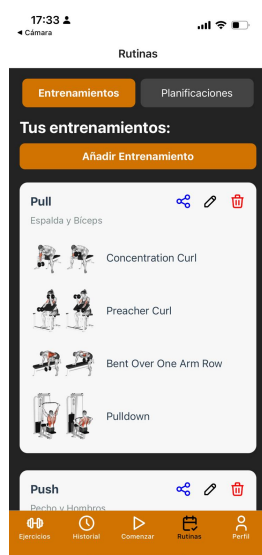


Fig. 9: Pantalla Entrenamientos

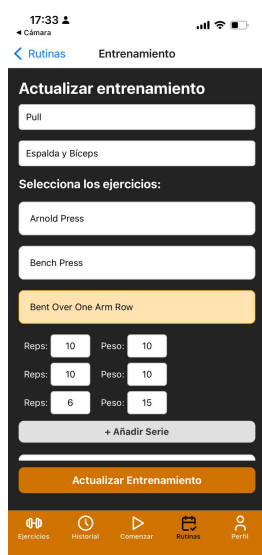


Fig. 10: Pantalla Actualización



Fig. 12: Pantalla Planificaciones

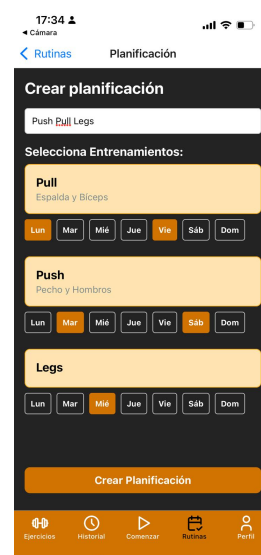


Fig. 13: Pantalla Crear Planificación

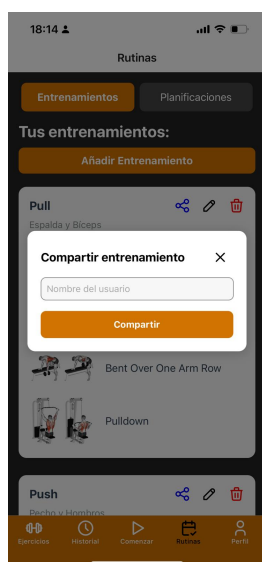


Fig. 11: Pantalla Compartir

Pantalla Comenzar: Esta es la pantalla inicial que el usuario verá al iniciar sesión tras registrarse. Su funcionalidad varía según el estado de las planificaciones:

- **Sin planificación activa:** Si el usuario no tiene ninguna planificación creada, por tanto ninguna activa, la pantalla mostrará un botón que redirige a la pantalla de *Rutinas*. Desde allí, el usuario podrá crear entrenamientos y planificaciones para empezar a utilizar la aplicación.
- **Con planificación activa:** Si el usuario tiene una planificación activa, se mostrará un listado de los entrenamientos incluidos en ella, ordenados según el día en el que estemos. Esto le permitirá iniciar el entrenamiento que desee de forma manual.

Gestión de planificaciones: El usuario puede visualizar el listado de las planificaciones. En esta pantalla el usuario puede:

- **Crear una planificación:** Seleccionando el botón “Añadir Planificación”, será redirigido a la pantalla de creación de planificaciones, donde especificará el nombre y seleccionará los entrenamientos previamente creados, asignándoles los días de la semana en los que se realizarán.
- **Eliminar una planificación:** Seleccionando el ícono de la papelera, se mostrará un pop-up donde podrá confirmar la eliminación.
- **Activar una planificación:** Seleccionando el botón de “Activar”, la planificación elegida pasará a estar activa.
- **Compartir una planificación:** Si el usuario tiene el rol de *Entrenador*, tendrá disponible el icono de compartir. Seleccionándolo, se mostrará un pop-up donde introducirá el nombre del usuario destino.

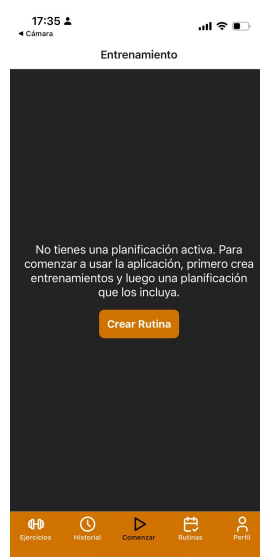


Fig. 14: Pantalla Comenzar sin Planificación

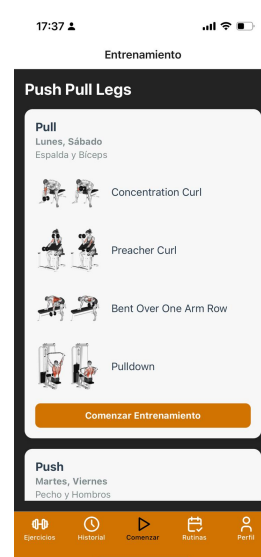


Fig. 15: Pantalla Comenzar con Planificación

Gestión de sesión de entrenamiento: Desde la pantalla de *Comenzar*, al seleccionar el botón “Comenzar entrenamiento”, el usuario será redirigido a la pantalla de *Sesión de Entrenamiento*. En esta pantalla, se mostrará los ejercicios del entrenamiento seleccionado, junto con sus series, repeticiones y pesos previamente establecidos. Mediante las flechas podremos ir cambiando de ejercicio. Además, la pantalla permitirá al usuario modificar estos valores en tiempo real, adaptándolos a sus necesidades o circunstancias durante la sesión.

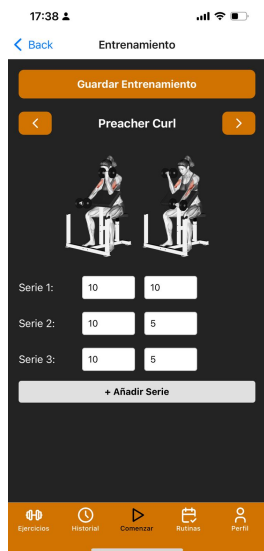


Fig. 16: Pantalla Sesión Entrenamiento

Pantalla Historial: Se muestra un listado de todos los entrenamientos finalizados, junto con sus detalles, ordenados por fecha del más reciente al más antiguo.

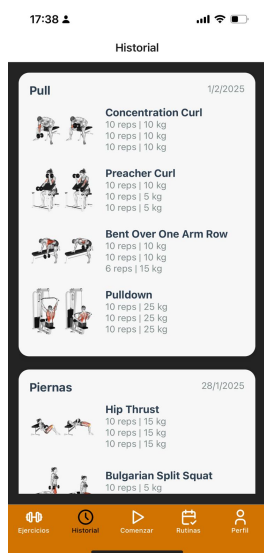


Fig. 17: Pantalla Historial

Panel de administración: El usuario “Administrador” tiene acceso a una pantalla exclusiva donde puede gestionar los usuarios registrados. Desde esta pantalla, podrá eliminar cuentas en caso de que se detecte un mal uso de la aplicación.

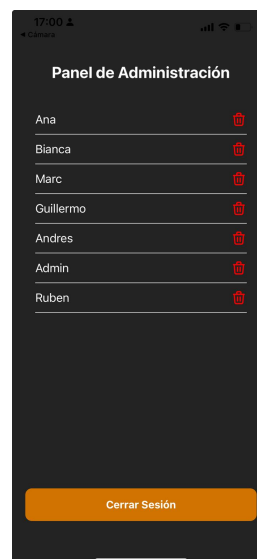


Fig. 18: Pantalla Panel de Administración

11 TESTS Y VALIDACIONES

11.1. Backend

Durante el desarrollo del backend, se realizaron diversas pruebas para asegurar el correcto funcionamiento de la API, la integridad de los datos y una gestión adecuada de errores.

Pruebas de endpoints de la API: Se llevaron a cabo pruebas utilizando **Postman** para garantizar que cada endpoint respondiera correctamente ante solicitudes válidas e inválidas.

Mensajes informativos: Se aseguró que los endpoints devolvieran mensajes claros y específicos tanto en casos de éxito como de error.

Validación de datos: Se aseguró que los endpoints exigieran todos los valores necesarios para procesar las solicitudes de forma correcta. Algunas validaciones realizadas:

- Verifica que los campos nombre, creador y un array de ejercicios sean obligatorios para realizar la creación de un entrenamiento.
- Verifica que los campos nombre, creador y un array de ejercicios sean obligatorios para realizar la creación de una planificación.
- Verifica que el campo creador es obligatorio para activar una planificación, ya que se necesita para desactivar las planificaciones existentes del mismo usuario.

Uso de logs de depuración: Para resolver problemas detectados, fue necesario crear logs en algunos endpoints, registrando los datos recibidos en las solicitudes. Estos logs permitieron identificar errores en la estructura de los datos enviados.

Manejo de errores mediante catch: Se incluyeron bloques try-catch para capturar y gestionar errores.

11.2. Frontend

Se realizaron pruebas en el frontend para asegurar la correcta interacción y la validación de los datos en la interfaz de usuario.

Validaciones de formularios: Se realizan algunas validaciones de los datos obligatorios en los formularios antes de enviar las solicitudes al Backend para asegurar que los datos enviados sean completos y consistentes.

Para cada una de estas validaciones, la aplicación muestra una alerta con el mensaje de error claro e informativo, evitando que el usuario continúe hasta que los datos ingresados cumplan con los requisitos necesarios.

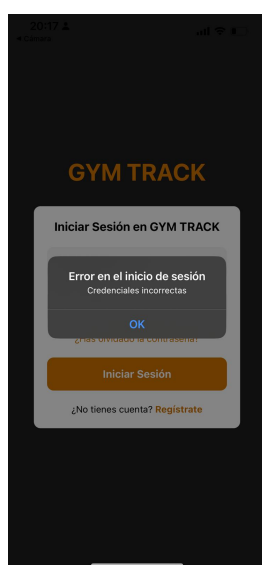


Fig. 19: Pantalla Error

Manejo de errores mediante catch: Se incluyeron bloques try-catch en los controladores del frontend para capturar y gestionar errores generados durante la interacción con la API y el procesamiento de datos.

Pruebas con usuarios: Durante todo el desarrollo, varios usuarios probaron la aplicación, permitiendo identificar áreas de mejora y errores que no habían surgido en las pruebas iniciales.

12 VALORACIÓN ECONÓMICA

La valoración económica de este proyecto se han realizado teniendo en cuenta los siguientes aspectos: los roles y horas de trabajo estimadas, el coste de licencias de software y hardware necesarios.

Coste de roles y horas de trabajo: Este cálculo se basa en una carga de trabajo total de 300 horas y utilizando los salarios medios para puestos Junior en cada rol.

Rol	Horas	€/h	Coste
Diseñador de UI/UX	30h	12€/h	360€
Desarrollador Frontend	130h	12€/h	1560€
Desarrollador Backend	60h	13,5€/h	810€
Project Manager	45h	14,5€/h	652,5€
Tester y QA	35h	12€/h	420€
Total:	300h		3802,5€

Fig. 20: Coste estimado

Coste de software y licencias: El proyecto utiliza herramientas que ofrecen planes gratuitos sin limitaciones relevantes para el alcance de este trabajo, como GitHub, Jira, Firebase, MongoDB Atlas, Figma, VSCode... Así, el coste de software y licencias es de 0€.

Coste de hardware: El desarrollo se realiza en un portátil valorado en 1000€, y para pruebas ahora se utiliza un dispositivo móvil iOS valorado en 800 €, resultando en un total de 1800€ en hardware.

Coste de derechos de imágenes [12]: Se han pagado 12,75 € por los derechos de las imágenes utilizadas en los ejercicios de este proyecto.

Coste real del proyecto: Si únicamente se consideran los gastos reales efectuados (derechos de imágenes), el coste total es de 12,75 €.

Coste total estimado: $3802,5€ + 0€ + 1800€ + 12,75 = 5615,25€$.

13 CONCLUSIONES

La aplicación para el registro y seguimiento de actividades deportivas en gimnasios ha cumplido los objetivos planteados y se ha conseguido crear una herramienta completa y funcional.

Aunque la aplicación ya cuenta con todas las funcionalidades esenciales, hay algunas ideas que han quedado pendientes para futuras versiones. Por ejemplo, la incorporación de estadísticas y gráficos, así como la posibilidad de realizar un seguimiento de datos de salud para ofrecer un análisis más detallado del progreso del usuario.

A nivel personal, este proyecto ha sido una gran oportunidad para aprender más sobre el desarrollo de aplicaciones móviles y mejorar la gestión y organización de proyectos aplicando metodologías ágiles.

En definitiva, la aplicación cumple con los objetivos iniciales y ofrece una solución efectiva para quienes quieren llevar un control detallado de sus entrenamientos en el gimnasio. A partir de aquí, hay muchas posibilidades de seguir mejorándola y ampliando sus funciones para que se adapte aún más a las necesidades de los usuarios.

14 AGRADECIMIENTOS

Agradecer primeramente a mi tutor, Marc Talló, por su guía durante todo este proceso. Gracias por resolver mis dudas y por ayudarme a sacar adelante este proyecto.

También quiero dar las gracias a mi familia por estar siempre apoyándome. En especial a Bianca, mi cuñada, por su ayuda con todo lo relacionado con el diseño. Sus consejos han sido clave para darle una identidad a la aplicación.

A mi pareja, Guille, y a mis amigos, gracias por vuestra paciencia infinita, por escucharme hablar sobre este proyecto (a veces más de la cuenta) y por hacer que todo este proceso haya sido mucho más llevadero.

REFERENCIAS

- [1] Lázaro, E. C. “Informe OBS: La industria del deporte y el fitness”. <https://www.obsbusiness.school/>. Accessed: Sept. 15, 2024. [Online.] Available: <https://marketing.onlinebschool.es/Prensa/Informes/Informe%20OBS%20Industria%20del%20deporte%20y%20el%20fitness.pdf>
- [2] Andújar, Juan A. “Los tres modelos de monetización más utilizados en los negocios digitales?”. <https://www.sage.com/es-es/>. [Online.] Available: <https://www.sage.com/es-es/blog/los-tres-modelos-de-monetizacion-mas-utilizados-en-los-negocios-digitales/>
- [3] Martins, J. “¿Qué es la metodología Kanban y cómo funciona?”. <https://asana.com/es>. Accessed: Sept. 24, 2024. [Online.] Available: <https://asana.com/es/resources/what-is-kanban>
- [4] Martins, J. “MoSCoW. ¿Qué es y cómo priorizar en el desarrollo de tu aplicación?”. <https://www.itdo.com/blog/>. Accessed: Sept. 26, 2024. [Online.] Available: <https://www.itdo.com/blog/moscow-que-es-y-como-priorizar-en-el-desarrollo-de-tu-aplicacion/>
- [5] MonkeyWit. API REST, Node js & MongoDB Atlas, Express — CRUD. (19 Sept 2021). Accessed: Oct 18, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=S4IgPTwwPBw&list=LL&index=21&t=2915s>
- [6] MonkeyWit. Despliegue de NODE JS en RENDER — Aplicación de API REST con Express. (12 Sept 2023). Accessed: Oct 24, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=Ck90-QhSnNo&list=LL&index=8&t=603s>
- [7] <https://reactnative.dev/>. [Online.] Available: <https://reactnative.dev/docs/getting-started>
- [8] “Create pages with Expo Router”. <https://docs.expo.dev/>. Accessed: Oct. 15, 2024. [Online.] Available: <https://docs.expo.dev/router/create-pages/>
- [9] “Tabs”. <https://docs.expo.dev/>. Accessed: Oct. 15, 2024. [Online.] Available: <https://docs.expo.dev/router/advanced/tabs/>
- [10] “5790 pixel-perfect icons for web design”. <https://tabler.io/admin-template>. Accessed: Oct. 15, 2024. [Online.] Available: <https://tabler.io/icons>
- [11] Grimm, S. “React Native Firebase Authentication with Expo Router?”. <https://galaxies.dev/>. Accessed: Nov. 7, 2024. [Online.] Available: <https://galaxies.dev/react-native-firebase-authentication-expo-router>
- [12] “Gym Visuals”. <https://gymvisual.com/>. Accessed: Jan. 8, 2025. [Online.] Available: <https://gymvisual.com/>

A.1. CRONOGRAMA DE JIRA SOFTWARE



A.2. REQUISITOS

REQUISITOS FUNCIONALES		
ID	DESCRIPCIÓN	PRIORIDAD
RF-01	El sistema debe diferenciar 3 tipos de usuarios: usuario estándar, usuario entrenador y usuario administrador.	Must
RF-02	El sistema debe pedir a los usuarios registrarse proporcionando su nombre, correo electrónico, contraseña y selección de uno de los siguientes roles: usuario estándar o usuario entrenador, para poder iniciar sesión posteriormente.	Must
RF-03	El sistema debe pedir a los usuarios iniciar sesión para acceder a su cuenta utilizando sus credenciales (correo electrónico y contraseña).	Must
RF-04	El sistema debe permitir al usuario administrador la visualización de los usuarios registrados.	Must
RF-05	El sistema debe permitir al usuario administrador la eliminación de otro usuario.	Must
RF-06	El sistema debe permitir a los usuarios estándar y entrenador la recuperación de su contraseña mediante el envío de un enlace de restablecimiento al correo electrónico.	Could
RF-07	El sistema debe permitir a los usuarios estándar y entrenador visualizar y editar su información personal, como nombre, foto de perfil, peso y medidas corporales.	Could
RF-08	El sistema debe permitir a los usuarios estándar y entrenador visualizar un catálogo de ejercicios de gimnasio.	Must
RF-09	El sistema debe permitir a los usuarios estándar y entrenador filtrar un ejercicio por nombre o categoría.	Must
RF-10	El sistema debe proporcionar una breve descripción, imágenes o videos demostrativos para cada ejercicio.	Must
RF-11	El sistema debe proporcionar para cada ejercicio, el historial del usuario mediante estadísticas o gráficos.	Could
RF-12	El sistema debe permitir a los usuarios estándar y entrenador crear entrenamientos especificando el nombre, descripción y los ejercicios que la componen asignando valores predeterminados como series, repeticiones y peso.	Must
RF-13	El sistema debe permitir a los usuarios estándar y entrenador eliminar los entrenamientos.	Must

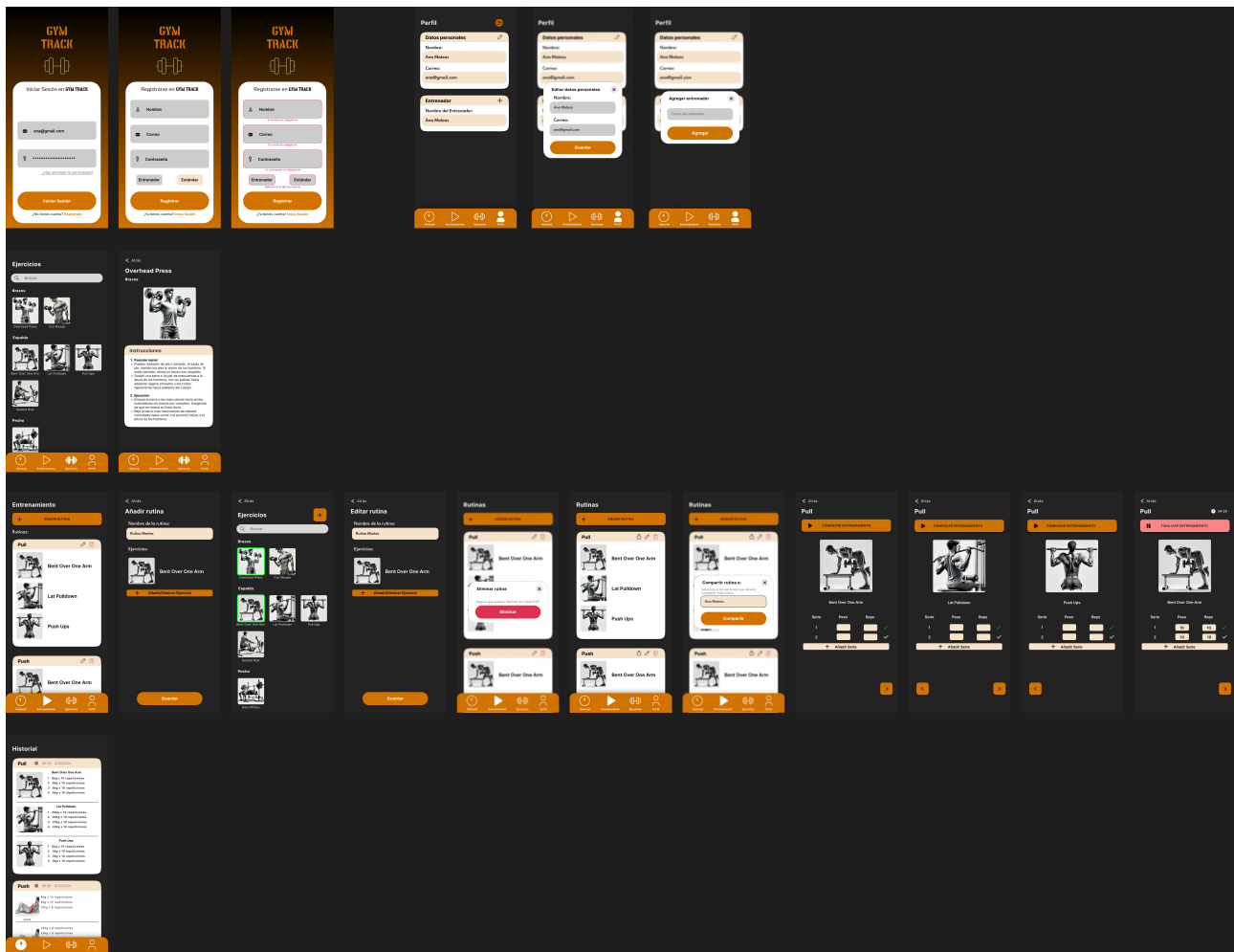
RF-14	El sistema debe permitir a los usuarios estándar y entrenador modificar los datos de los entrenamientos previamente creados.	Must
RF-15	El sistema debe permitir a los usuarios entrenador compartir un entrenamiento o planificación específica a otro usuario estándar.	Should
RF-16	El sistema debe permitir a los usuarios entrenador solicitar la vinculación con un usuario estándar para poder compartir rutinas.	Should
RF-17	El sistema debe permitir a los usuarios estándar solicitar la vinculación con un usuario entrenador para recibir las rutinas compartidas.	Should
RF-18	El sistema debe permitir a los usuarios estándar y entrenador iniciar el seguimiento de una sesión de entrenamiento asociada a un entrenamiento.	Must
RF-19	Al iniciar el seguimiento de un entrenamiento, el sistema debe inicializar un cronómetro para calcular el tiempo transcurrido en realizar la rutina.	Could
RF-20	Durante el seguimiento de un entrenamiento, el sistema debe permitir al usuario registrar detalles como series, repeticiones y peso utilizado por cada serie.	Must
RF-21	El sistema debe permitir a los usuarios visualizar un historial los entrenamientos realizados, mostrando el entrenamiento correspondiente con la fecha de cuando se realizó y los detalles que proporcionaron.	Must
RF-22	El sistema debe permitir a los usuarios estándar y entrenadores solicitar la vinculación con un usuario estándar para agregarlos a su círculo de amistades.	Could
RF-23	El sistema debe permitir a los usuarios visualizar los entrenamientos de su círculo de amistades.	Could
RF-24	El sistema debe permitir la interacción entre usuarios mediante comentarios o mensajes sobre los entrenamientos.	Could
RF-25	El sistema debe permitir a los usuarios estándar y entrenador la creación de planificaciones especificando el nombre y los entrenamientos que la incluyen, asignándoles días de la semana.	Must
RF-26	El sistema debe permitir a los usuarios estándar y entrenador la activación de una planificación.	Should
RF-27	El sistema debe permitir a los usuarios estándar y entrenador la eliminación y modificación de planificaciones.	Must

REQUISITOS NO FUNCIONALES

ID	DESCRIPCIÓN	PRIORIDAD
RNF-01	El sistema debe validar que el campo de correo electrónico de registro siga el formato xxx@yyy.zz.	Must
RNF-02	El sistema debe incluir mensajes de error cuando los usuarios ingresen información incorrecta o incompleta.	Should
RNF-03	El sistema debe establecer todos los campos de inicio de sesión/registro como obligatorios.	Must
RNF-04	El sistema debe solicitar el peso del usuario en formato kilogramo (kg).	Should
RNF-05	El sistema debe establecer un límite de 3 dígitos para el campo de peso.	Should
RNF-06	El sistema debe solicitar las medidas corporales del usuario en formato centímetros (cm).	Should
RNF-07	Los datos personales de los usuarios, como el correo electrónico y la contraseña, no deben compartirse con terceros ni ser visibles para otros usuarios.	Must
RNF-08	El código debe estar organizado y documentado de manera que cualquier desarrollador pueda entender la estructura básica de la aplicación.	Must
RNF-09	La aplicación debe ser fácil de usar, permitiendo a los usuarios navegar por las principales funcionalidades en pocos clics.	Must
RNF-10	El sistema debe validar que el campo nombre único sea exclusivo.	Must
RNF-11	El sistema debe solicitar el nombre único para la vinculación entre usuarios.	Could
RNF-12	El sistema debe permitir únicamente una planificación activa a la vez.	Must

REQUISITOS TÉCNICOS		
ID	DESCRIPCIÓN	PRIORIDAD
RT-01	La aplicación debe funcionar en dispositivos móviles con Android.	Must

A.3. PROTOTIPO



A.4. DIAGRAMA DE CASOS DE USO

