



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

PowerLog Móvil. Una aplicación móvil para la gestión de  
entrenamientos con componentes sociales para atletas y  
entrenadores

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Magraner Tamarit, Pablo

Tutor/a: Valderas Aranda, Pedro José

CURSO ACADÉMICO: 2022/2023



# Resumen

A lo largo de esta memoria va a quedar documentado todo el proceso de creación de una aplicación móvil conectada con un backend que forma parte de un entorno que se completa con la aplicación web. La idea surge de dos amigos que al tener conversaciones con gente del mundo del entrenamiento en el gimnasio nos percatamos de la falta de un sistema que de manera síncrona gestione los entrenamientos de atletas. En este documento se presentan todos los pasos que se han seguido para llevar a cabo la idea, desde la motivación inicial, pasando por la fase de diseño, fase de implantación y despliegue y fase de pruebas con usuarios reales y los resultados obtenidos después de varios meses de uso de la aplicación, por último proyectos futuros e ideas para continuar mejorando la aplicación junto con una parte de conclusiones de todo el proceso de creación de la aplicación.

**Palabras clave:** PowerLog, Flutter, Desarrollo Centrado en el Usuario, React, Digital Ocean, Vercel, HTTP, HTTPS, MongoDB, PowerLifting, Deporte

---

# Abstract

"Throughout this report, the entire process of creating a mobile application connected to a backend, which is part of an environment completed by the web application. The idea originated from two friends who, through conversations with people in the gym training world, noticed the lack of a system that synchronously manages athletes' workouts. This document presents all the steps that have been followed to carry out the idea, from the initial motivation, through the design phase, implementation and deployment phase, testing phase with real users, and the results obtained after several months of using the application. Finally, it includes future projects and ideas for further improving the application, along with a conclusion section summarizing the entire application creation process."

**Key words:** PowerLog, Flutter, User Centered Design, React, Digital Ocean, Vercel, HTTP, HTTPS, MongoDB, PowerLifting, Sport

---

# Índice general

---

<b>Índice general</b>	<b>III</b>
<b>Índice de figuras</b>	<b>V</b>
<b>Índice de Listados</b>	<b>VI</b>
<hr/>	
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación . . . . .	1
1.2 Objetivos . . . . .	2
1.3 Impacto esperado . . . . .	2
1.4 Metodología . . . . .	2
1.5 Estructura . . . . .	4
1.6 Colaboraciones . . . . .	4
<b>2 Estado del Arte</b>	<b>6</b>
2.1 Strava . . . . .	6
2.2 Hevy . . . . .	7
2.3 Virtuagym Fitness . . . . .	9
2.4 Crítica al estado del arte . . . . .	10
2.4.1 Crítica a Strava . . . . .	10
2.4.2 Crítica a Hevy . . . . .	10
2.4.3 Crítica a Virtuagym Fitness . . . . .	10
2.5 Propuesta . . . . .	11
<b>3 Análisis del problema</b>	<b>12</b>
3.1 Especificación de requisitos . . . . .	12
3.1.1 Requisitos funcionales . . . . .	12
3.1.2 Requisitos no funcionales . . . . .	13
3.2 Casos de uso y diagrama de clases . . . . .	14
3.2.1 Casos de uso . . . . .	14
3.2.2 Diagramas de clases . . . . .	15
3.3 Seguridad . . . . .	16
3.4 Análisis energético o de eficiencia algorítmica . . . . .	17
3.4.1 Comunicación en red . . . . .	17
3.5 Análisis del marco legal y ético . . . . .	18
3.5.1 Análisis de la protección de datos . . . . .	18
3.5.2 Propiedad intelectual . . . . .	18
3.5.3 Marco ético . . . . .	19
3.6 Análisis de riesgos . . . . .	19
3.7 Análisis de posibles soluciones . . . . .	20
3.8 Solución propuesta . . . . .	21
3.9 Plan de trabajo . . . . .	21
<b>4 Diseño</b>	<b>22</b>
4.1 Estructura de documentos . . . . .	22
4.2 Diseño de las interfaces . . . . .	24
<b>5 Desarrollo</b>	<b>27</b>

5.1	Arquitectura del sistema . . . . .	27
5.2	Tecnología utilizada . . . . .	28
5.3	Implementación y ejemplos de código . . . . .	30
5.4	Implantación . . . . .	37
<b>6</b>	<b>Producto desarrollado</b>	<b>38</b>
<b>7</b>	<b>Conclusiones</b>	<b>48</b>
<b>8</b>	<b>Relación del trabajo desarrollado con las asignaturas cursadas</b>	<b>50</b>
<b>9</b>	<b>Trabajos Futuros</b>	<b>51</b>
	<b>Bibliografía</b>	<b>52</b>
<b>A</b>	<b>Anexo: Análisis de ODS</b>	<b>54</b>

# Índice de figuras

---

2.1	Gráficas en Strava. . . . .	7
2.2	App de Hevy. . . . .	8
2.3	App de Virtuagym. . . . .	9
3.1	Diagrama de casos de uso. . . . .	14
3.2	Diagrama UML. . . . .	15
4.1	Documento del DayLog . . . . .	22
4.2	Documento del Entrenamiento . . . . .	23
4.3	Documento de las Constraints . . . . .	23
4.4	Documento del ExerciseList . . . . .	24
4.5	Boceto de la aplicación móvil . . . . .	25
5.1	Arquitectura del sistema. . . . .	28
5.2	Patrón bloc . . . . .	30
5.3	Estructura de directorios en /lib . . . . .	31
5.4	Estructura de directorios en /packages . . . . .	32
5.5	Ejemplo de la arquitectura hexagonal al completo. . . . .	32
5.6	Ejemplo DTO de un entrenamiento . . . . .	33
5.7	Ejemplo repositorio de un entrenamiento. . . . .	33
5.8	Ejemplo controlador para actualizar los datos de un entrenamiento . . . . .	34
5.9	Ejemplo estructura de directorios para las rutinas de entrenamiento. . . . .	34
5.10	Login y LogOut de la app móvil mediante el cliente de Google Firebase. . . . .	35
5.11	Actualizar un entrenamiento desde la app móvil. . . . .	35
5.12	Consultar un entrenamiento desde la app móvil. . . . .	36
5.13	Devolver el entrenamiento asignado a un día desde el backend. . . . .	36
6.1	Pantalla de carga de PowerLog . . . . .	38
6.2	Pantalla de inicio de sesión. . . . .	39
6.3	Pantalla de registro. . . . .	39
6.4	Pantalla de calendario con entrenamiento disponible. . . . .	40
6.5	Pantalla de calendario sin entrenamiento. . . . .	40
6.6	Pantalla de resumen de un entrenamiento. . . . .	41
6.7	Pantalla de mi cuenta. . . . .	42
6.8	Pantalla de detalles con el botón activo. . . . .	43
6.9	Pantalla de detalles con el botón inactivo. . . . .	43
6.10	Diálogo de aviso de cerrar sesión. . . . .	44
6.11	Pantalla del listado de amigos. . . . .	45
6.12	Pantalla con el listado de todos los usuarios. . . . .	46
6.13	Ejemplo de uso del buscador. . . . .	46
6.14	Entrenamientos totales registrados en PowerLog. . . . .	46
6.15	Atleta que más entrenamientos ha modificado. . . . .	47

# Índice de Listados

---

---

---

# CAPÍTULO 1

## Introducción

---

En la actualidad la forma en la que realizamos distintas actividades cotidianas ha cambiado debido a los rápidos avances de la tecnología, para ser más concretos uno de los sectores que más ha avanzado en los últimos años son los dispositivos móviles. En la actualidad existen aproximadamente 15 mil millones de dispositivos, casi el doble de la población mundial, y no cabe duda que se ha vuelto imprescindible en nuestro día a día.

Partiendo de esta base, el mundo del deporte no queda exento de este avance de la tecnología y por tanto surgen aplicaciones móviles con la finalidad de facilitar a los usuarios la práctica de dicho deporte. El mundo del fitness y la actividad deportiva con pesas no es una excepción, un mundo que está en constante expansión pero todavía lejos de ser uno de los deportes más practicados del mundo.

Dentro del entrenamiento con pesas cabe destacar que existen distintas modalidades dependiendo de la decisión del usuario y pueden ser dos principalmente, la primera de ellas es el fitness, donde el usuario entrena para conseguir un cuerpo saludable y generalmente más estético. La segunda opción es la que se aborda en esta aplicación y es el powerlifting, una disciplina donde se entrena para levantar la mayor cantidad de peso posible dentro de los tres movimientos principales de competición: Sentadilla, press de banca y peso muerto, se dividen dentro de numerosas categorías agrupadas por el peso corporal del atleta.

En este contexto surge esta aplicación, una manera efectiva de gestionar la comunicación entre el atleta y el entrenador de manera que se pueden realizar planificaciones más precisas y con un registro de todos los entrenamientos de una manera sencilla y accesible por ambas partes.

### 1.1 Motivación

---

Lo que me llevó a seleccionar esta temática concretamente fueron diferentes situaciones que se llevaron a cabo, la primera de ellas y principal es que yo estaba comenzando con la programación en dispositivos móviles multiplataforma y estaba cansado de realizar tutoriales y aplicaciones sencillas, por tanto pensé que embarcarme en un proyecto más serio y presentarlo como trabajo de fin de grado era una idea fantástica para ponerme a prueba con todo lo aprendido y ver hasta donde era capaz llegar. La segunda situación que se cumplió es que yo he practicado deporte toda mi vida, en un inicio baloncesto y cuando dejé de jugar me apunté al gimnasio y por tanto siempre ha sido una temática que me ha fascinado.

Por último otra motivación que tenía es que era gran oportunidad de sacar la aplicación a mercado y tratar de conseguir las máximas descargas posibles debido a que el

powerlifting es un nicho muy pequeño y que está actualmente en auge pero no existe ninguna aplicación similar que permita una comunicación entre atleta y usuario.

## 1.2 Objetivos

---

El objetivo de esta aplicación es satisfacer una necesidad principal que surge y es mejorar la comunicación entre atletas y entrenadores. Comunicándome con diferentes entrenadores todos comentaban el mismo problema y era que la manera de gestionar los entrenamientos era muy poco organizada y en el caso de tener un gran número de atletas llegaba a ser confuso.

- El primer objetivo de la aplicación es que los atletas puedan crear y gestionar una cuenta para mantener almacenados sus entrenamientos.
- El segundo objetivo es que el atleta tenga acceso a los entrenamientos planteados por su entrenador y pueda completarlos con los resultados de su sesión junto con una serie de comentarios.
- El tercer objetivo es un componente social en el que los atletas puedan agregar usuarios a su lista de amigos.
- El cuarto objetivo es que la aplicación sea sencilla de utilizar para los usuarios y que al ser para dispositivos móviles, el usuario desde el mismo lugar de entrenamiento puede ir introduciendo sus resultados.

## 1.3 Impacto esperado

---

Actualmente la metodología que siguen los atletas consiste en apuntar en un papel de cualquier manera los resultados del entrenamiento planteado por su entrenador y al llegar a casa transcribirlos a un documento de excel compartido. El impacto que se busca es doble, en primer lugar facilitar al atleta escribir dichos resultados desde la app en el momento que realiza el movimiento y también se busca un impacto medioambiental tratando de evitar el uso de libretas de papel.

Por supuesto, efectos colaterales de que las aplicaciones de entrenamientos se popularicen es que se da muchísima visibilidad al deporte lo cual beneficia a todas las personas que quieran realizar un cambio en su vida y comenzar un cambio físico.

## 1.4 Metodología

---

La metodología aplicada para llevar a cabo el cumplimiento de los objetivos es la del Diseño Centrado en el Usuario. Esta metodología consiste en darle un enfoque de diseño, al producto que vamos a elaborar, dirigido directamente a las personas que van a hacer uso de este producto.

En primer lugar se plantearon los objetivos de la aplicación y cuales eran las ideas principales, todo esto fue comentado con mi compañero de desarrollo pero nada documentado ya que eran simples ideas. Una vez encontrado el objetivo claro, el cual era elaborar una app que sincronizase a atletas y entrenadores de manera que fuera sencillo para ambos la comunicación y el desarrollo de los entrenamientos, a este objetivo principal se llegó

gracias a la comunicación con entrenadores que nos comunicaron que para ellos era muy tedioso mantener a una gran cantidad de atletas con las herramientas que tenían disponibles actualmente. Una vez planteados los objetivos se pasó a la fase de diseño donde aquí se optó por un desarrollo centrado en el usuario(DCU), ya que es algo que habíamos estudiado en varias asignaturas y consideramos que era la mejor opción. Para llevar a cabo un DCU donde el usuario es el principal protagonista y se encarga de ir validando los avances, el primer paso es realizar un análisis de las necesidades el cual consta de tres partes: investigación cualitativa, técnica personas y descripción de escenarios.

Para la investigación cualitativa se optó por el método de las entrevistas personales, debido a que conocíamos de primera mano a los entrenadores y atletas que eran el perfil para el que iba destinada la app se agendaron reuniones donde se comentaba cuales eran sus necesidades en la app, es decir que funcionalidades debería de tener para cumplir con el objetivo de los entrenadores y atletas. Una vez identificadas las necesidades de los usuarios se empleó la técnica personas para crear un modelo de entrenador y atleta que serían los perfiles de usuarios con los que cuenta la app. Y por último una vez realizados esos pasos se plantearon una serie de escenarios para tener una breve descripción de como estos perfiles de usuarios iban a utilizar la app.

Continuando con el desarrollo centrado en el usuario, lo siguiente es realizar un diseño y una evaluación de dicho diseño. En primer lugar una vez recopilada la información necesaria en el anterior apartado se realizaron prototipos de como iba a quedar la app con la finalidad de dar soporte a los escenarios planteados sin olvidar todos los aspectos clave en el diseño de la usabilidad, una vez finalizada con esta parte se agendaron reuniones con los futuros usuarios con la finalidad de evaluar el diseño y obtener un resultado final sobre el que comenzar la implementación.

Para terminar una vez cerrada la fase de diseño y análisis, la siguiente fase es la de implementación para lo que abarca este proyecto que es la parte de la aplicación móvil y el backend para la parte de los atletas. En primer lugar y siguiendo los prototipos planteados se elaboró un mínimo producto viable(MVP) donde los atletas eran capaces de crearse una cuenta y hacer login en caso de ya tener una cuenta existente y una pantalla con el calendario para consultar su entrenamiento y modificarlo, por la parte del back este MVP contaba con los endpoints necesarios para recuperar el entrenamiento dependiendo del día seleccionado en el calendario y para actualizar dicho entrenamiento, para ello primero probaron los usuarios que tenían dispositivos android ya que esta plataforma ofrece una gran facilidad para distribuir el .apk por cualquier plataforma de mensajería. Después de las primeras pruebas nos dieron feedback del uso de la aplicación el cual permitió resolver los bugs que fueron apareciendo y dejar pulida la primera versión funcional de la app. A continuación y mediante la cuenta de desarrollador de apple se distribuyó ente los usuarios de iOS lo cual sirvió para pulir las funcionalidades en esta plataforma ya que si bien es cierto que se utiliza el mismo código fuente ya que es una de las características que ofrece flutter[11] puede haber ligeras variaciones en las versiones debido a como se tratan los diferentes sistemas operativos las apps. Cuando el mvp estaba funcionando correctamente se fueron sacando actualizaciones, la primera de ellas para el módulo de mi cuenta, la segunda para añadir comentarios en los entrenamientos y la tercera para la funcionalidad de la lista de amigos. Todas estas actualizaciones contaban con un período de prueba con los usuarios que se consideraban "beta testers"para posteriormente distribuirlo a todos los usuarios de la app.

En resumen el desarrollo de la solución hasta llegar al proceso final ha sido un camino por el cual tanto desarrolladores como usuarios han formado parte con la finalidad de

conformar el entorno que mejor se adaptaba a las necesidades planteadas y de suplir las carencias que existían con el uso de otras apps o modelos de gestión de rutinas. A lo largo de este proceso los usuarios han sido los principales protagonistas ya que han influido en cada etapa del desarrollo asegurando de que la aplicación sea intuitiva y cumpla con sus expectativas.

## 1.5 Estructura

---

En este apartado se va a presentar de forma breve que aspectos se van a tratar y desarrollar a lo largo de esta memoria.

2. A continuación de la introducción se presentará el apartado del estado del arte donde se comentarán las aplicaciones que más se asemejan a la planteada en este proyecto, tanto aspectos positivos como negativos de éstas y una propuesta donde se plantea la idea que se va a desarrollar.
3. En el siguiente apartado se analiza el problema de diferentes formas, especificación de requisitos, casos de uso, diagramas UML etc... y por último se presentan las posibles soluciones y finalmente la elegida.
4. A continuación del análisis se plantea el apartado de diseño donde se muestran cuales son las estructuras de documentos empleadas en la base de datos así como un boceto inicial previo a la fase de desarrollo.
5. Una vez presentado la fase de diseño se plantea la fase de desarrollo, donde se describe cual es la arquitectura del sistema, que tecnologías se han utilizado y finalmente como ha sido la implantación y la implementación.
6. En el apartado de producto desarrollado se va a presentar el resultado final de la aplicación móvil que utilizan los atletas además de varias consultas a la base de datos de Mongo donde se puede comprobar ciertos aspectos relacionados con el uso que le han dado los usuarios.
7. En el apartado de conclusiones se describe cuáles han sido las sensaciones y valoraciones finales de todo el proceso de desarrollo de la aplicación.
8. En el apartado de relación del trabajo desarrollado con las asignaturas cursadas se presentan cuales han sido las asignaturas de la carrera que más han influenciado en mi opinión al desarrollo del proyecto y por que.
9. En el último apartado se comentan cuales son las futuras ideas para seguir trabajando en la aplicación.

## 1.6 Colaboraciones

---

Esta idea no solo se lleva a cabo con una aplicación móvil, que es lo que se va a desarrollar a lo largo de la memoria, si no que también consta de una aplicación web donde el entrenador planifica los entrenamientos que el atleta puede visualizar en su dispositivo.

He colaborado junto con Manuel Emilio Eljishi Calzada en su proyecto "PowerLog Web. Plataforma web para la gestión de entrenamientos" en la realización de esta idea, mi compañero se encarga de crear los endpoints necesarios en el backend donde poder

---

realizar las peticiones por el lado de los entrenadores y el desarrollo de una web para que los entrenadores puedan crear rutinas. Esto está directamente sincronizado con la parte que abordo yo y consiste en la visualización de dichas rutinas creadas y la posibilidad de enviar un feedback el cual se ve reflejado en la web de los entrenadores además de los endpoints necesarios en el backend para que los atletas puedan visualizar y modificar sus rutinas.

---

---

## CAPÍTULO 2

# Estado del Arte

---

En esta sección se abordarán las diferentes aplicaciones que actualmente están destacando en el ambiente fitness y de la creación de entrenamientos. A lo largo de este apartado se van a analizar 3 aplicaciones las cuales desde mi punto de vista son las que mejor cumplen la función de gestión de entrenamientos, también realizará un análisis sobre cuales son las diferencias de dichas aplicaciones con PowerLog y un crítica hacia ellas.

### 2.1 Strava

---

Strava[1] es una aplicación que no está destinada tanto al ámbito fitness si no más a cualquier tipo de deporte pero cuenta con una gran cantidad de funcionalidades debido al tiempo que llevan en el mercado y que la conforman un equipo muy grande de trabajadores. La fecha de lanzamiento de Strava[1] fue el 18 de Agosto de 2009 llegando a las 50 actividades registradas en el día de salida.

Desde mi punto de vista me parece una aplicación puntera en cuanto a la gestión de rutinas y la cantidad de personalización que permite al usuario a la hora de rellenar las sensaciones de los entrenamientos y creación de rutinas personalizadas. Por ese motivo me gustaría analizar Strava[1] ya que aunque no esté centrada en el entrenamiento con pesas posee una gran cantidad de funcionalidades muy interesantes. A continuación me gustaría señalar cuáles son en cuanto a gestión de rutinas y perfil de usuario, ya que son los temas que más relación tienen con la aplicación que yo estoy desarrollando.

- En primer lugar cuenta con un componente social en el cual mediante agregar amigos puedes consultar sus últimos entrenamientos incluso ponerles comentarios. Es una funcionalidad muy interesante ya que convierte una aplicación que en un principio es de uso individual para almacenar tus rutinas en una red social donde la gente puede consultar los entrenamientos y perfiles de otras personas.
- El apartado del perfil de usuario tiene una gran cantidad de personalización para que el usuario se moldee a su gusto las preferencias. Permite modificar el idioma y las unidades en las que se muestran los entrenamientos, lo que la convierte en una aplicación con un grado de internacionalidad alto, también permite gestionar todo lo relacionado con la privacidad y gestión de los datos.
- De Strava[1] es muy remarcable lo intuitiva que es la aplicación a pesar de la gran cantidad de funcionalidades que tiene, crear un en entrenamiento no resulta una tarea tediosa ni complicada debido a que el diseño está muy centrado en el usuario. No solo el apartado de crear una rutina, el apartado social funciona muy similar a

otras redes sociales que ya conocemos, lo que hace que un usuario no tenga complicaciones a la hora de entender su funcionamiento.

- Una característica por la que destaca Strava[1] es la gran cantidad de gráficas y seguimiento que presenta conforme vas rellenando entrenamientos en la app, estos ofrecen datos muy detallados y se puede ver el progreso del usuario.



Figura 2.1: Gráficas en Strava.

## 2.2 Hevy

Hevy[2] es la aplicación actual con más similitudes con la que estoy desarrollando, ya que comparten muchas funcionalidades pero con ligeras diferencias. Hevy[2] es una empresa Española fundada en 2019 que actualmente es una de las más reconocidas mundialmente ya que fue una de las empresas que mejor llevó a cabo la transición del papel y libreta para apuntar los entrenamientos a llevarlo todo digitalmente por medio de la app.

Es una aplicación que permite a los usuarios ser los protagonistas de sus rutinas con una gran cantidad de ejercicios que los usuarios pueden utilizar para elaborar sus entrenamientos. Me gustaría señalar las características más destacables de la aplicación de Hevy[2]:

- En primer lugar y por la parte por la que más destaca Hevy[2] es la creación de rutinas. El usuario es capaz de crear una serie de ejercicios y ordenarlos de la manera que considere oportuna para que se adapte al máximo a las necesidades que tenga, también este tipo de rutinas pueden marcarse para que se repitan una serie de días que el usuario considere, así una vez ha creado un entrenamiento no es necesario volver a crearlo cada semana.
- Continuando con el tema de la creación de rutinas y más específicamente con los ejercicios que las componen, los ejercicios se muestran de manera muy gráfica permitiendo a todo tipo de usuarios comprender como se realiza el ejercicio. La aplicación cuenta con una gran cantidad de ejercicios introducidos con sus respectivos vídeos, lo que permite que el usuario tenga una gran capacidad de personalización a la hora de la creación de rutinas.
- Otro aspecto en lo que destaca es la cantidad de estadísticas que muestra la app cuando el usuario va introduciendo los resultados de los entrenamientos, permitiendo al usuario monitorear sus avances en los diferentes tipos de ejercicios destacando que la app va mostrando una serie de medallas y logros para motivar al usuario a que se de cuenta de sus progresos a lo largo de los entrenamientos. La aplicación muestra el mayor peso que ha sido capaz de levantar el usuario tanto a 1 repetición como en volumen total de kg movidos (peso x repeticiones).
- Hevy[2] cuenta con un componente social muy remarcable lo que hace que se diferencie de la competencia, permite agregar usuarios a tu lista de amigos para consultar sus entrenamientos y poder comentar sobre ellos. También cuenta con un apartado de atletas sugeridos donde aparecen otros usuarios a los cuales no tienes agregados y te permite conocerlos y seguirlos si el contenido te parece interesante.

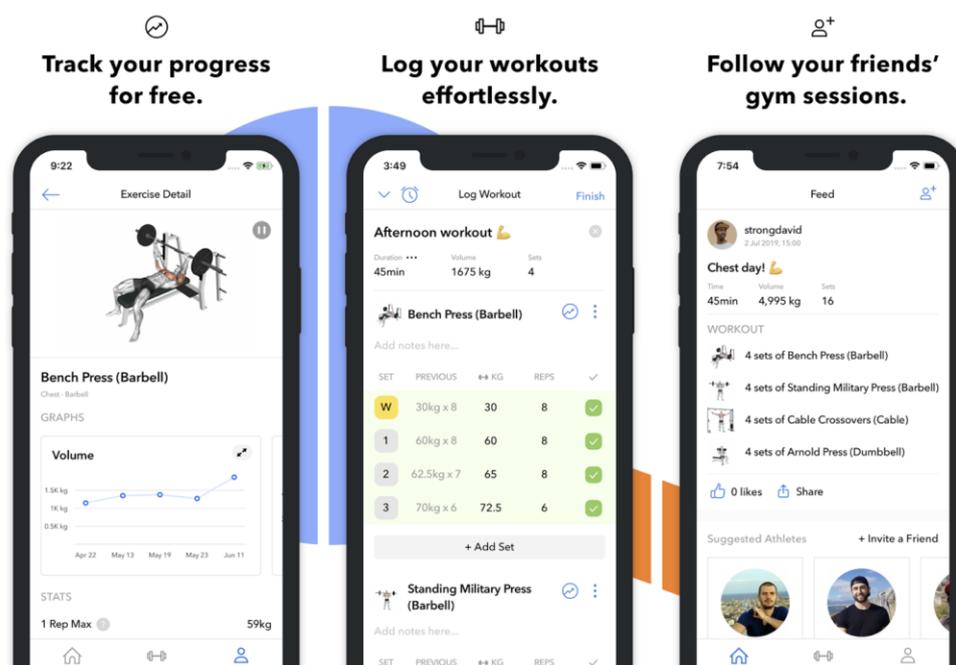


Figura 2.2: App de Hevy.

## 2.3 Virtuagym Fitness

Virtuagym[3] Fitness es una aplicación con características similares a las dos previamente comentadas pero con algunos aspectos que hacen que sea única. La empresa se dió a conocer al mercado en el año 2012 lanzando su aplicación al mercado, siendo esta inicialmente una app fitness simple para consumidores, por el año 2019 añadió el apartado de bienestar a la aplicación y en 2020 el sistema de suscripción con la llegada de la pandemia. En mi opinión es una de las aplicaciones más interesantes por las diferentes características que tiene, las cuales voy a enumerar a continuación:

- La parte por la que destaca esta aplicación no es solo por el tema de la creación de rutinas si no por como está integrado el sistema para que simule que tienes un entrenador personal. La app cuenta con un sistema de inteligencia artificial el cual permite que acorde a unos parámetros y los objetivos del usuario se genere un plan de entrenamiento personalizado, como punto negativo es que esta opción es para usuarios "PRO" de la aplicación lo cual es de pago.
- La creación de rutinas por el usuario es muy completa con videos explicativos y gran variedad de opciones, pero e esta app lo que destaca a la hora de seleccionar ejercicios es que hace mucho incapié en las zonas del cuerpo que se trabajan en cada uno de los ejercicios ayudando a los usuarios menos experimentados a entender la base del entrenamiento con pesas e ir mejorando progresivamente.
- Otro aspecto que hace que esta aplicación sea distinta a las anteriormente comentadas y se diferencia de la competencia es que permite instalar una extensión para llevar todo el tema de la nutrición. Adaptando una gran cantidad de parámetros y objetivos la propia app te genera una dieta y explica con detalle el porqué de cada uno de los alimentos seleccionado, incluso generando recetas y una lista de la compra para llevarlo a cabo de la manera más sencilla posible.
- Virtuagym[3] cuenta con un apartado para explorar el cual te permite como usuario ver videos de sesiones de entrenamientos para realizarlos conjuntamente, escuchar sesiones de música e incluso retos que se proponen a la comunidad con los resultados que han obtenido los usuarios.

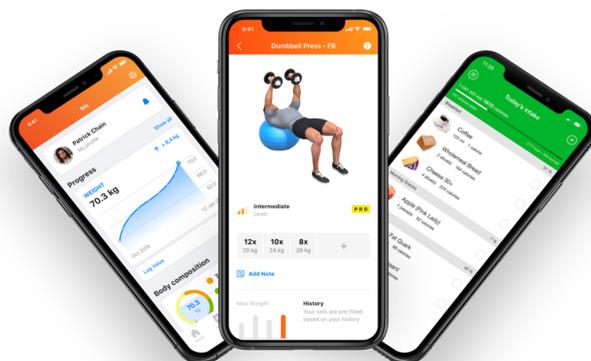


Figura 2.3: App de Virtuagym.

---

## 2.4 Crítica al estado del arte

---

### 2.4.1. Crítica a Strava

Strava[1] es una gran aplicación y puntera en el ámbito de la creación de rutinas cuenta con un gran respaldo de usuarios que corroboran que es una de las mejores aplicaciones para entrenamientos. Como puntos negativos y centrados en el ámbito que nos concierne en esta memoria, Strava[1] no es una aplicación destinada al entrenamiento con pesas, por lo tanto se queda corta en cuanto a contenidos en este aspecto. A diferencia de otras aplicaciones anteriormente comentadas como Hevy[2] y Virtuagym[3] fitness Strava[1] no cuenta con videos explicativos ni creación de rutinas para un entrenamiento con pesas, pero ofrece una gran cantidad de parámetros que estas dos no cuentan.

Otro aspecto negativo y remarcable es que casi todas las opciones requieren de tener una suscripción de pago y las funciones gratuitas están muy limitadas. Para tener una experiencia completa en la creación de rutinas también requiere de acceso a la localización y a una conexión a internet, lo cual limita bastante las funcionalidades de la app.

Por último comentar que la interfaz está muy centrada en el usuario, pero cabe la posibilidad que para ciertos usuarios con menos experiencia en el uso del dispositivo móvil o con el uso de este tipo de aplicaciones, puede resultar algo confuso la cantidad de funcionalidades que presenta y hacer que para el usuario sea complicado encontrar lo que busca.

### 2.4.2. Crítica a Hevy

Hevy[2] es la aplicación con mayores semejanzas con la que estamos elaborando en esta memoria pero se diferencia en algunos aspectos, uno de ellos es que Hevy[2] está completamente centrada en el usuario, la cual no permite que un entrenador te plantee la rutina, es el usuario el que se gestiona todas las rutinas .

Respecto a los puntos negativos de Hevy[2] realmente son pocos ya que cumple a la perfección con su finalidad, cuentan con un gran equipo de trabajo que sabe perfectamente como adaptar la app a los usuarios y saben lo que necesitan hacer para conseguirlo. Como único aspecto negativo a la hora de la creación de rutinas es que estás limitado a los ejercicios que se encuentran registrados dentro de la app, si quieres crear una rutina en la que el ejercicio que quieres realizar no se encuentra en el listado, no se podrá incluir, a diferencia de PowerLog, el entrenador va creando los ejercicios que debe realizar su atleta y esto permite realizar rutinas con tantos ejercicios diferentes como el entrenador le plantee.

### 2.4.3. Crítica a Virtuagym Fitness

Virtuagym[3] es una aplicación muy completa y que cuenta con gran cantidad de funcionalidades interesantes, pero también presenta sus puntos negativos y son una mezcla de los que se han planteado en Hevy[2] y en Strava[1].

En primer lugar comentar que casi todas las funcionalidades más interesantes de la app están reservadas para los usuarios "PRO", los cuales deberán pagar una suscripción. Esto no afecta en exceso a su uso ya que las funcionalidades gratuitas cumplen para la gran mayoría de los usuarios.

En segundo lugar y como le ocurre a Hevy[2], a la hora de la creación de rutinas está limitado a los ejercicios que se encuentran en la app, no permitiendo al usuario añadir

ejercicios nuevos. Otro punto negativo es que está centrada simplemente en los usuarios por tanto no permite que un entrenador cree rutinas a un atleta.

## 2.5 Propuesta

---

De todas estas conclusiones extraídas tras analizar las tres mejores aplicaciones en cuanto a la creación y gestión de rutinas, nace PowerLog, una aplicación que trata de suplir una carencia que se suele ver repetida en este tipo de aplicaciones. PowerLog cuenta con dos entornos, en primer lugar el del entrenador, el cual consiste en una aplicación web donde un entrenador puede monitorear a todos sus atletas de una manera sencilla, permitiéndole crear rutinas, eliminarlas y ver su progreso. El entrenador tiene un listado de todos los atletas que tiene registrados y de manera muy sencilla puede acceder a cada uno de ellos.

Por otra parte se encuentra la app móvil, la cual se desarrolla a lo largo de esta memoria y cuenta con todo el entorno relacionado con los atletas, cuentan con una pantalla principal donde después de hacer "login" se les muestra un calendario donde pueden ver los entrenamientos que le ha planteado su entrenador y ellos mismos pueden rellenar los campos que necesita el entrenador e incluso poner comentarios acerca de como ha ido la rutina. Por el momento PowerLog solo cuenta con esta interacción atleta y entrenador, no está pensada para que un usuario pueda crear sus propios entrenamientos dentro de la app.

---

---

## CAPÍTULO 3

# Análisis del problema

---

Una vez finalizado con el estudio del estado del arte, es buen momento para presentar un análisis del problema planteado para el desarrollo de este proyecto. Este análisis de va a realizar siguiendo una serie de técnicas y métodos adecuados para la realización de éste, algunos de los métodos aplicados van a ser:

- Especificación de requisitos, en este apartado se realizará un estudio de los requisitos funcionales y los no funcionales, subclasificando este último tipo para ayudar a crear una buena lista de requisitos que debe cumplir el sistema planteado.
- Casos de uso y diagrama de clases, en este apartado se presentan los casos de uso que existen en la aplicación junto con un diagrama de casos de uso, además de un diagrama de clases con la finalidad de presentar las clases del proyecto junto con las relaciones existentes entre ellas.
- Por último se realizarán análisis sobre la seguridad, marco legal y ético, energético o de eficiencia algorítmica y de riesgos.

Una vez analizados estos aspectos, en este apartado se va a plantear una identificación y análisis de soluciones posibles, solución propuesta y plan de trabajo con la finalidad de documentar el resultado obtenido de todo el análisis realizado.

### 3.1 Especificación de requisitos

---

En este apartado se va a realizar la especificación de requisitos, con la finalidad de conocer los requisitos que se van a imponer en el diseño y verificación del producto y documentar todas las necesidades y restricciones del usuario y del sistema de manera clara y precisa. En el caso de este desarrollo los requisitos se obtuvieron realizando entrevistas informales, tanto atletas como entrenadores, se decidió este método debido a la cercanía con los futuros usuarios de la aplicación, realizando quedadas y preguntando cuales eran las funcionalidades y requerimientos que necesitaban para una correcta experiencia con la aplicación. Se va a realizar una subdivisión entre requisitos funcionales y no funcionales y éstos últimos divididos en subsecciones como se ha explicado anteriormente.

#### 3.1.1. Requisitos funcionales

Un requisito funcional es una descripción de lo que será el sistema y cómo funcionará para satisfacer las necesidades del usuario. Proporcionan una descripción clara de cómo se supone que el sistema debe responder a un comando en particular, las características

y lo que esperan los usuarios. A continuación se presentan las funcionalidades que se esperan de la aplicación:

- El usuario deberá de poder ser capaz de crearse una cuenta con un email, contraseña, nombre y un id de entrenador, éste id le permite vincularse con su entrenador y que le aparezca como uno de sus atletas para que sea capaz de planificarle la rutina.
- En caso de tener una cuenta el usuario debe ser capaz de realizar login con sus credenciales y acceder a la pantalla principal de la app.
- En la pantalla principal el usuario es capaz de ver su listado de amigos y acceder a sus peticiones de amistad nuevas y buscar usuarios nuevos para agregar a la aplicación.
- Desde la pantalla de calendario el usuario puede acceder a los diferentes días y se le plantea en forma de tabla los entrenamientos planteados por su entrenador, en caso de no tener aparece un mensaje explicativo. En caso de aparecer el entrenamiento el atleta debe de ser capaz de rellenar los campos necesarios en la tabla y al finalizar escribir comentarios y sensaciones para su entrenador.
- Desde la pantalla de ajustes el usuario puede acceder a toda la configuración de su cuenta donde podrá cambiar el nombre, enviar un email para la recuperación de contraseña o cerrar sesión.

### 3.1.2. Requisitos no funcionales

Los requisitos no funcionales explican las limitaciones y restricciones del sistema a diseñar. Estos requisitos no tienen ningún impacto en la funcionalidad de la aplicación. Para la realización del análisis en este documento se van a dividir en diferentes subcategorías:

- Requisito de rendimiento: La aplicación debe tardar un máximo de un segundo en mostrar el entrenamiento planteado por el entrenador, mostrar el listado de amigos prácticamente instantáneo y cerrar sesión de manera casi instantánea para evitar esperas del usuario.
- Requisitos de disponibilidad: El sistema debe de estar disponible un mínimo de 0.99 o 99 % del tiempo, para evitar que los usuarios vayan a entrenar y no puedan registrar sus resultados.
- Requisitos de usabilidad: La aplicación debe de resultar sencilla, sin funcionalidades rebuscadas y que para la gran mayoría de acciones de la aplicación sea suficiente con pulsar 1 o 2 veces la pantalla.
- Requisitos de escalabilidad: El sistema debe de ser capaz de soportar un entrenamiento al día durante muchos días sin causar sobrecargas o ralentizamientos.
- Requisitos de testeabilidad: El sistema debe someterse a una pruebas previas al lanzamiento el cual corroboran que el funcionamiento es el adecuado, estas pruebas se realizarán tanto de forma automática con tests unitarios como de forma manual con la experiencia de uno o dos beta testers.

## 3.2 Casos de uso y diagrama de clases

En este apartado se van a desarrollar los casos de usos con los que cuenta la aplicación móvil junto con un diagrama de clases que conforman todo el sistema planteado.

### 3.2.1. Casos de uso

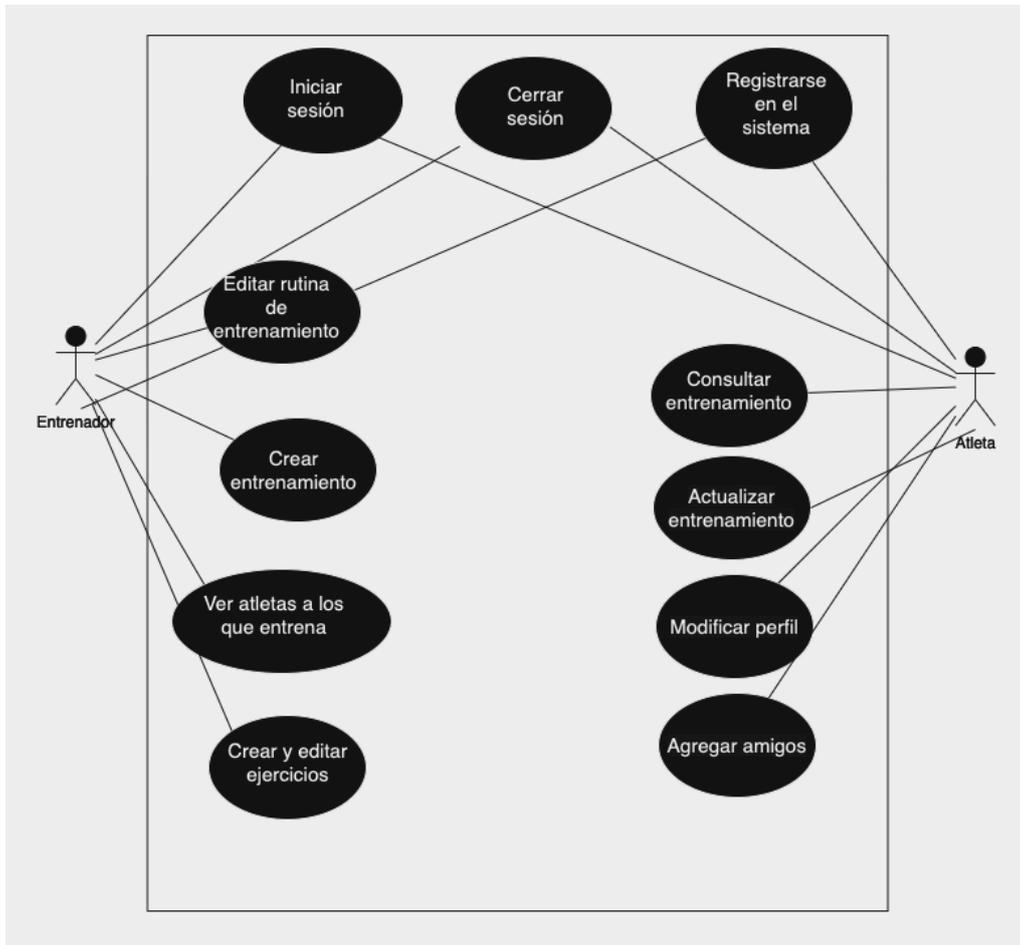


Figura 3.1: Diagrama de casos de uso.

- En caso de que el usuario haya cerrado sesión o sea su primera vez entrando en la app llegara a la pantalla de login, desde aquí el usuario puede acceder a la pantalla de registrarse para crearse una nueva cuenta si es su primera vez, una vez se crea la cuenta entra directamente a la app con los credenciales que ha utilizado. En caso de ya tener cuenta, se logea con sus credenciales y accede a los contenidos de la app.
- El usuario consulta la lista de amigos que tiene agregados desde la pantalla principal de la aplicación, pulsando el botón de la parte inferior derecha puede acceder al apartado de búsqueda de nuevos usuarios para agregar nuevos a su listado de amigos.
- El usuario modifica su perfil desde el apartado de configuración, se permite al usuario cambiar su nombre visible dentro de la app, enviar un email de recuperación de contraseña y cerrar sesión desde este apartado.

- El usuario consulta los entrenamientos en la pantalla del calendario se le permite al usuario ver todos los entrenamientos disponibles si va seleccionando las fechas, una vez se encuentra en un día si tiene entrenamiento le aparece al usuario en forma de tabla y en caso de no haber aparece un texto aclarativo. En caso de si haber entrenamiento el usuario puede rellenar los campos necesarios actualizando la base de datos para que el entrenador pueda ver los resultados de forma inmediata, una vez terminado el entrenamiento y pulsado el botón de finalizar entrenamiento aparece una pantalla resumen y para escribir comentarios y sensaciones que le llegarán directamente al entrenador.

### 3.2.2. Diagramas de clases

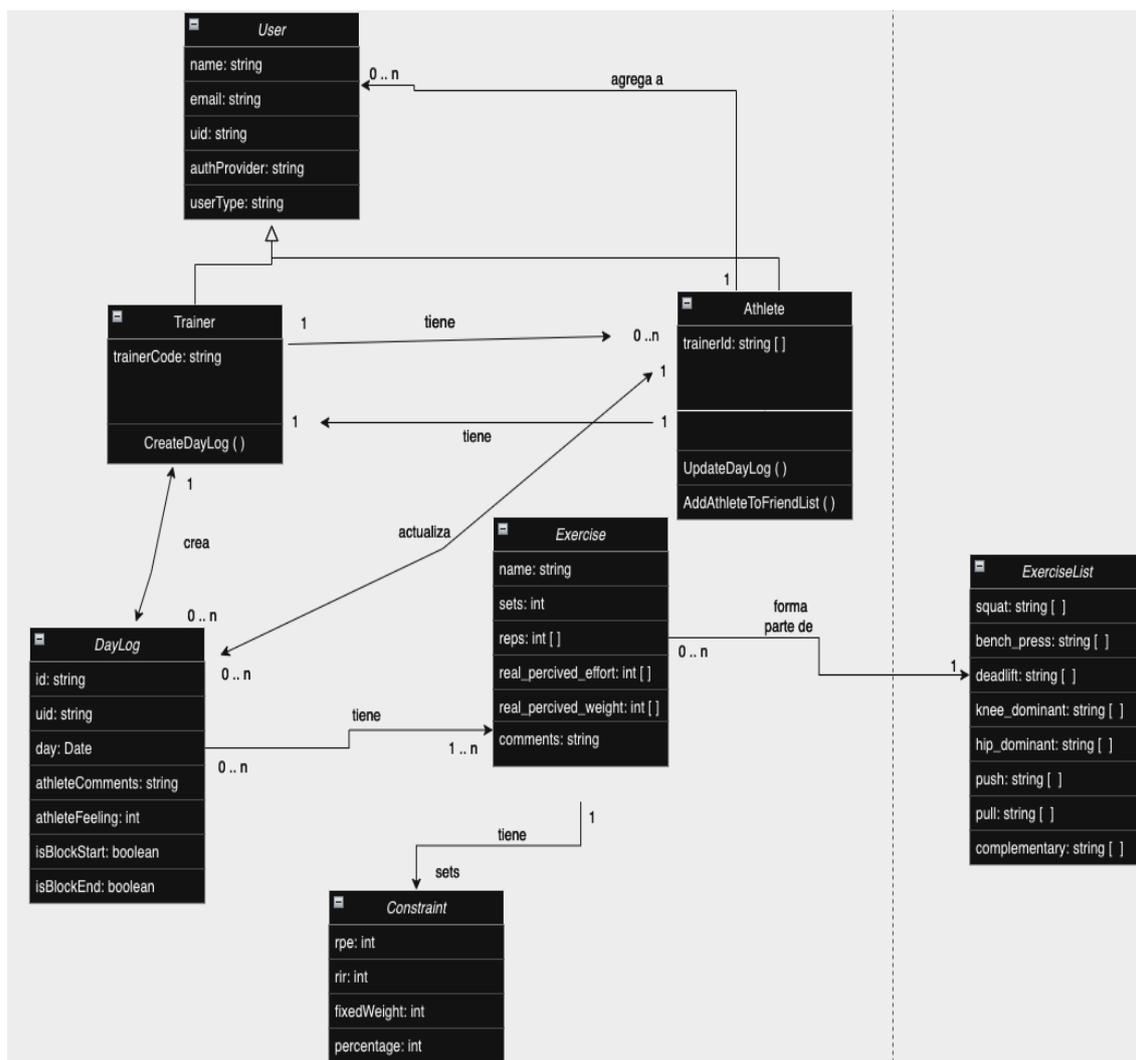


Figura 3.2: Diagrama UML.

El diagrama de clases está formado por en primer lugar usuarios que pueden ser de dos tipos o Atletas o Entrenadores, de ahí que exista una clase "User" y dos que hereden de ella que son "Athlete" y "Trainer". Las relaciones entre estas tres tablas son: un entrenador puede tener cero o infinitos atletas pero un atleta solo puede tener un entrenador, un atleta puede añadir a su lista de amigos cualquier usuario del sistema sin importar si es entrenador o atleta.

Por otra parte se encuentran los "DayLog" que está conformado por un listado de ejercicios "Exercise" entre otros campos, un ejercicio a su vez está formado por un listado de constraints "Constraint" que hay tantas como series tenga el ejercicio, y a su vez el ejercicio forma parte de un único grupo de ejercicios ".ExerciseList".

Un entrenador es capaz de crear cero o muchos entrenamientos pero un entrenamiento solo puede ser creado por un entrenador, a su vez un atleta puede modificar cero o muchos entrenamientos pero solo puede ser modificado por un atleta.

### 3.3 Seguridad

---

En este apartado se va a analizar uno de los aspectos más importantes como es la seguridad, actualmente hackear y robar los datos de los usuarios de aplicaciones es algo muy sencillo debido al avance de las tecnologías y el desconocimiento de la gran mayoría de los usuarios, en este apartado se van a desarrollar los aspectos de seguridad empleados en el desarrollo de la aplicación. Para un correcto análisis se van a diferenciar dos apartados, en primer lugar la aplicación móvil y en segundo lugar el backend de dicha aplicación.

Por parte de la aplicación móvil, todo el tema de la autenticación viene gestionada por Firebase[4], que consiste en una plataforma ofrecida por google la cual permite mantener y desarrollar aplicaciones en la nube. Firebase[4] ofrece una función de autenticación mediante diferentes vías como pueden ser Google, Facebook, Twitter o vía correo electrónico y contraseña. Firebase[4] gestiona todo el tema de la seguridad haciendo que el desarrollador no tenga que hacerlo y ofrece una tranquilidad y seguridad a los usuarios ya que está respaldada por google. En el caso de la aplicación móvil se ha optado por un inicio de sesión con email y contraseña, la cual puede ser recuperada en cualquier momento y sustituida por una nueva. Los usuarios quedan almacenados en el apartado de autenticación de Firebase[4] y además para esta aplicación se ha creado una Firestore database para almacenar campos de los usuarios los cuales no se contemplan en la autenticación de Firebase[4]. La Firestore database es una base de datos en tiempo real la cual permite almacenar en la nube cualquier tipo de información y que sea accesible de forma sencilla desde cualquier aplicación tanto web como móvil.

Para el backend se ha utilizado la base de datos de mongoDB[5], esta base de datos lo que permite es una securizarla de diferentes maneras, una de ellas es la autenticación lo que permite que solo usuarios que formen parte de la aplicación puedan realizar operaciones de lectura y escritura además de poder diferenciar varios roles lo que permite que un atleta no pueda crear entrenamientos por ejemplo. Otra característica de seguridad de mongo es que permite la encriptación de datos en reposo, esta funcionalidad permite securizar los datos del disco y aunque el atacante acceda a él no pueda tomar el control de la base de datos.

El despliegue del backend se realizó mediante la plataforma Digital Ocean[6], la cual se desarrolla en profundidad en apartados posteriores, que permite crear droplets que son máquinas virtuales en las que se aloja el código y funcionan a modo de servidor. Estas máquinas virtuales ofrecen una gran cantidad de personalización con diferentes planes de pago pero en el caso de PowerLog se configuraron los puertos desde los que se podían realizar peticiones(3000) además de utilizar una herramienta llamada NGINX[7] para realizar un port forwarding y que permite redireccionar el tráfico a los puertos 80 para peticiones HTTP y 443 para HTTPS manteniendo de esa manera los estándares.

## 3.4 Análisis energético o de eficiencia algorítmica

---

En este apartado se va a analizar un aspecto el cual es cada vez más importante ya que debido al coste de los dispositivos conviene alargar la vida útil lo máximo posible, además de proveer de una buena experiencia de usuario sin tiempos de carga elevados ni ralentizaciones. En este apartado se va a analizar que operaciones ocurren en primer y segundo plano y como afectan a la degradación del dispositivo.

En primer lugar conviene analizar que se ejecuta en primer plano, en este caso cuando el usuario está realizando el entrenamiento debe de mantener la app abierta o ir entrando cada vez que finaliza una serie o ejercicio para rellenar los datos correspondientes, la app no cuenta con funcionalidades demasiado costosas lo que hace que aunque esté abierta en primer plano el consumo de batería no es excesivo. En segundo plano la app no cuenta con funcionalidades que hagan que su coste sea elevado ya que no contiene por ejemplo notificaciones, permisos de ubicación o sensores que es lo que puede llegar a hacer que se disparee el consumo. En definitiva el consumo que puede llegar a tener la app en segundo plano vendrá determinado por como lo gestione el sistema operativo del dispositivo.

### 3.4.1. Comunicación en red

En este subapartado se va a analizar las operaciones que se realizan en la red y posibles mejoras que se podrían haber realizado en este apartado.

Como he comentado anteriormente la aplicación no cuenta con una gran cantidad de funcionalidades lo que hace por lo tanto que la carga de operaciones en la red no sea excesiva pero si conviene comentar las que se realizan. En primer lugar para obtener la lista de amigos se hace una petición a firebase[4] donde se obtienen los usuarios que tiene agregados en la lista, es una operación muy poco costosa y que solo se realiza al entrar a la app o al actualizar la lista, en caso de ya tener los datos en caché no vuelve a realizar la petición. En segundo lugar, en el apartado de entrenamientos es donde se encuentra el grueso de peticiones ya que para el sistema que quería montar era necesario, por si cabía la posibilidad de que un entrenador modifique la rutina de su atleta mientras está en la app, para ello cada vez que el usuario entra a este apartado de calendario se realiza una petición al backend para obtener su lista de ejercicios del día. Por otro lado cada vez que el usuario rellena un campo se ha introducido un debouncer(tiempo de espera en el que se almacenan los datos y después de ese tiempo realiza una acción) con la finalidad de que no haga una petición al backend cada vez que rellena un campo si no que sea después de un tiempo prudencial. Por último comentar las operaciones referentes al perfil, como actualizar los datos o hacer login, al actualizar los datos se realiza una petición a Firebase[4] para modificar lo que sea necesario lo cual no es una carga para la app y referente al login, la app cuenta con un sistema de caché para almacenar en el dispositivo del usuarios sus credenciales de tal manera que una vez inicia sesión no se cierra a no ser que el usuario lo requiera, aligerando así tiempos de carga y sobrecarga de peticiones.

Referente al backend la comunicación en la red es un aspecto muy importante para tratar de evitar ataques ya se ha comentado en apartados anteriores aspectos referentes a la seguridad pero referente a la comunicación en red se ha configurado el droplet de Digital Ocean[6] de tal manera que escuche en el puerto 3000 del servidor y utilizando la herramienta NGINX[7] se ha realizado un port forwarding con la finalidad de redireccionar a los puertos deseados en este caso el 80 para peticiones HTTP y el 443 para peticiones HTTPS. Se utiliza principalmente el protocolo HTTPS ya que ofrece al usuario una mayor

seguridad y permite cifrar los datos que se transmitan entre el usuario y el navegador, esto lo que provoca es que HTTPS sea mucho más seguro en cuanto a ataques de tipo "man-in-the-middle" por ejemplo.

Como aspectos mejorables en este apartado podría ser mejorar la forma en la que está montado el sistema de entrenamientos para evitar sobrecargas de peticiones y tiempos de espera que pueden hacer que el usuario no se encuentre cómodo utilizando la app.

## 3.5 Análisis del marco legal y ético

---

En este apartado se va a analizar el marco legal y ético ya que es un aspecto importante con el que deben contar las aplicaciones con la finalidad de construir relaciones sanas con los usuarios de la aplicación y que se sientan seguros al utilizar la app. Esta sección se va a dividir en subsecciones que creo conveniente abordar para un correcto análisis.

### 3.5.1. Análisis de la protección de datos

Este apartado lo considero de vital importancia ya que en la era actual que vivimos en la que el robo de datos está a la orden del día conviene tener una buena protección de datos para garantizar la seguridad de los usuarios, en adición la aplicación gestiona cuentas de correo electrónico y contraseñas por tanto con más razón para asegurar los datos.

Como se ha comentado en previos apartados Firebase[4] es el encargado de gestionar las cuentas de los usuarios, para no repetir los contenidos y a modo de resumen, Firebase[4] es un sistema respaldado por Google y con numerosos métodos de inicio de sesión los cuales garantizan la integridad de los datos de los usuarios.

### 3.5.2. Propiedad intelectual

En este apartado se va a desarrollar la propiedad intelectual de una aplicación, en una app es importante abarcar varios aspectos legales relacionados con el desarrollo. Se va a dividir en subsecciones que considero importantes de remarcar en este aspecto.

- En cuando a licencias de software cabe recalcar que para el desarrollo de la app se han utilizado librerías "open source" que se pueden encontrar en <https://pub.dev/>[8], aquí los usuarios suben su código a repositorios de git para que el resto de usuarios puedan reutilizar los componentes que ellos han creado y así mejorar la calidad de vida de otros desarrolladores. En esta aplicación se han utilizado librerías como "Bloc"[15] para el manejo de estados dentro de la app o "Calendar Picker" para el calendario de la app, este apartado de librerías estará más desarrollado en apartados posteriores de desarrollo de la solución
- Por último comentar como se manejarían las situaciones en caso de que un usuario utilice la app de manera indebida o sin autorización, en este caso se procedería a un retiro de la cuenta y por parte de los entrenadores se eliminaría el atleta de su lista y no se le permitiría volver a usar la app, para garantizar un buen ecosistema entre todos los usuarios.

### 3.5.3. Marco ético

En este apartado se va a desarrollar la ética de una aplicación, donde se van a analizar los diferentes dilemas morales con los que se puede encontrar el usuario.

En cuanto al diseño, éste estará centrado en el usuario para facilitar que sea accesible tanto a usuarios expertos como no expertos en el uso de este tipo de tecnologías para garantizar una experiencia positiva, este apartado se desarrollará con más detalle en apartados posteriores. En cuanto a responsabilidad medioambiental PowerLog fomenta el uso del dispositivo móvil en lugar de apuntar los resultados en una libreta de papel lo cual hace que se reduzca su uso y contribuir con el medio ambiente. Por último la aplicación garantiza que no se fomenta ningún tipo de comportamiento adictivo.

## 3.6 Análisis de riesgos

---

En este apartado se van a analizar los posibles riesgos que puede tener el desarrollo de la aplicación y al ser una aplicación centrada para usuarios se desarrollarán en detalle los riesgos de aceptación y los indicadores de satisfacción que se pueden encontrar en ella.

El riesgo de aceptación es la posibilidad de que el producto desarrollado sea bien o mal recibido por los usuarios, para analizarlos los posibles riesgos de aceptación conviene remarcar los siguientes puntos:

- Resistencia al cambio: Cabe la posibilidad de que los usuarios ya cuenten con una aplicación que cumpla las funciones que necesitan por tanto quieran resistirse a cambiarse de aplicación y entender las funcionalidades nuevas que presenta.
- Falta de valor agregado: Como he comentado anteriormente y más desarrollado en el apartado de estado del arte, es posible que otras aplicaciones del mercado ofrezcan más funcionalidades de las que ofrece PowerLog y produzca rechazo al realizar el cambio.
- Problemas de funcionalidad: Como tanto mi compañero como yo somos bastante inexpertos a la hora de realizar este tipo de proyectos y nos estamos iniciando en este mundo con esta app, cabe la posibilidad de que haya algunos errores de funcionalidad que afecten negativamente a la percepción de los usuarios. También es posible que la frecuencia con la que se reciben actualizaciones no sea la esperada comparando con otras aplicaciones más grandes y con un mayor número de trabajadores y más experimentados.

Una vez los usuarios están haciendo uso de la aplicación podemos realizar un análisis sobre los indicadores de satisfacción y de esa manera conocer el grado de satisfacción de los usuarios de la app. Algunos de esos indicadores pueden ser:

- La tasa de retención: Este indicador sirve para conocer la cantidad de usuarios que han descargado la app y se han quedado en ella y la han adoptado como su aplicación principal para los entrenamientos, se puede medir en tiempo por ejemplo 3-6 meses. En nuestro caso la tasa de retención es elevada ya que la mayoría de usuarios son conocidos y venían buscando una aplicación que cumpliera con las características que PowerLog ofrece por tanto se han quedado.
- Calificaciones y reseñas: Es una manera directa de conectar con los usuarios y que ellos expresen lo que les transmite la app, se puede hacer mediante encuestas y

valoraciones pero en nuestro caso fue hablando de manera directa con los usuarios y que nos comunicaran lo que les parece que está bien y mal.

- Porcentaje de errores: Esta métrica nos permite conocer si la app falla mucho o no, en caso de si hacerlo puede resultar tedioso para los usuarios que esté fallando continuamente y que no estén satisfechos con el rendimiento. En nuestro caso personal, en un inicio aparecieron más errores de los esperados que fuimos solventando hasta crear una app sólida.

En conclusión es importante realizar un análisis de riesgos con la finalidad de obtener los resultados necesarios para ver donde se puede mejorar y ver en que cosas la app está cumpliendo con su funcionalidad, el objetivo es desarrollar una aplicación lo más completa posible y con unos usuarios contentos después de su uso.

### 3.7 Análisis de posibles soluciones

---

En este apartado se van a desarrollar algunas de las posibles soluciones al problema planteado en este documento pero en el siguiente se desarrollará la que finalmente fue seleccionada.

- En primer lugar una posible solución al problema sería crear un excel muy profesional, con gran cantidad de gráficos y facilidades al usuario para que solo tenga que apuntar los resultados de los entrenamientos, este documento estaría compartido entre atleta y entrenador. Esta solución no es la óptima ya que es un sistema "antiguo" comparado con las capacidades que nos ofrece actualmente la tecnología, otra de las desventajas es que la comunicación es muy escueta, siendo una celda el único punto donde se podrían comunicar atleta y entrenador o tendrían que utilizar otras herramientas de chat. Por último la capacidad de enviar archivos como vídeos del entrenamiento o audios el excel no lo permite, por tanto no es la mejor de las opciones para abordar el problema planteado.
- En segundo lugar otra de las posibles soluciones es elaborar una app solo para atletas como Hevy[2], una de las anteriormente desarrolladas, y que el entrenador le proporcione el entrenamiento por otro medio como por ejemplo vía Gmail o WhatsApp y sea el propio atleta que se encargue de introducir su entrenamiento y de proporcionar los resultados a su entrenador. Esta solución es más viable que la anteriormente comentada porque es mucho más completa pero tiene un inconveniente y es que es muy lento y tedioso para ambos tener que estar introduciendo el entrenamiento a mano cada vez y el entrenador los resultados. Existe una variante de esta solución que consistiría en que el atleta le deje los credenciales de la app al entrenador y sea él mismo el que rellene el entrenamiento, pero tiene el principal inconveniente que por el lado del entrenador debería de recordar todos los credenciales de los atletas a los que lleva, que en caso de ser pocos no habría mucho problema, pero en caso de ser muchos se complicaría mucho y sería muy tedioso para el entrenador.

El estudio de estas dos soluciones nos lleva a la solución que finalmente se ha adaptado para llevar a cabo en este proyecto, es una solución la cual intenta maximizar el número de "pros" minimizando el número de "contras", esta solución se va a desarrollar con más detalle en el siguiente apartado pero básicamente consiste en un sistema síncrono en el que el entrenador cuenta con sus atletas y plantea las rutinas que se plasmarán en las apps móviles de los atletas.

---

## 3.8 Solución propuesta

---

Después de un proceso de comunicación tanto con entrenadores como con atletas conocidos, la conclusión fue montar un sistema en el cual el entrenador pueda publicar entrenamientos y tener de forma sencilla control de todos sus atletas y que por parte de los atletas puedan consultar y completar la información necesaria sobre sus entrenamientos. Para ello la solución implementada consiste en un frontend para entrenadores en web donde se logean los entrenadores y gestionan todo desde el panel de control y otro frontend en dispositivos móviles para atletas donde se les plasma dicha información generada por sus entrenadores, todo ello conectado con un backend que se encuentra alojado en un servidor para tratar de mantener la disponibilidad del sistema el máximo tiempo posible.

---

## 3.9 Plan de trabajo

---

El proyecto de PowerLog comenzó en enero, como fue un Desarrollo Centrado en el usuario durante los meses de enero y febrero conformaron la fase de planteamiento de objetivos, fase de diseño y fase de análisis, agendando reuniones con los futuros usuarios y analizando sus requisitos. De Marzo a Mayo fue la etapa de desarrollo o implementación donde se programó todo el código necesario tanto en el frontend como en el backend y se hicieron los despliegues necesarios junto con la fase de pruebas en la que los usuarios comentaban sus sensaciones al utilizar la app, por último en el mes de Julio y Agosto redacción de la memoria.

En cuanto al presupuesto que ha requerido este proyecto no ha sido muy elevado para la cantidad de cosas que se han realizado, en primer lugar la cuenta de desarrollador de apple que fue necesaria a la hora de desplegar en dispositivos iOS que fueron 100€, a la hora de desplegar el backend también había un coste que fue desde Marzo y son unos 6€ al mes aproximadamente, con esto se concluye que el coste total en dinero ha sido aproximadamente de 140€. En cuanto al coste en horas de trabajo aproximadamente ha sido una media de 2h al día ya que había días más productivos que otros y algunos festivos y fin de semanas en los que no trabajaba en el proyecto, en estos 8 meses han sido unos 150-160 días de media haciendo un calculo aproximado de entre 300 y 360h de trabajo.

---

## CAPÍTULO 4

# Diseño

---

A lo largo de este apartado se van a desarrollar como ha sido la fase de diseño presentando las estructuras de documentos y los bocetos que se han realizado como fase previa al desarrollo.

### 4.1 Estructura de documentos

---

Para este proyecto se ha utilizado la base de datos de MongoDB[5], una base de datos no relacional que está compuesta por documentos, estos documentos tienen una clave y un valor y permiten tener diferentes entradas. Para este proyecto se han utilizado los siguientes documentos:

```
export class DayLog {
  @Prop({ required: true })
  day: Date;

  @Prop({ required: true })
  exercises: Exercise[];

  @Prop({ required: true })
  athleteUid: string;

  @Prop({ required: false })
  isBlockStart: boolean;

  @Prop({ required: false })
  isBlockEnd: boolean;

  @Prop({ required: false })
  athleteComments: string;

  @Prop({ required: false })
  athleteFeelings: number;
}
```

Figura 4.1: Documento del DayLog

Esta figura representa el documento del DayLog donde se puede comprobar los atributos que lo componen, en este caso son la fecha del entrenamiento, el listado de ejercicios que lo componen, la uid del atleta, si es inicio de bloque, si es final de bloque, los comentarios del atleta y las sensaciones. Se puede comprobar también cuales de estos campos son requeridos y cuales no lo son.

```
class Exercise {
  exercise_name: string;
  sets: number;
  reps: string[];
  constraints: IConstraint[];
  real_perceived_effort: number[];
  real_weight: number[];
  comments: string;
}
```

Figura 4.2: Documento del Entrenamiento

Esta figura representa el documento del Entrenamiento donde se puede comprobar los atributos que lo componen, en este caso son el nombre del ejercicio, las series, las repeticiones, el listado de constraints, el esfuerzo percibido, el peso manejado y los comentarios.

```
export class IConstraint {
  @Prop({ required: true })
  rpe: int;
  @Prop({ required: true })
  rir: int;
  @Prop({ required: true })
  fixed_weight: int;
  @Prop({ required: true })
  percentage: int;
}
```

Figura 4.3: Documento de las Constraints

Esta figura representa el documento de las Constraints donde se puede comprobar los atributos que lo componen, en este caso son esfuerzo percibido(rpe), repeticiones en recámara(rir), peso movido, y porcentaje del máximo que es capaz de mover el atleta.

```
export class ExerciseList {
  @Prop({ required: true })
  trainerId: string;
  @Prop({ required: true })
  squat: string[];
  @Prop({ required: true })
  bench_press: string[];
  @Prop({ required: true })
  deadlift: string[];
  @Prop({ required: true })
  knee_dominant: string[];
  @Prop({ required: true })
  hip_dominant: string[];
  @Prop({ required: true })
  push: string[];
  @Prop({ required: true })
  pull: string[];
  @Prop({ required: true })
  complementary: string[];
}
```

Figura 4.4: Documento del ExerciseList

Esta figura representa el documento del listado de ejercicios donde se puede comprobar los atributos que lo componen, en este caso son todos los posibles tipos de ejercicios a los que puede formar parte un ejercicio creado por un entrenador.

## 4.2 Diseño de las interfaces

---

En este apartado se van a presentar el primer boceto realizado en papel al inicio del proceso de creación de la aplicación con la finalidad de presentar a los usuarios de manera visual una idea de lo que iban a contener las diferentes interfaces para tratar de obtener y crear un desarrollo lo más adaptado al usuario posible.

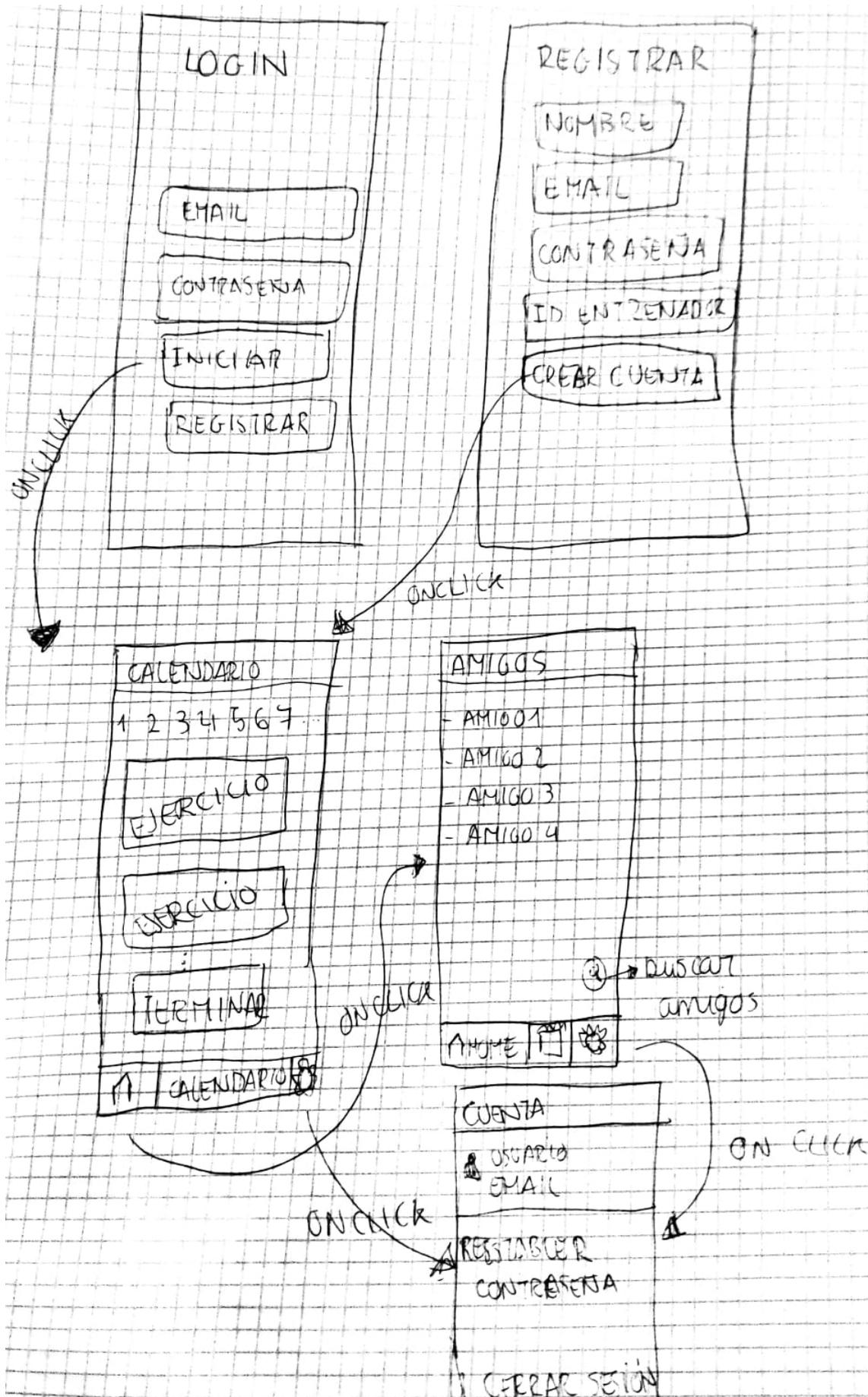


Figura 4.5: Boceto de la aplicación móvil

---

En este boceto se pueden comprobar las diferentes interfaces con las que iba a contar la aplicación, era una primera versión pero se podía comprobar también los diferentes flujos con los que contaba la aplicación, es decir de donde se podía navegar desde cada pantalla al hacer click en los diferentes inputs clickables. Tras estos bocetos y la validación de los usuarios por parte de la aplicación móvil se comenzó directamente con la fase de desarrollo de dichas pantallas y funcionalidades.

---

---

## CAPÍTULO 5

# Desarrollo

---

En este apartado se va a desarrollar como se va a llevar a cabo el diseño de la solución, una vez identificados los requisitos del sistema es conveniente desarrollar que se ha de hacer para cumplirlos. A lo largo de este capítulo se comentará en profundidad la arquitectura, tecnologías empleadas, implementación en implantación para el desarrollo de la app PowerLog.

### 5.1 Arquitectura del sistema

---

El sistema está conformado por tres capas que son Cliente, Backend y SGBD. La aplicación móvil es el cliente en este caso y es el que se comunica con el Backend, este cliente lo que realiza son peticiones con la finalidad de realizar diferentes acciones sobre la base de datos, ya puede ser consultar los entrenamientos o actualizarlos por ejemplo. El Backend actúa de intermediario entre el cliente y la base de datos, el backend está montado en un framework llamado NestJS[10], que se explica en detalle en apartados posteriores, y alojado en un droplet de DigitalOcean para el despliegue, este backend es el que interactúa directamente con la base de datos para realizar acciones como actualización de cualquier dato o consulta de estos mismos. Por último el sistema cuenta con una base de datos de mongoDB[5] que es un sistema de bases de datos que almacena datos en formato BSON(similar al JSON), esta base de datos es la encargada de almacenar todos los datos referentes a los entrenamientos y los usuarios y permite realizar las consultas que sean necesarias para el funcionamiento de la aplicación. Todas las tecnologías que se han empleado para llevar a cabo el desarrollo se explican con más detalle en el apartado posterior, aquí un ejemplo gráfico de cual es la estructura de PowerLog desde el punto de vista de la aplicación móvil como cliente, en caso de utilizar la aplicación web, sería la misma estructura pero el cliente en este caso sería el frontend web.

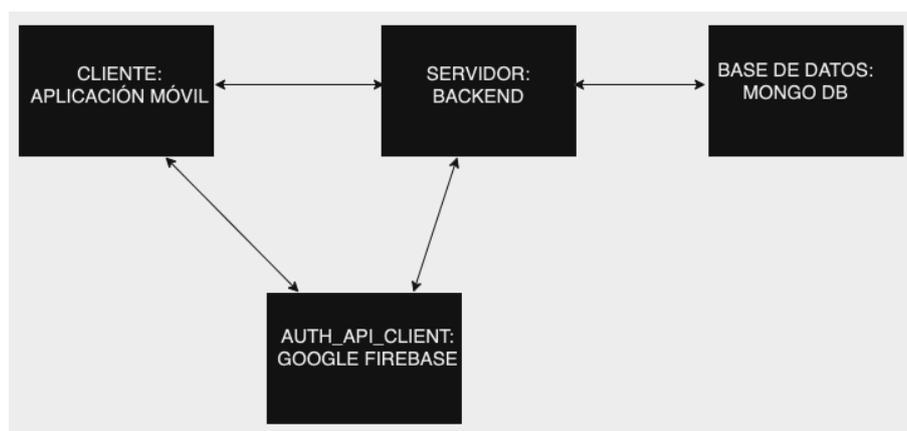


Figura 5.1: Arquitectura del sistema.

## 5.2 Tecnología utilizada

En este apartado se va a desarrollar en profundidad que tecnologías se han llevado a cabo tanto en la app móvil como en el backend para el desarrollo de este proyecto.

En cuanto a la aplicación móvil que van a utilizar los atletas se ha utilizado Flutter[11], un framework de código abierto el cual nos permite el desarrollo de aplicaciones móviles multiplataforma a partir de un único código base lo que agiliza el desarrollo y que además cuenta con gran cantidad de beneficios, los cuales son:

- Uno de los principales beneficios del uso de esta tecnología, además del anteriormente comentado, es que cuenta con el motor gráfico Skia[12], lo que permite un renderizado similar al nativo, proporcionando al usuario una experiencia de aplicación nativa sin realmente serlo.
- El segundo de los beneficios que ayudó a elegir Flutter[11] para el desarrollo de la app móvil es lo que se denomina "Hot Reload", es una de las características más populares con las que cuenta este framework y permite refrescar la interfaz sin necesidad de volver a lanzar la app lo que permite que el desarrollo de interfaces sea muy ágil.
- Por último comentar que la comunidad de Flutter[11] es cada vez más grande y van creando librerías y proyectos los cuales sirven para que otros desarrolladores implementen en sus proyectos.

Para el tema de despliegue con Flutter es muy sencillo ya que existen dos plataformas principales si queremos realizar un despliegue de la app para dispositivos móviles, estas son Android y iOS. Android no ofrece ninguna complicación a la hora de compartir la app, así que simplemente con un comando se construye un archivo .apk en la raíz del proyecto y se puede compartir por cualquier medio como por ejemplo WhatsApp o vía mail para que la prueben los beta testers. En iOS es necesaria una cuenta de desarrollador la cual tiene un coste de 100€ al año y cualquiera de las versiones de la app tienen que pasar por los testers de Apple y validarla para poder distribuirla mediante el uso de una app llamada TestFlight[14] que sirve para la distribución de versiones de aplicaciones. A lo largo del desarrollo de este proyecto no se ha publicado en la Play Store ni en la App Store porque no era necesario para el alcance de este proyecto, pero con Flutter[11] es tan sencillo como enviar las versiones a las diferentes "Stores" y esperar la validación, ya se ha desarrollado con anterioridad en profundidad lo que era Flutter[11] con sus puntos

positivos y el porque de esta elección, pero en este apartado se va a hacer incapié en las librerías y herramientas que se han utilizado para facilitar el desarrollo. En primer lugar se ha utilizado un ordenador Mac ya que si él es imposible buildear apps para iOS ya que Apple lo tiene restringido, dentro del ordenador se han utilizado 3 programas clave para el desarrollo:

- El primero de ellos es el editor de código, en mi caso IntelliJ[20] con numerosos plugins para facilitar el desarrollo de código como por ejemplo el de colores en los paréntesis para ayudar a encontrar con facilidad donde cierran los bloques de código.
- En segundo lugar otro programa que se ha utilizado es el Xcode, este programa consiste entre otras cosas en un simulador de dispositivos iOS, ha sido esencial en el desarrollo para comprobar de manera directa como va quedando el diseño de la app utilizando la funcionalidad del "Hot Reload" comentada previamente.
- Por último la app llamada Transporter[13], es una app desarrollada por Apple que sirve para enviar versiones de la app a los desarrolladores para que la prueben y validen que cumple con las políticas.

Entrando en el código se han utilizado varias librerías clave para el desarrollo entre ellas destacar Bloc[15] para emplear la arquitectura Bloc[15], otra muy importante para el desarrollo ha sido firebase\_auth la cual ha permitido realizar la autenticación de los usuarios mediante Firebase[4], por último destacar la librería del calendario creada por un usuario la cual nos ha permitido implantar y modificar el tipo de calendario que se presenta en la app que es ligeramente diferente al que presenta Google.

Para el backend se ha utilizado el framework de Typescript llamado NestJs[10], este framework se centra en la creación de aplicaciones sostenibles y escalables combinando distintos conceptos de la programación que lo hace una muy buena opción para la creación de backends. Los puntos positivos de NestJs[10] son que mejora algunos de los puntos de NodeJs como por ejemplo la dificultad de mantenimiento del proyecto y nos aporta todo el sistema de librerías con el que Node cuenta, por tanto es como utilizar node pero solucionando los inconvenientes. En segundo lugar para alojar el backend en la nube se ha utilizado Digital Ocean[6], esta plataforma ofrece una gran cantidad de servicios como Droplets, Kubernetes, Bases de datos, etc.. pero la que se ha utilizado para la app de Powerlog son los Droplets. Un Droplet consiste en una instancia de una máquina virtual la cual nos permite ejecutar sistemas operativos, aplicaciones y servicios en la nube y cuenta con una gran cantidad de beneficios:

- Uno de los principales beneficios es la gran cantidad de aspectos configurables con los que cuenta, los cuales pueden ser gasto de CPU, memoria y almacenamiento, que permiten adaptar la instancia a las necesidades de la aplicación.
- Otro de los aspectos por los cuales destaca Digital Ocean[6] es que cuenta con una gran facilidad para implementarlo, incluso para usuarios poco experimentados, tiene una dashboard en la cual se puede monitorear todas las configuraciones del droplet.
- Las imágenes preconfiguradas también son un aspecto a tener en cuenta a la hora de decantarse por Digital Ocean[6] para alojar el backend ya que cuenta con imágenes preconfiguradas para facilitar la implementación de servicios.

En resumen Digital Ocean[6] cuenta con todas las características que se adaptan al desarrollo de esta aplicación ya que es de coste bajo, configurable y de un sencillo uso además

de ofrecer un alojamiento versátil y escalable.

Por último y para finalizar la arquitectura del sistema, el frontend del entrenador. Este apartado se desarrolla con más detalle en el proyecto de mi compañero "PowerLog Web. Plataforma web para la gestión de entrenamientos", pero se ha utilizado Vercel[17]. Esta plataforma se especializa en alojamiento de webs y destaca por las siguientes características:

- Vercel[17] cuenta con un despliegue muy sencillo además de contar con una integración y entrega continua(CI CD) la cual permite su actualización con cualquier cambio en el repositorio.
- La característica principal por la que se ha seleccionado Vercel[17] es por la cantidad de frameworks que soporta, entre ellos se encuentra React[16], Next[18], Angular, etc... Para el desarrollo del frontend de los entrenadores se ha empleado ReactJs[16], una librería de JavaScript mantenida por Facebook y open source, la cual es ideal para el desarrollo de interfaces de usuario atractivas y dinámicas, con una gran comunidad y que destaca por el rendimiento de las aplicaciones desarrolladas.

### 5.3 Implementación y ejemplos de código

En este apartado se van a presentar y desarrollar las arquitecturas que se han empleado tanto en el backend como en la app móvil para llevar a cabo la implementación.

En el caso de la app móvil se ha empleado la arquitectura Bloc[15], esta arquitectura consiste en una vista la cual ejecuta un evento, este evento invoca una acción dentro del Bloc[15](llamada Event) y se encarga de modificar el estado(llamado State), este estado es el que se muestra a la vista y se actualiza acorde a los cambios. Este patrón de arquitectura viene complementado con el uso de los repositorios los cuales se encargan de la comunicación con la API y realizar el "fetching" de datos y devolverle al usuario una respuesta. La vista en Flutter[11] está dividida en componentes los cuales se denominan Widgets y permiten servir de "triggers" para ejecutar los eventos del bloc[15] y producir cambios en la app.

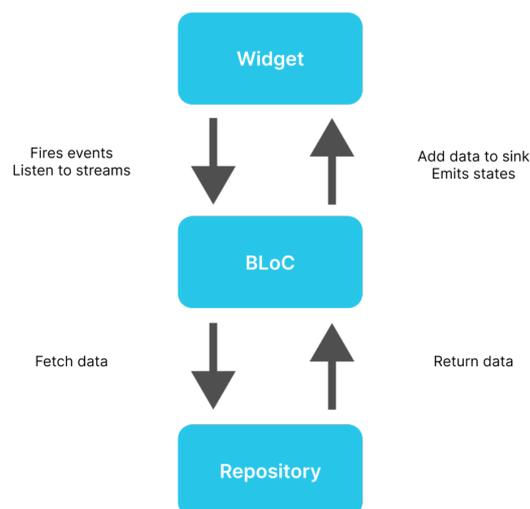


Figura 5.2: Patrón bloc

La estructura de directorios empleada para el correcto desarrollo del patrón Bloc[15] es dividir en dos directorios principales los cuales son /lib y /packages, en /lib encontramos todo lo referente a vistas y bloc[15] y en /packages lo referente a los repositorios. Entrando un poco más en detalle, en /lib se dividirá en subcarpetas teniendo en cuenta las funcionalidades de la app como puede ser por ejemplo hacer login, y dentro de la funcionalidad login se encuentra la carpeta /view con las vistas de la funcionalidad, la carpeta /bloc que contiene el bloc[15] y un archivo barrel que se llama igual que la funcionalidad para exportar los archivos necesarios. Es posible que se utilice cubit que es una variante de bloc[15] simplificada para funcionalidades que no requieran un manejo de estado complejo. La estructura planteada para esta app es la siguiente:

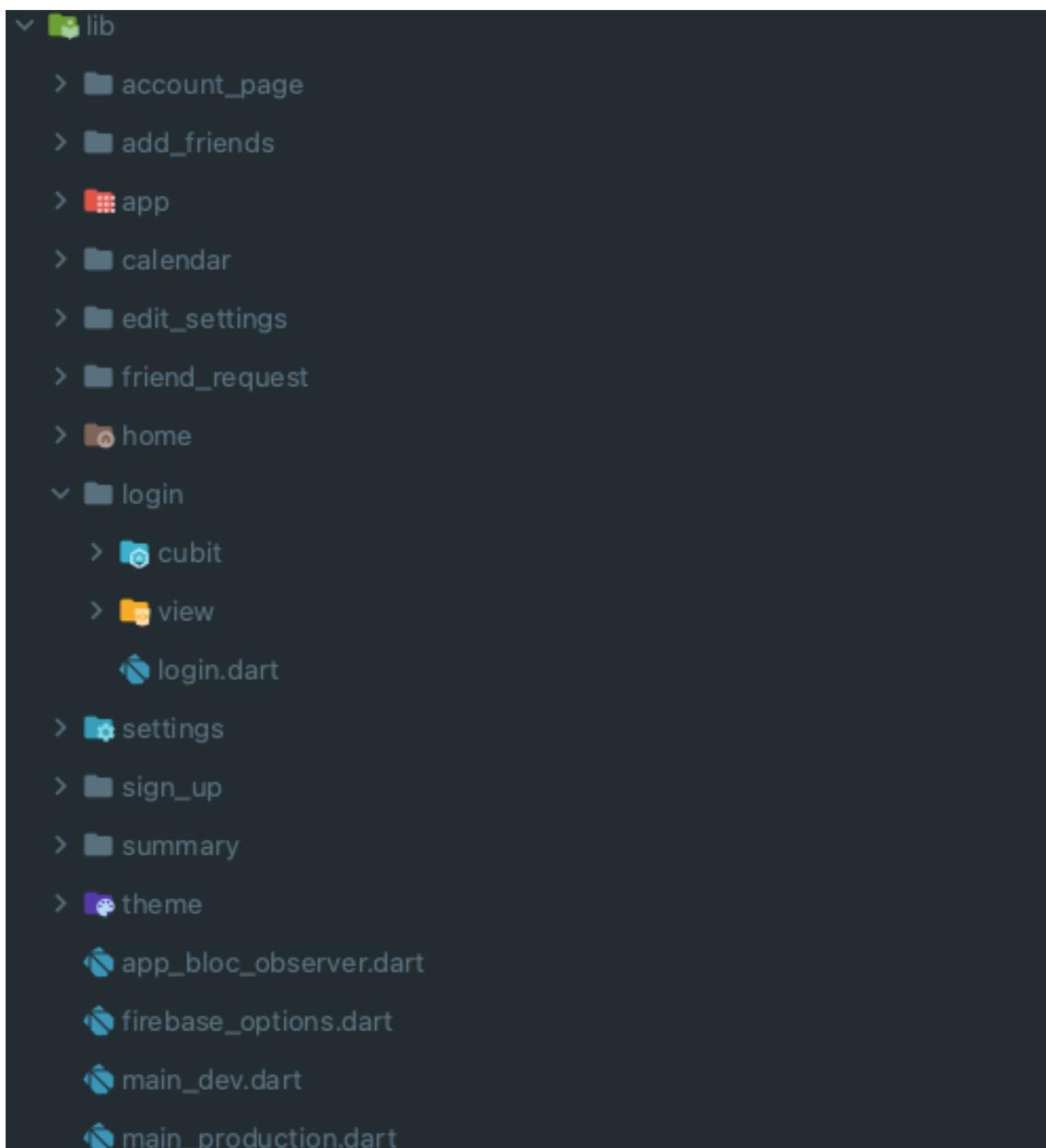


Figura 5.3: Estructura de directorios en /lib



Figura 5.4: Estructura de directorios en /packages

Es muy común tener un directorio denominado /assets el cual contenga todas las imágenes que conforman la app.

En cuanto a la arquitectura del backend es la denominada como arquitectura hexagonal, pero en nuestro caso es una pequeña variación de ella.

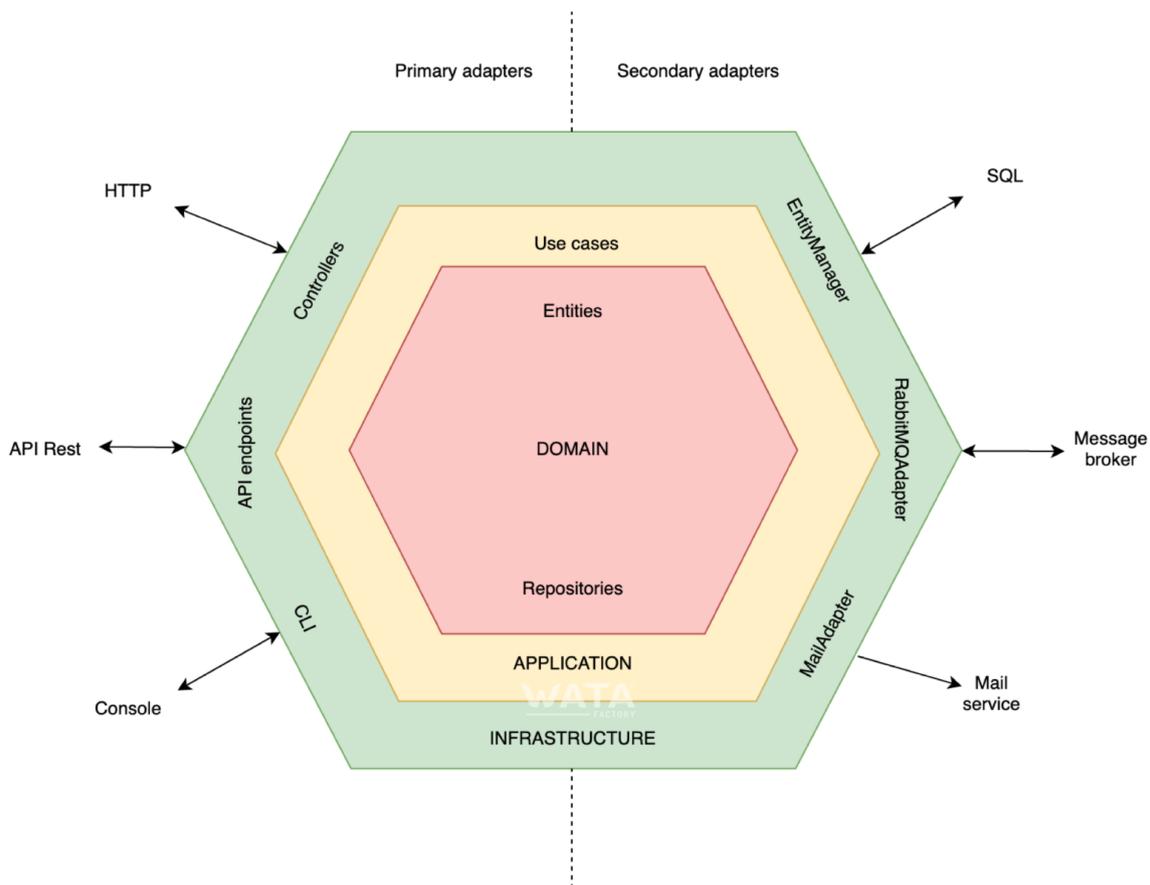


Figura 5.5: Ejemplo de la arquitectura hexagonal al completo.

En primer lugar se han definido los llamados DTO(Data Transfer Object) que consisten en los esquemas que deben tener los objetos para poder ser almacenados en la base de datos, para el caso de un entrenamiento es:

```
1  import { IConstraint } from "src/classes/classes.general";
2
3  export class ChartsDataDto {
4    athleteUid: string;
5    exercise_name: string;
6    data: ChartsDataInfo[];
7  }
8
9  export class ChartsDataInfo {
10   day?: Date;
11   block?: number;
12   real_weight?: number;
13   reps?: number;
14   constraint?: IConstraint;
15   estimated_weight?: number;
16 }
17
```

Figura 5.6: Ejemplo DTO de un entrenamiento

Una vez definidos todos los DTOs que conformarán la app el siguiente paso es la creación de un repositorio en el que se escriben todas las queries que se van a realizar sobre cada uno de los DTOs creados anteriormente por ejemplo:

```
async createDay(dayLog: DayLogClass) {
  return await this.dayLogModel.findOneAndUpdate(
    { day: dayLog.day, athleteUid: dayLog.athleteUid },
    dayLog,
    { upsert: true },
  );
}
```

Figura 5.7: Ejemplo repositorio de un entrenamiento.

Y por último se han configurado los endpoints a los que se debe llamar para realizar acciones sobre la base de datos, esto se ha realizado en un controlador, aquí un ejemplo del controlador necesario para actualizar un día que es el que se utiliza en la app móvil para modificar los datos de los entrenamientos:

```
//ruta que se usa para actualizar los datos de un dia a traves del movil
@Put()
@HttpCode(HttpStatus.OK)
async updateDay(@Body() dayLog: DayLogDto) {
  return this.routineService.updateDay(dayLog).then((value) => {
    if (value.modifiedCount === 0) {
      throw new HttpException(
        "No day found for this date",
        HttpStatus.NOT_FOUND,
      );
    } else {
      return "Day updated successfully";
    }
  });
}
```

Figura 5.8: Ejemplo controlador para actualizar los datos de un entrenamiento

Por último remarcar que las comunicaciones externas se llevan a cabo con el uso del protocolo HTTPS que sirve para una transferencia de datos segura entre cliente y servidor, esta seguridad se logra mediante la encriptación de datos la cual permite proteger la integridad y privacidad de la información.

Por último en cuanto a la estructura de directorios se ha llevado a cabo siguiendo el modelo hexagonal adaptado que se ha empleado y consta de un directorio padre que puede ser en este caso /routine para las rutinas de entrenamientos y con subdirectorios para los DTOs necesarios /dto y tres archivos principales que son los módulos, repositorios y controlador. Un ejemplo práctico de lo que se ha llevado a cabo en PowerLog es el siguiente:

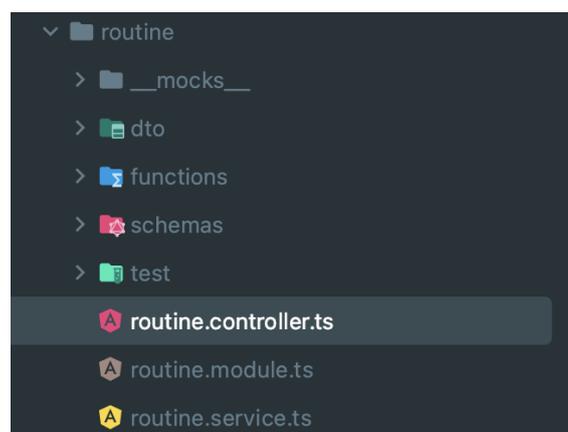


Figura 5.9: Ejemplo estructura de directorios para las rutinas de entrenamiento.

Otros ejemplos de código que se pueden añadir para demostrar las funcionalidades pueden ser por ejemplo:

```
Future<void> loginWithEmailAndPassword({
  required String email,
  required String password,
}) async {
  try {
    await _firebaseAuth.signInWithEmailAndPassword(
      email: email, password: password);
  } catch (e) {
    rethrow;
  }
}

Future<void> logOut() async {
  try {
    await Future.wait([_firebaseAuth.signOut()]);
  } catch (e) {
    rethrow;
  }
}
```

Figura 5.10: Login y LogOut de la app móvil mediante el cliente de Google Firebase.

En este fragmento de código se demuestra como la aplicación móvil se conecta con el cliente de firebase[4] para realizar las acciones de login con email y contraseña y logout, las comprobaciones son transparentes al usuario ya que es firebase[4] el encargado de realizarlas.

```
void updateDayLog(DayLog training) async {
  final response = await http.put(Uri.parse('${backendUrl}/api/routine'),
    headers: <String, String>{
      'Content-Type': 'application/json; charset=UTF-8',
    },
    body: training.toJson());

  if (response.statusCode == 200) {
  } else {
    throw Exception('Failed to update daylog.');
  }
}
```

Figura 5.11: Actualizar un entrenamiento desde la app móvil.

En este fragmento de código se muestra como se realiza una actualización del entrenamiento desde el punto de vista de la aplicación móvil ya que previamente se ha presentado desde el backend.

```
Future<DayLog> getDayLog(DateTime selectedDate, String backendUrl) async {
  String parsedDate = parseDate(selectedDate);
  final String userId = AuthRepository().getUserUid() as String;
  final response = await http.post(
    Uri.parse('$backendUrl/api/routine/date'),
    headers: <String, String>{
      'Content-Type': 'application/json; charset=UTF-8',
    },
    body: jsonEncode(<String, String>{
      'date': parsedDate,
      'athleteUid': userId,
    }),
  );

  if (response.statusCode == 200) {
    try {
      DayLog day = DayLog.fromJson(response.body);
      return day;
    } catch (e) {
      return DayLog.empty();
    }
  } else {
    throw Exception('Failed to find daylog.');
  }
}
```

Figura 5.12: Consultar un entrenamiento desde la app móvil.

```
@Post("/date")
@HttpCode(HttpStatus.OK)
async findByDate(@Body() date: GetDayDto) {
  const resp = await this.routineService.findByDate(
    date.date,
    date.athleteUid,
  );
  if (!resp) {
    throw new HttpException(
      "No day found for this date",
      HttpStatus.NOT_FOUND,
    );
  } else {
    return resp;
  }
}
```

Figura 5.13: Devolver el entrenamiento asignado a un día desde el backend.

Por último estos dos ejemplos son muy visuales ya que se puede comprobar como se realiza una petición desde la aplicación del móvil enviando la fecha y al uid del atleta y el backend devuelve la respuesta en caso de encontrarla.

## 5.4 Implantación

---

En este apartado se va a desarrollar en detalle como se ha realizado la implantación de la solución, es decir como se ha pasado de un diseño a un resultado final el cual nos permite obtener resultados de la solución final. Este proceso implica el lanzamiento del backend en Digital Ocean[6] y la distribución de las versiones de android y de iOS. Este proceso incluye pruebas exhaustivas para resolver los diferentes bug que han ido apareciendo con el lanzamiento de las versiones.

La aplicación móvil se ha realizado con el framework flutter[11], desarrollado anteriormente, pero a modo de resumen permite mediante una única base de código desarrollar aplicaciones con un funcionamiento prácticamente nativo. Para las primeras distribuciones se utilizó el comando "flutter build apk" el cual genera un archivo .apk en la raíz del proyecto que consiste en una versión compilada del mismo de manera gratuita y sencilla, esto permitió a los primeros usuarios el testeo de la aplicación para transmitir el feedback a los desarrolladores. Una vez probada esta versión en android, mediante el uso de la cuenta de desarrollador de apple se distribuyó una versión de iOS la cual tras 24h aproximadamente de revisión por los desarrolladores de apple pasaba a estar disponible para los usuarios en una app llamada TestFlight[14] la cual notifica a los usuarios que se encuentra una app nueva para descargar o una nueva versión de una que ya tengan instalada. En concreto en el desarrollo de este proyecto, la primera vez que se envió una versión a apple no fue aceptada ya que había que permitir unos permisos los cuales necesita la app aunque realmente la aplicación no los maneje, en el caso de PowerLog fueron unos permisos de notificaciones que pese a que la app no los utilice se tienen que proporcionar desde la dashboard de apple store connect[9] que es desde donde se gestiona la compilación.

Para el backend se ha utilizado el framework NestJs[10] que utilizando Typescript nos permite crear backends de manera sencilla y de fácil mantenimiento. Para la implantación de este backend y que los usuarios pudieran utilizarlo se ha empleado Digital Ocean[6] como se ha comentado previamente, para ello se adquirió un plan que se adaptaba a los requerimientos del proyecto. En primer lugar lo que se buscaba era que estuviera activo la mayor cantidad de tiempo posible, aunque fuéramos conscientes de que un usuario no iba a entrenar por la noche que tuviera la capacidad de transcribir los resultados por la app en el momento del día que quisiera, este plan contaba con un tiempo activo del 99,99% lo que hacía que fuera extremadamente raro que no estuviera funcionando. También se configuraron otros aspectos como CPU y RAM ya que era una máquina virtual pero para esto se escogieron los más básicos ya que PowerLog es una aplicación muy ligera y tener excesiva potencia aumentaría el coste. También se hicieron pruebas para depositar videos y fotos en el Space que ofrece Digital Ocean[6] pero al final se descartó la idea, el coste normal eran 6€ al mes y cuando estuvimos haciendo pruebas con archivos ascendió a 12€ al mes. El backend cuenta con el dominio <https://powerlogapi.manueljishi.dev> y está conectado Github[19] que es la herramienta utilizada para llevar a cabo el control de versiones y sirve para automatizar el despliegue cuando se realizan cambios en el repositorio, el backend está actuando en el puerto 3000 del servidor de Digital Ocean[6] y se ha empleado un sistema de reverse proxy con la herramienta NGINX[7] que nos permite redirigir el tráfico a los puertos 80 para HTTP y 443 para HTTPS que son los estándares.

---

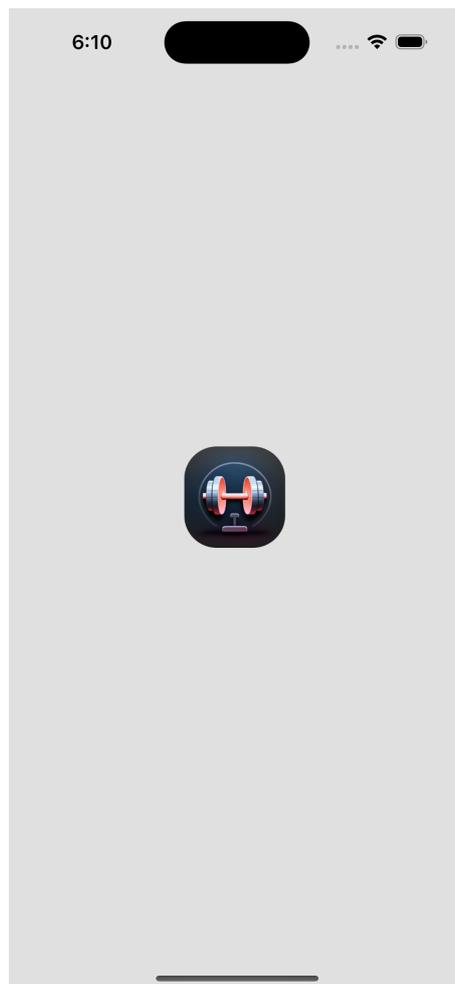
## CAPÍTULO 6

# Producto desarrollado

---

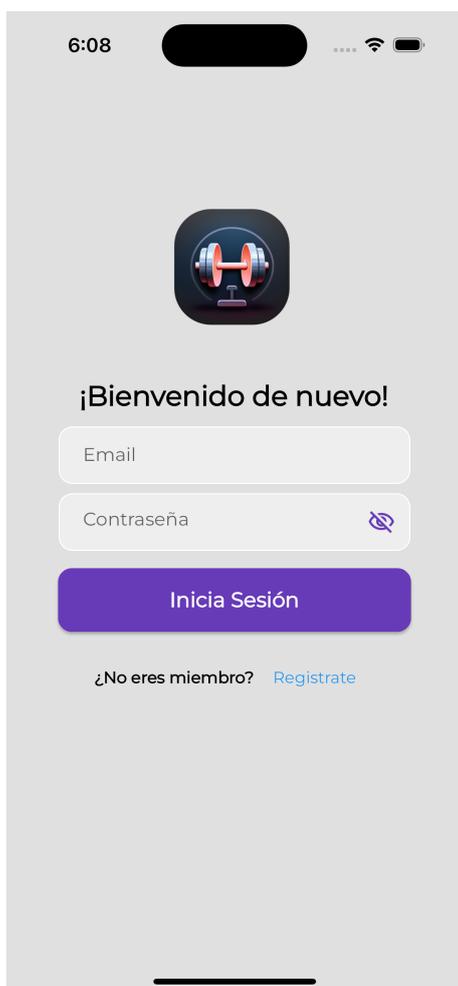
En este apartado se va a desarrollar las diferentes pruebas del resultado final de la aplicación, es decir como todo conectado en su conjunto ha dado el resultado esperado.

En primer lugar se ha creado una pantalla de carga para que la espera a que toda la aplicación cargue sea lo más amena posible y haga que el usuario no vea una pantalla en blanco simplemente, lo que mejora la experiencia de usuario.

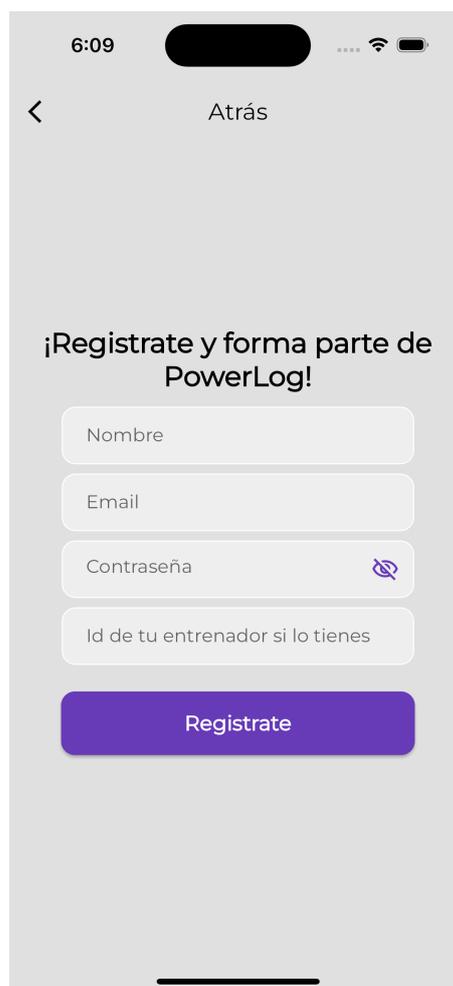


**Figura 6.1:** Pantalla de carga de PowerLog

Una vez pasada la pantalla de carga, que varía la duración dependiendo de la potencia del dispositivo, el usuario tiene dos rutas posibles. Si ya ha iniciado sesión previamente y no ha cerrado sesión la aplicación le dirige directamente a la pantalla principal y si no le dirige a la pantalla de iniciar sesión. En el caso de la pantalla de iniciar sesión tiene dos campos, uno contraseña y otro email para realizar el inicio con firebase[4] como se ha comentado en puntos previos, en caso de no tener cuenta puede el usuario pulsar el botón de registrarse y le lleva a la pantalla para hacerlo.



**Figura 6.2:** Pantalla de inicio de sesión.



**Figura 6.3:** Pantalla de registro.

Una vez el usuario tiene cuenta la app le dirige al apartado de calendario donde podrá gestionar su rutina, por defecto aparece el día actual seleccionado en el calendario y existen dos variantes, la primera de ellas es que el usuario no cuente con un entrenamiento para el día seleccionado, lo cual la app muestra un texto de aviso y en caso de si tener entrenamiento lo muestra en forma de tabla.



Figura 6.4: Pantalla de calendario con entrenamiento disponible.



Figura 6.5: Pantalla de calendario sin entrenamiento.

En caso de existir un entrenamiento, al usuario se le muestra una tabla con tres columnas que vienen predefinidas por el entrenador y dos que debe rellenar el atleta las cuales son:

- **Set:** representa el número de serie o las veces que tiene que repetir el ejercicio, en la figura de la imagen el primer ejercicio cuenta con tres series.
- **Reps:** representa el número de veces que tiene que realizar la acción del ejercicio, en caso de ser una sentadilla por ejemplo indicaría al usuario cuantas veces tiene que subir y bajar.
- **Restricciones:** representa como quiere el entrenador que realices el ejercicio y existen tres posibilidades, KG si el entrenador quiere que el atleta mueva un peso en específico, rir si el entrenador quiere que hagas tantas repeticiones hasta quedarte a tantas como indique el RIR de tu máximo, por poner un ejemplo si el entrenador marca rir 1 el atleta debe quedarse a 1 de su máximo, y por último RPE que representa el esfuerzo percibido por el atleta y se mide del 1 al 10, por ejemplo si el entrenador anota rpe 8 el atleta debe hacer el ejercicio con un peso que para él represente un esfuerzo 8/10.
- **Kg:** representa el peso que el atleta ha utilizado finalmente para realizar la serie, ya que puede encontrarse mejor o peor en el entrenamiento y variar ligeramente con las anotaciones del entrenador.

- **Intensidad:** En caso de que las restricciones sean RPE o KG la intensidad que debe anotar el atleta se va a medir en RPE y funciona de la misma manera que la del entrenador, en caso de que las restricciones que proporciona el entrenador sea RIR, la intensidad que debe rellenar se medirá en RIR también.

Cuando el atleta ha finalizado su entrenamiento y pulsa el botón de Finalizar, la app le lleva a la pantalla de resumen donde el usuario puede comprobar todas las series que ha realizado y el volumen total de peso, junto con un input para añadir comentarios y una escala para marcar las sensaciones del entrenamiento.



**Figura 6.6:** Pantalla de resumen de un entrenamiento.

A continuación se presenta la pantalla de ajustes de la cuenta donde el usuario puede cerrar sesión, modificar su nombre dentro de la app y reestablecer contraseña. En primer lugar la pantalla de ajustes tiene una vista general:

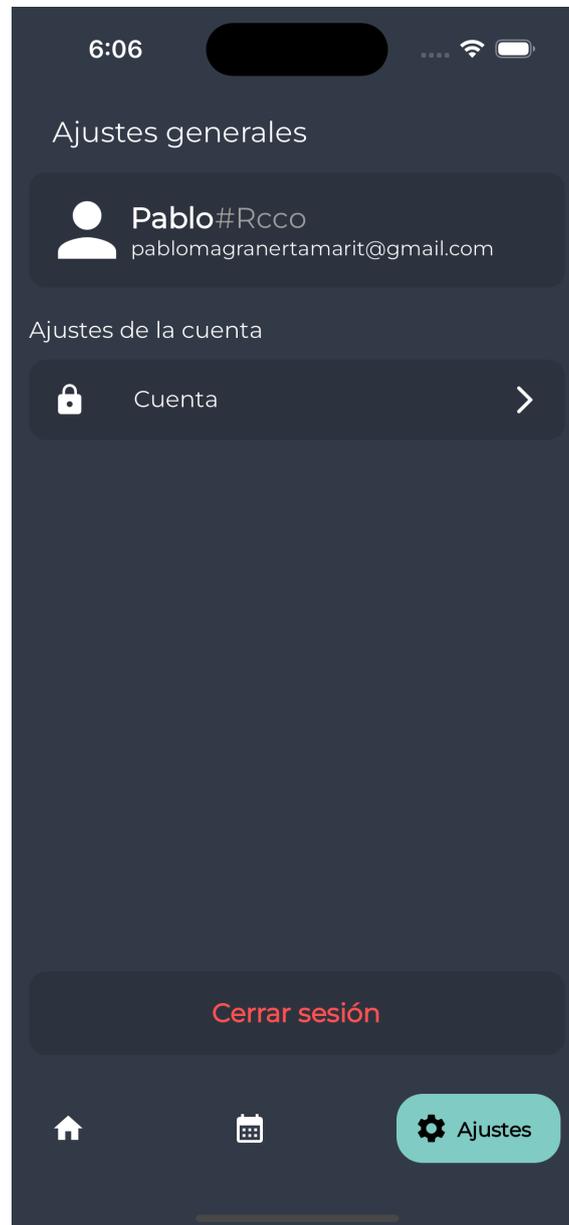
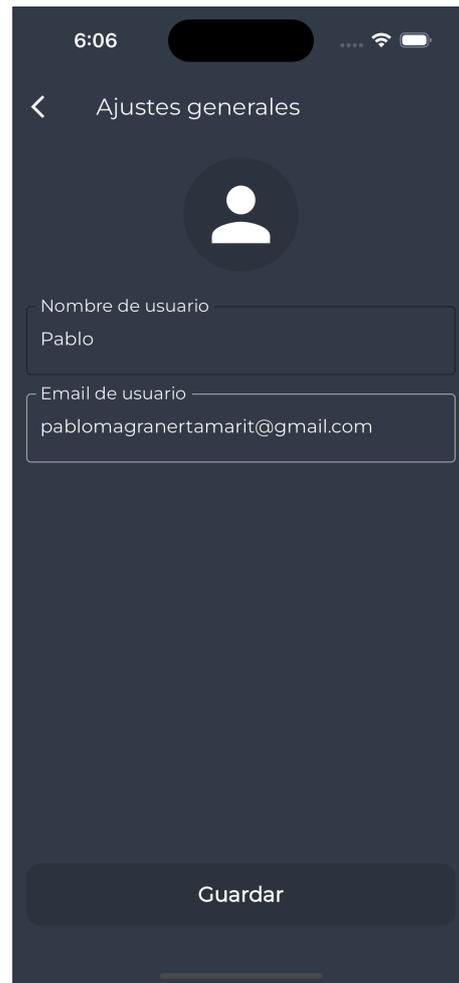


Figura 6.7: Pantalla de mi cuenta.

Si se pulsa sobre el nombre del usuario se puede entrar a la pantalla de detalles para cambiar el nombre de usuario, esta pantalla tiene dos estados, una con el botón activado si el usuario ha escrito un nombre distinto al que tenía previamente o desactivado si el usuario no ha cambiado nada.



**Figura 6.8:** Pantalla de detalles con el botón activo.



**Figura 6.9:** Pantalla de detalles con el botón inactivo.

Por último si el usuario pulsa el botón de cerrar sesión aparece una diálogo que pregunta al usuario si está seguro de su acción y en caso de confirmar cierra sesión y le lleva a la pantalla de login.

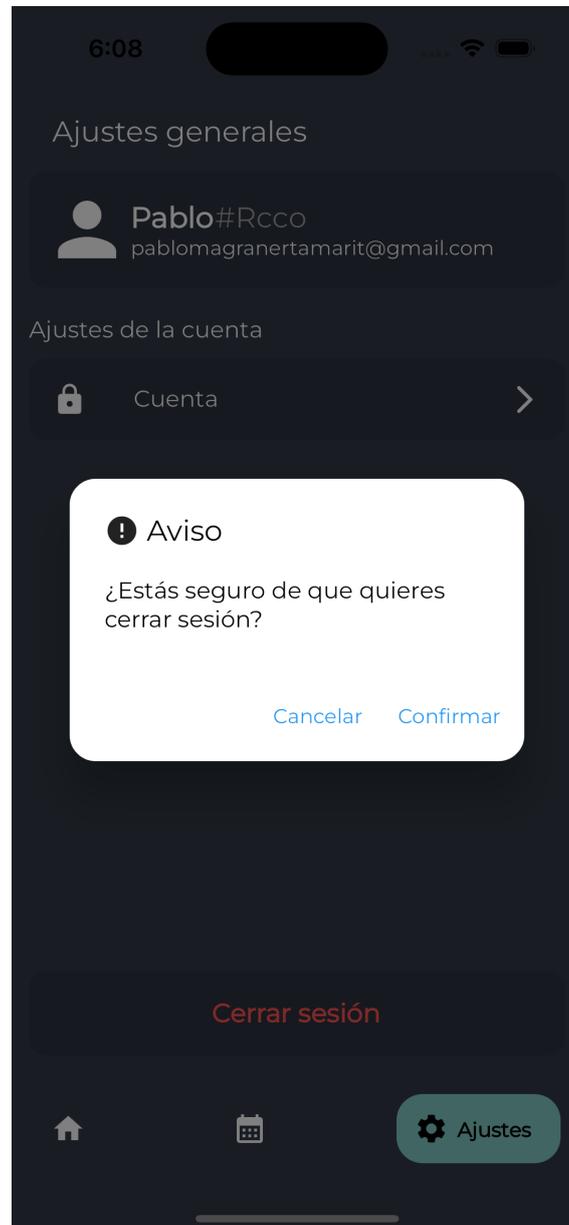
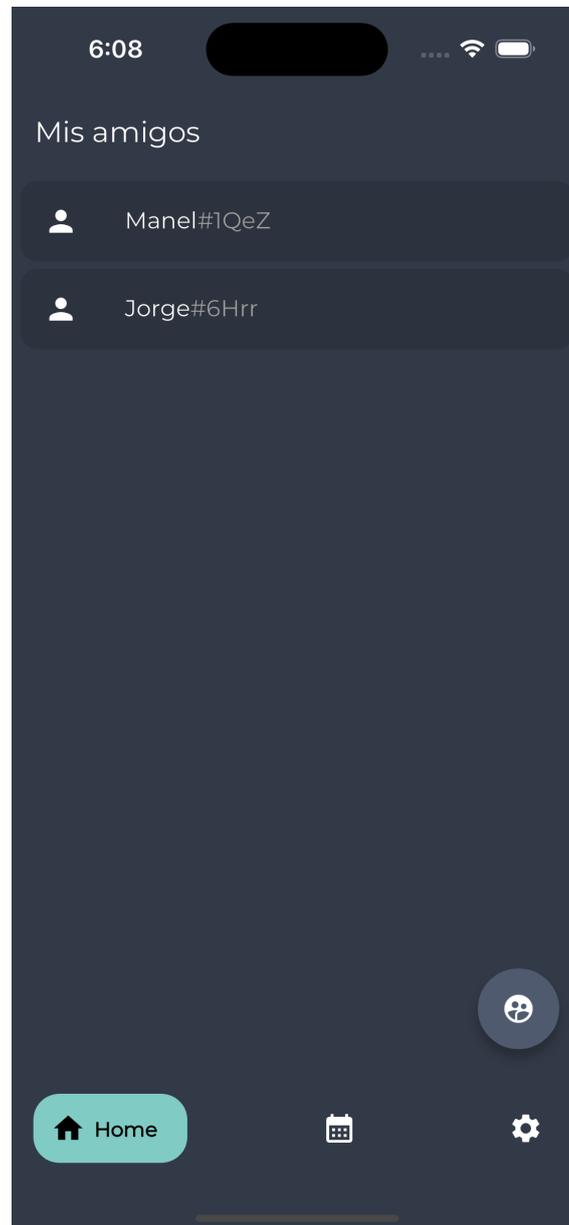


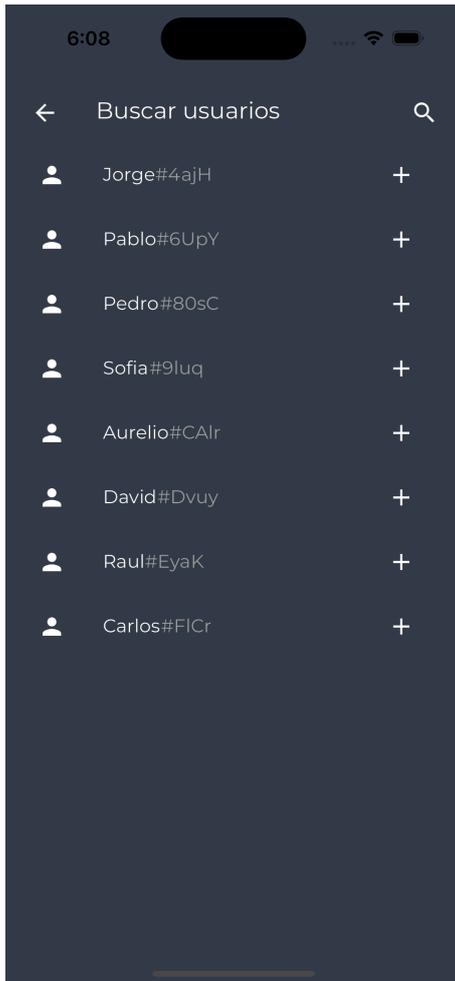
Figura 6.10: Diálogo de aviso de cerrar sesión.

Respecto a la pantalla de la lista de amigos el usuario puede comprobar que personas tiene agregadas y puede viajar a la pantalla de buscar nuevos usuarios para agregarlos.

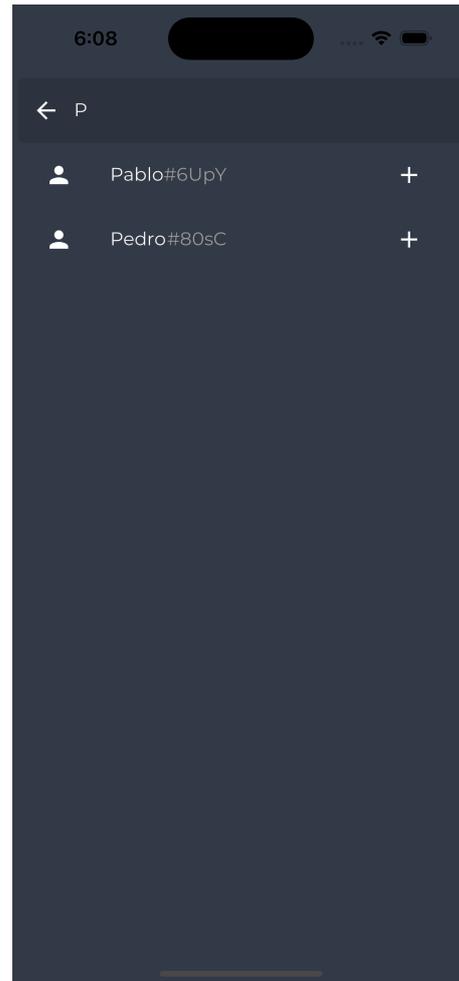


**Figura 6.11:** Pantalla del listado de amigos.

Si el usuario quiere agregar nuevos amigos a la lista puede viajar a la pantalla de buscar usuarios donde puede utilizar la barra de buscador y encontrar nuevos usuarios.



**Figura 6.12:** Pantalla con el listado de todos los usuarios.



**Figura 6.13:** Ejemplo de uso del buscador.

Referente al backend y como conclusión final al apartado de pruebas conviene presentar resultados de las bases de datos de mongoDB[5] como prueba del uso de la aplicación. Se han realizado diversas consultas con el objetivo de obtener el número de entrenamientos totales registrados en el aplicación y cual es el atleta que más entrenamientos ha modificado, es decir que ha completado.

## daylogs

STORAGE SIZE: 804KB   LOGICAL DATA SIZE: 2.94MB   TOTAL DOCUMENTS: 1219   INDEXES TOTAL SIZE: 60KB

**Figura 6.14:** Entrenamientos totales registrados en PowerLog.

The image shows two screenshots of a MongoDB aggregation pipeline. The first screenshot, labeled 'Stage 1 \$group', shows a query that groups data by athlete. The aggregation result shows two groups: one with an athleteCount of 122 and another with an athleteCount of 133. The second screenshot, labeled 'Stage 2 \$sort', shows the same data sorted by athleteCount in descending order. The aggregation result shows one group with an athleteCount of 142 and another with an athleteCount of 133.

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: {athleteUID : "$athleteUID"},
7   athleteCount: {$sum: 1},
8 }
9
10
```

Aggregation result

```
{ "_id": "Object", "athleteCount": 122 }
{ "_id": "Object", "athleteCount": 133 }
```

```
1 /**
2  * Provide any number of field/order pairs.
3  */
4 {
5   athleteCount: -1
6 }
```

Aggregation result

```
{ "_id": "Object", "athleteCount": 142 }
{ "_id": "Object", "athleteCount": 133 }
```

**Figura 6.15:** Atleta que más entrenamientos ha modificado.

En la primera de las imágenes se puede comprobar como la aplicación actualmente cuenta con 1219 entrenamientos guardados en la base de datos, en la segunda de las capturas se ven los 4 atletas que más entrenamientos han completado desde la app móvil, el máximo con un total de 142 y la aplicación cuenta con un total de 15 usuarios activos que crean y finalizan entrenamientos diariamente.

---

---

## CAPÍTULO 7

# Conclusiones

---

En este apartado se va a realizar la conclusión del proyecto y de la memoria. Power-Log nació con la idea de dos amigos de desarrollar una aplicación funcional tratando de suplir una necesidad que encontramos en el mundo de las apps de entrenamiento. Respecto a los objetivos planteados al inicio de la memoria se han llevado a cabo con éxito:

- Objetivo 1: Los atletas pueden crear y gestionar su cuenta desde la propia aplicación pudiendo cambiar el nombre y reestablecer su contraseña y gestionar desde ahí todas las funcionalidades de la app.
- Objetivo 2: Los usuarios pueden visualizar los entrenamientos planteados por el entrenador, rellenar los campos necesarios y al finalizar acceder al resumen donde pueden aportar comentarios y sensaciones.
- Objetivo 3: Los usuarios cuentan con un listado de amigos agregados y buscar nuevos para agregarlos.
- Objetivo 4: La app se ha creado con la metodología de Desarrollo Centrado en el Usuario, la cual ha permitido que se adapte perfectamente a las necesidades de los usuarios y sea sencilla de utilizar para todo tipo de usuarios.

Durante el desarrollo se han empleado conocimientos aprendidos a lo largo de los 4 años de grado, poniendo en práctica lo aprendido en asignaturas como DCU, GPR, DEW y TBD. El desarrollo frontend de atletas se ha llevado a cabo con flutter[11], que es un framework con el que había trabajado previamente pero no tenía un gran experiencia lo que hizo que fuera complicado el inicio ya que prácticamente todo era nuevo para mí. La parte más complicada sin lugar a dudas fue la de distribuir la aplicación en iOS ya que apple es muy estricto con sus políticas y llevar a cabo la configuración del proyecto, al ser la primera vez, no fue tarea sencilla. En lo que respecta al backend tenía muy poco contacto con las tecnologías utilizadas lo que conllevó a que en un inicio tuve que hacer una tarea de investigación y formación básica para entender como funcionaba, finalmente dió buen resultado y a lo largo del proyecto aprendí muchísimas cosas sobre el desarrollo de backends con NestJs[10]. La parte más retante fue la de alojamiento del backend en Digital Ocean[6], después de valorar varias opciones fue la más adecuada para este desarrollo, este apartado hizo que tuviera que aplicar todo lo aprendido en un entorno real.

Respecto a errores cometidos se podría remarcar que pese a haber realizado una larga etapa de análisis y diseño cuando se alcanzó el punto de desarrollo, tanto mi compañero como yo nos dimos cuenta de cosas que deberíamos haber especificado previamente y

que nos faltaron, esto se dió por la falta de experiencia en el DCU, pero después de la realización de este proyecto estos errores ya no se volverán a cometer por los conocimientos que hemos adquirido.

En conclusión el desarrollo de este proyecto ha sido muy satisfactorio ya que ver que las ideas que estaban en mi mente se han ido plasmando poco a poco y funcionando en un entorno real de usuarios. El aprendizaje de las nuevas tecnologías previamente comentadas también ha sido muy útil para mi carrera como desarrollador, ya que actualmente he realizado las prácticas en una empresa donde empleaban flutter[11] y gran parte de los problemas que he tenido con PowerLog no se han vuelto a repetir.

Por último comentar que una vez lanzada la versión final a la que da alcance este proyecto, el feedback recibido por los usuarios es bueno y han dejado de utilizar sus métodos tradicionales para utilizar PowerLog para la gestión de sus rutinas, lo cual es muy positivo y se puede concluir con que la aplicación cumple las necesidades planteadas en un inicio.

---

---

## CAPÍTULO 8

# Relación del trabajo desarrollado con las asignaturas cursadas

---

En este apartado se van a enumerar y desarrollar las asignaturas cursadas que considero que más han influido en el desarrollo del proyecto.

En primer lugar destacar DCU, la cual ha aportado las bases para llevar a cabo un buen diseño y análisis, en esta asignatura se tratan todos los conceptos necesarios y en las prácticas se llevan a cabo en proyectos lo que se asemeja mucho a lo que ha sido esta fase del desarrollo en PowerLog. Nos ha permitido realizar una aplicación siguiendo las guías de usabilidad y que por el feedback recibido cumple con los requisitos de los usuarios.

En segundo lugar destacar BDA y TBD, son dos asignaturas las cuales han establecido los pilares para gestionar bases de datos, ha sido muy útil para tener conocimientos a la hora de gestionar a atletas y entrenadores junto con sus datos.

Por último me gustaría destacar la asignatura de DEW, que pese a que no se empleen las mismas tecnologías, es una asignatura la cual te plantea retos de desarrollo de una web y de una publicación de un backend gestionado por el profesor que son las bases para el desarrollo y la publicación de un proyecto real.

En resumen esas asignaturas han influenciado en gran medida a este proyecto pero a lo largo de las asignaturas cursadas he ido aprendiendo otros conceptos como la capacidad de buscar información correcta en internet, redes de computadores, gestión de proyectos o eficiencia en los algoritmos creados.

---

---

## CAPÍTULO 9

# Trabajos Futuros

---

A lo largo del desarrollo han ido surgiendo ideas que podrían mejorar la aplicación pero por falta de tiempo no se han podido llevar a cabo. La primera de ellas era que el atleta pudiera subir vídeos de como realiza un determinado ejercicio, de esta manera el entrenador podría observar la técnica del ejercicio, lo cual puede llegar a ser un problema en atletas principiantes.

Otra de las ideas que han surgido era la mejora del componente social, ya que en estos momentos simplemente cuenta con un listado de amigos por falta de tiempo, pero hubiera sido interesante haber desarrollado un poco más este aspecto y tener la capacidad de consultar entrenamientos de otros atletas, incluso que funcione a modo de red social para que se den a conocer los entrenadores.

Otro de los aspectos que es interesante es el hecho de que el atleta pueda visualizar sus progresos de forma gráfica e incluso recibir algún tipo de recompensa si cumple con muchos días de entrenamiento sin fallar como hemos visto en apps como la de Hevy[2], la cual cuenta con un sistema de medallas y trofeos interno en la aplicación para motivar más a los usuarios.

Por último referente a la distribución y monetización sería interesante en un futuro buscar una manera de monetizar la aplicación y tratar de distribuirla a más usuarios publicándola en las tiendas de Google Play Store y App Store.

# Bibliografía

---

- [1] STRAVA. *Get more out of your workouts with custom goals, workout analysis and progress tracking* `https://www.strava.com/about`
- [2] HEVY. *A free workout tracker and planner for iOS and Android.* `https://www.hevyapp.com/about-us/`
- [3] VIRTUAGYM FITNESS. *Haciendo del mundo un lugar más sano y feliz a través de la tecnología innovadora.* `https://business.virtuagym.com/es/sobre-nosotros/`
- [4] FIREBASE. *Firestore provides tools to grow your app and business, for startups and global enterprises* `https://firebase.google.com/`
- [5] MONGODB. *Fully Managed Database Service Automated Deployments and Config* `https://www.mongodb.com/atlas/database`
- [6] DIGITAL OCEAN. *Deploy faster and scale easier with an application server that saves your team time and money* `https://www.digitalocean.com/`
- [7] NGINX. *NGINX accelerates content and application delivery, improves security, and facilitates availability and scalability* `https://www.nginx.com/`
- [8] PUB DEV. *Pub is the package manager for the Dart programming language, containing reusable libraries and packages for Flutter and general Dart programs.* `https://pub.dev/`
- [9] APPLE STORE CONNECT. *Dashboard de gestión de la cuenta de desarrollador de apple.* `https://appstoreconnect.apple.com/login`
- [10] NESTJS. *NestJS is a framework for building efficient, scalable Node.js web applications.* `https://nestjs.com/`
- [11] FLUTTER. *Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase* `https://flutter.dev/`
- [12] SKIA. *Skia is an open source 2D graphics library which provides common APIs that work across a variety of hardware and software platforms* `https://skia.org/`
- [13] TRANSPORTER. *App para enviar versiones a los desarrolladores de apple* `https://apps.apple.com/es/app/transporter/id1450874784?mt=12`
- [14] TESTFLIGHT. *App para distribuir las versiones entre los usuarios* `https://apps.apple.com/es/app/transporter/id1450874784?mt=12`
- [15] BLOC LIBRARY. *A predictable state management library for Dart.* `https://bloclibrary.dev/`
- [16] REACT. *La biblioteca para interfaces de usuario web y nativas* `https://es.react.dev/`

- 
- [17] VERCEL. *Vercel's frontend cloud gives developers the frameworks, workflows, and infrastructure to build a faster, more personalized Web.* <https://vercel.com/>
- [18] NEXTJS. *The React Framework for the Web* <https://nextjs.org/>
- [19] GITHUB. *The open source community* <https://github.com/>
- [20] INTELLIJ. *Code IDE developed by JetBrains* <https://www.jetbrains.com/es-es/idea/>

---



---

APÉNDICE A

## Anexo: Análisis de ODS

---

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

**Tabla A.1:** Tabla sobre los objetivos de desarrollo sostenible.

En esta tabla se presentan los 17 objetivos globales que establecieron las naciones unidas, se han clasificado como alto, medio, bajo o no procede en función de como afecta dicho objetivo al proyecto desarrollado en esta memoria. En primer lugar y quizá el objetivo que más predomina en este proyecto es el de la salud y el bienestar, esta aplicación va sobre entrenamientos en el gimnasio lo cual es una actividad deportiva que interfiere directamente en la salud, tanto física como mental ya que la práctica de deporte está demostrado que también ayuda a otras actividades como pensar, aprender incluso sirve para disminuir el riesgo de depresión y ansiedad, en definitiva promueve el movimiento y la práctica de ejercicio.

En cuanto a la educación si que es cierto que tiene un impacto bajo, pero no se debe dejar de lado que esta aplicación está orientada al entenamiento mediante entrenador y no todos los usuarios tienen porque ser experimentados en el ámbito del powerlifting, con lo cual usuarios primerizos van a aprender las bases del entrenamiento gracias a la ayuda de su entrenador y el uso de la aplicación.

La igualdad de género es un aspecto muy importante en la actualidad y el entrenamiento

con pesas es un deporte que cada vez se está expandiendo más a las mujeres ya que en un inicio era practicado principalmente por hombres, desde PowerLog se fomenta a que cualquier tipo de atleta entrene y se beneficie de las cosas positivas que tiene este deporte.

El consumo responsable es un aspecto de baja influencia pero que también es importante, se ha comentado previamente los aspectos positivos de la práctica de actividad física, pero los extremos no son buenos y una obsesión por mejorar las marcas o mejorar estéticamente puede llevar a trastornos como vigorexia o anorexia. Desde PowerLog se fomenta una práctica responsable, donde los usuarios compitan consigo mismos y contra sus amigos de manera sana y siempre con la finalidad de desarrollarse como personas.

El último aspecto que hacer incapié dentro de los ODS es el de acción por el clima, uno de los sistemas utilizados previamente para almacenar entrenamientos era el de papel y boli, el cual si eras una persona bastante asidua en el gimnasio podías llegar a consumir varias libretas al año, PowerLog centraliza todos los procesos y utiliza herramientas que todos los usuarios utilizamos en nuestro día a día como pueden ser ordenadores o dispositivos móviles, favoreciendo de esta manera a que se actualicen esos métodos tan arcaicos y se pasen a métodos más modernos y mas sostenibles con el medio ambiente y evitando fenómenos cada vez más preocupantes como el cambio climático.