



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

— **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de
Telecomunicación

LiftLinked: Aplicación web para aficionados al
entrenamiento y la musculación

Trabajo Fin de Grado

Grado en Tecnología Digital y Multimedia

AUTOR/A: Pinar Azorín, Juan Manuel

Tutor/a: Alcaraz Bellido, Ángel Mario

CURSO ACADÉMICO: 2023/2024



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

— **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

A mis padres por dar su vida para que yo pueda vivir la mía



Citas

"No cuentes los días, haz que los días cuenten." — Muhammad Ali

Agradecimientos

En primer lugar, quiero agradecer a mis amigos, que, aunque llegaron tarde, lo hicieron para quedarse. A pesar de las dudas, han hecho que cada año de carrera haya valido la pena.

También quiero agradecer a los profesores que verdaderamente sienten pasión por su trabajo y se esfuerzan día a día por inspirar a sus alumnos. Gracias por su compromiso y por ser una fuente constante de motivación.

Por último, agradecer a mi tutor por la paciencia a lo largo del año y la motivación constante a pesar de los problemas que tuve al inicio.

Resumen

LiftLinked es una aplicación diseñada para deportistas y entusiastas del fitness y la musculación. Ofrece una red social donde los usuarios pueden interactuar, compartir progresos y conectarse con otros deportistas. Además, permite crear y unirse a comunidades basadas en intereses comunes. La aplicación incluye herramientas para el diseño y seguimiento de rutinas, permitiendo a los usuarios personalizar sus entrenamientos y registrar su progreso, detallando aspectos como la frecuencia y la intensidad de cada ejercicio. También está pensada para entrenadores personales, quienes pueden crear rutinas específicas para sus clientes.

LiftLinked ha sido desarrollada utilizando React para el frontend, proporcionando una interfaz de usuario interactiva y dinámica, y Firebase para el backend, ofreciendo una base de datos en tiempo real y servicios de autenticación seguros. Aunque es una aplicación web, su diseño está optimizado para dispositivos móviles, asegurando accesibilidad y comodidad para los usuarios durante sus entrenamientos.

En conclusión, LiftLinked combina elementos sociales con herramientas avanzadas de entrenamiento, creando una plataforma integral que facilita tanto la interacción entre usuarios como el seguimiento detallado de su progreso físico.

Palabras clave: Fitness; musculación; red social; entrenamientos; personalización; comunidades; React; Firebase; diseño móvil; seguimiento de progreso.



Abstract

LiftLinked is an application designed for fitness enthusiasts and bodybuilders. It offers a social network where users can interact, share progress, and connect with other athletes. Additionally, it allows users to create and join communities based on common interests. The app includes tools for designing and tracking workout routines, enabling users to personalize their training and record their progress, detailing aspects like frequency and intensity for each exercise. It is also tailored for personal trainers, who can create specific routines for their clients.

LiftLinked was developed using React for the frontend, providing an interactive and dynamic user interface, and Firebase for the backend, offering a real-time database and secure authentication services. Although it is a web application, its design is optimized for mobile devices, ensuring accessibility and convenience for users during their workouts.

In conclusion, LiftLinked combines social elements with advanced workout tools, creating a comprehensive platform that facilitates both user interaction and detailed tracking of their physical progress.

Keywords: Fitness; bodybuilding; social network; workouts; personalization; communities; React; Firebase; mobile design; progress tracking.

Resum

LiftLinked és una aplicació dissenyada per a esportistes i entusiastes del fitness i la musculació. Ofereix una xarxa social on els usuaris poden interactuar, compartir progressos i connectar-se amb altres esportistes. A més, permet crear i unir-se a comunitats basades en interessos comuns. L'aplicació inclou eines per al disseny i seguiment de rutines, permetent als usuaris personalitzar els seus entrenaments i registrar el seu progrés, detallant aspectes com la freqüència i la intensitat de cada exercici. També està pensada per a entrenadors personals, que poden crear rutines específiques per als seus clients.

LiftLinked ha estat desenvolupada utilitzant React per al frontend, proporcionant una interfície d'usuari interactiva i dinàmica, i Firebase per al backend, oferint una base de dades en temps real i serveis d'autenticació segurs. Encara que és una aplicació web, el seu disseny està optimitzat per a dispositius mòbils, assegurant accessibilitat i comoditat per als usuaris durant els seus entrenaments.

En conclusió, LiftLinked combina elements socials amb eines avançades d'entrenament, creant una plataforma integral que facilita tant la interacció entre usuaris com el seguiment detallat del seu progrés físic.

Paraules clau: Fitness; musculació; xarxa social; entrenaments; personalització; comunitats; React; Firebase; disseny mòbil; seguiment de progrés.

RESUMEN EJECUTIVO

La memoria del TFG del Grado en Tecnología Digital y Multimedia debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la tecnologías digitales y multimedia

CONCEPT (ABET)	CONCEPTO (traducción)	¿Cumple? (S/N)	¿Dónde? (páginas)
1. IDENTIFY:	1. IDENTIFICAR:		
1.1. Problem statement and opportunity	1.1. Planteamiento del problema y oportunidad	S	1-9
1.2. Constraints (standards, codes, needs, requirements & specifications)	1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones)	S	5-8
1.3. Setting of goals	1.3. Establecimiento de objetivos	S	2
2. FORMULATE:	2. FORMULAR:		
2.1. Creative solution generation (analysis)	2.1. Generación de soluciones creativas (análisis)	S	10-26
2.2. Evaluation of multiple solutions and decision-making (synthesis)	2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis)	S	26-44
3. SOLVE:	3. RESOLVER:		
3.1. Fulfilment of goals	3.1. Evaluación del cumplimiento de objetivos	S	44
3.2. Overall impact and significance (contributions and practical recommendations)	3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas)	S	44-46



Índice

Capítulo 1.	Introducción.....	1
1.1	Contexto y motivación	1
1.2	Objetivos.....	2
1.2.1	Objetivo principal.....	2
1.2.2	Objetivos específicos	2
1.3	Planificación temporal del proyecto	2
Capítulo 2.	Estado del Arte	5
2.1	Análisis de la competencia	5
2.1.1	Harbiz	5
2.1.2	Strong:	5
2.1.3	Hevy:	6
2.2	Tecnologías Utilizadas	6
2.3	Diferenciación de LiftLinked	6
Capítulo 3.	Análisis del problema	8
3.1	Requisitos funcionales.....	8
3.1.1	Visualización de posts	8
3.1.2	Mensajería	8
3.1.3	Diseño de rutinas	8
3.1.4	Visualización de rutinas.....	8
3.1.5	Comunidades	8
3.1.6	Perfil	9
3.2	Requisitos no funcionales.....	9
3.3	Especificaciones técnicas	9
Capítulo 4.	Solución propuesta	10
4.1	Arquitectura de la aplicación.....	10
4.1.1	Arquitectura del cliente – FrontEnd	10
4.1.2	Arquitectura del servidor – BackEnd	11
4.2	Casos de uso	12
4.3	Diseño del modelo de datos.....	17
4.4	Diseño de la interfaz	18
4.4.1	Diagrama de flujo	18
4.4.2	Prototipos.....	19



4.5	Línea estética	23
4.5.1	Paleta de colores	23
4.5.2	Tipografía	24
4.5.3	Logotipo y marca.....	24
Capítulo 5.	Desarrollo de la solución	27
5.1	Tecnologías empleadas.....	27
5.1.1	Visual Studio Code.....	27
5.1.2	GitHub	27
5.1.3	Firebase.....	27
5.1.4	React	27
5.1.5	Create React App (CRA).....	28
5.1.6	React Router v6 (RRD@6).....	28
5.1.7	React Icons	28
5.1.8	Postman	28
5.2	API Wger.....	28
5.3	Firebase.....	29
5.4	Pantallas finales	29
5.4.1	Home	30
5.4.2	Buscar usuarios.....	30
5.4.3	Mensajes:.....	30
5.4.4	Diseño de rutinas	30
5.4.5	Mis rutinas	31
5.4.6	Comunidades	32
5.4.7	Perfil	32
Capítulo 6.	Conclusiones y trabajo futuro.....	45
6.1	Conclusiones.....	45
6.2	Asignaturas relacionadas	45
6.3	Costes del trabajo	46
6.3.1	Costes materiales	47
6.3.2	Costes indirectos.....	47
6.4	Trabajo futuro	48



Índice de figuras

Figura 1. Diagrama de Gantt con la planificación del proyecto.....	4
Figura 2. Esquema de la estructura de los directorios del frontend.....	11
Figura 3. Modelo de datos de LiftLinked.....	17
Figura 4. Diagrama de flujo de LiftLinked.....	18
Figura 5. Prototipo de la pantalla “Home”.....	19
Figura 6. Prototipo de la pantalla mis rutinas.....	20
Figura 7. Prototipo de la pantalla "Diseño de rutinas".....	20
Figura 8. Prototipo de la visualización de la rutina.....	21
Figura 9. Prototipo de la pantalla "Perfil".....	21
Figura 10. Prototipo de la pantalla "Comunidades".....	22
Figura 11. Paleta de colores de LiftLinked.....	23
Figura 12. Tipografía de LiftLinked (Montserrat).....	25
Figura 13. Tipografía de LiftLinked (Open Sans).....	25
Figura 14. Logos de la aplicación LiftLinked.....	26
Figura 15. Barra de navegación de la aplicación (Navbar).....	33
Figura 16. Pantalla de inicio de sesión.....	33
Figura 17. Pantalla de registro.....	33
Figura 18. Pantalla de la visualización del entrenamiento enlazado a un post.....	34
Figura 19. Pantalla Home.....	34
Figura 20. Pantalla de vista previa de perfil.....	35
Figura 21. Vista previa de pantalla "Buscar usuarios".....	35
Figura 22. Vista previa de la pantalla "Mis chats".....	36
Figura 23. Ejemplo de chat privado entre usuarios.....	36
Figura 24. Campos del diseñador de rutinas.....	37
Figura 25. Selección de usuario para la rutina.....	37
Figura 26. Pop up que muestra el listado de ejercicios.....	37
Figura 27. Opción para ver tan sólo los ejercicios propios del usuario.....	38
Figura 28. Pop up para crear un ejercicio propio.....	38
Figura 29. Vista previa de una rutina.....	39
Figura 30. Ejercicios añadidos a la rutina en desarrollo.....	39
Figura 31. Pop up para crear un temporizador.....	40
Figura 32. Registro de una rutina.....	40



Figura 33. Chat comunitario.....	41
Figura 34. Menú desplegable con el listado de comunidades.	41
Figura 35. Menú desplegable con el listado de miembros de la comunidad selec	42
Figura 36. Pop up para crear una comunidad.	42
Figura 37. Vista previa del perfil de una comunidad.	43
Figura 38. Pop up para buscar comunidades.	43
Figura 39. Opciones del perfil.	44
Figura 40. Pantalla "Pefil"	44



Índice de tablas

Tabla 1. Tabla de objetivos Sprint 1.....	2
Tabla 2. Tabla de objetivos Sprint 2.....	2
Tabla 3. Tabla de objetivos Sprint 3.....	3
Tabla 4. Tabla de objetivos Sprint 4.....	3
Tabla 5. Tabla de objetivos Sprint 5.....	3
Tabla 6. Tabla de objetivos Sprint 6.....	4
Tabla 7. Caso de uso 01.....	12
Tabla 8. Caso de uso 02.....	12
Tabla 9. Caso de uso 03.....	13
Tabla 10. Caso de uso 04.....	13
Tabla 11. Caso de uso 05.....	13
Tabla 12. Caso de uso 06.....	14
Tabla 13. Caso de uso 07.....	14
Tabla 14. Caso de uso 08.....	14
Tabla 15. Caso de uso 09.....	15
Tabla 16. Caso de uso 10.....	15
Tabla 17. Caso de uso 11.....	15
Tabla 18. Caso de uso 12.....	16
Tabla 19. Caso de uso 13.....	16
Tabla 20. Cálculo de costes del trabajo.	47

Capítulo 1. Introducción

1.1 Contexto y motivación

Como entusiasta del fitness [1] que ha practicado entrenamiento de fuerza durante los últimos años, he tenido la oportunidad de trabajar con distintos entrenadores y nutricionistas. En mi experiencia, todos utilizan algún tipo de software para gestionar sus servicios, desde hojas de cálculo en Excel hasta aplicaciones específicas centradas en la programación de dietas o entrenamientos. Sin embargo, estas herramientas suelen ser limitadas y fragmentadas, enfocándose en un solo aspecto del entrenamiento.

Una parte esencial del progreso en cualquier disciplina deportiva es el seguimiento continuo. Para mejorar en cualquier ámbito, es crucial medir y registrar el progreso a lo largo del tiempo. He utilizado desde aplicaciones hasta papel y boli para llevar a cabo un registro de mis entrenamientos. No obstante, llevar un cuaderno al gimnasio resulta incómodo, y estas aplicaciones tienden a quedarse cortas al no cubrir todas las necesidades de un usuario avanzado o que busca un enfoque más completo.

En el deporte, dos componentes clave son la motivación y la disciplina. Pertenecer a una comunidad, conectar con personas que comparten los mismos objetivos y poder compartir resultados con ellos es una forma poderosa de encontrar esa motivación necesaria para comenzar y mantener la constancia en los desafíos personales.

En los últimos años, hemos visto un crecimiento exponencial en la popularidad de los deportes relacionados con la musculación y el fitness. Este auge ha sido impulsado, en gran medida, por las redes sociales (RRSS), donde influencers [2] como Joan Pradells, Jorge Tabet y Madelman, con más de un millón de seguidores en conjunto, no han perdido la oportunidad para unirse a la tendencia de las asesorías. Estas se centran en la creación de programas de entrenamiento y dietas personalizadas. Sin embargo, estos influencers carecen de una plataforma centralizada que también permita una interacción más directa y continua con sus seguidores.

LiftLinked surge como una solución innovadora para cubrir los distintos puntos mencionados. No solo combina herramientas para la creación y seguimiento de entrenamientos, sino que también introduce un modelo de red social donde influencers y usuarios pueden interactuar más estrechamente. A través de la creación de “Comunidades” dentro de la aplicación, los usuarios pueden formar grupos, interactuar entre sí, y conectar con sus entrenadores. De esta forma LiftLinked fomenta la motivación y el entrenamiento a través de grupos con intereses comunes.

1.2 Objetivos

1.2.1 Objetivo principal

Desarrollar una aplicación web, diseñada para dispositivos móviles, que combine funcionalidades de red social y herramientas de gestión de entrenamiento para deportistas y entusiastas del fitness, con un enfoque particular en la musculación.

1.2.2 Objetivos específicos

- Creación de una aplicación web con React.js, orientada a dispositivos móviles.
- Definir el modelo y base de datos.
- Implementar métodos de autenticación seguros y eficientes.
- Diseñar las pantallas principales, definiendo sus funcionalidades principales
 - Diseñar un sistema de publicaciones
 - Diseñar un sistema de creación de rutinas y actualización de estas
 - Diseñar un sistema de interacción entre usuarios, individual y grupal
 - Diseñar un sistema de logros
- Optimización de la aplicación para dispositivos móviles

1.3 Planificación temporal del proyecto

La planificación temporal se ha basado en la metodología ágil Scrum [3]. **Scrum** es un marco de trabajo ágil que organiza el trabajo en sprints, que son ciclos de desarrollo cortos, generalmente de una a cuatro semanas. Durante cada sprint, un equipo de desarrollo trabaja en un conjunto de tareas previamente planificadas que deben ser completadas al final del sprint. El objetivo es entregar incrementos funcionales del producto de manera iterativa y continua.

A continuación, se muestran los sprints pautados.

FASE 1

SEMANA	OBJETIVOS
1	Reunión con el profesor
	Análisis de mercado
	Estudio de las tecnologías necesarias
	Planificación previa

Tabla 1. Tabla de objetivos Sprint 1.

FASE 1 Definición y planificación del proyecto

SEMANA	OBJETIVOS
2	Definición de requisitos funcionales y no funcionales
	Planificación de la arquitectura y estructura de la aplicación
	Diseño de bocetos y wireframes iniciales

Tabla 2. Tabla de objetivos Sprint 2.

FASE 2 IMPLEMENTACIÓN BÁSICA DEL BACKEND

<i>SEMANAS</i>	<i>OBJETIVOS</i>
3 y 4	Configuración previa de Firebase: autenticación, Database y Storage Implementación de la API Wger y pruebas con POSTMAN Desarrollo de la lógica para la creación y gestión de usuarios y comunidades Desarrollo de la estructura de datos de los entrenamientos y rutinas

Tabla 3. Tabla de objetivos Sprint 3.

FASE 3 DISEÑO Y DESARROLLO DEL FRONTEND

<i>SEMANAS</i>	<i>OBJETIVOS</i>
5 y 6	Implementación de la API Wger y pruebas con POSTMAN Desarrollo de la lógica para la creación y gestión de usuarios y comunidades Desarrollo de la estructura de datos de los entrenamientos y rutinas

Tabla 4. Tabla de objetivos Sprint 4.

FASE 3 DISEÑO Y DESARROLLO DEL FRONTEND

<i>SEMANAS</i>	<i>OBJETIVOS</i>
6 y 7	Revisión del diseño, comprobación de la visualización en dispositivos móviles Testeo completo de la aplicación Corrección de errores Despliegue de la aplicación con Firebase Hosting

Tabla 5. Tabla de objetivos Sprint 5.

FASE 4 CIERRE DE PROYECTO

SEMANAS

8

OBJETIVOS

Documentación del proyecto

Revisión con el tutor

Entrega final (2-5 sept)

Tabla 6. Tabla de objetivos Sprint 6.

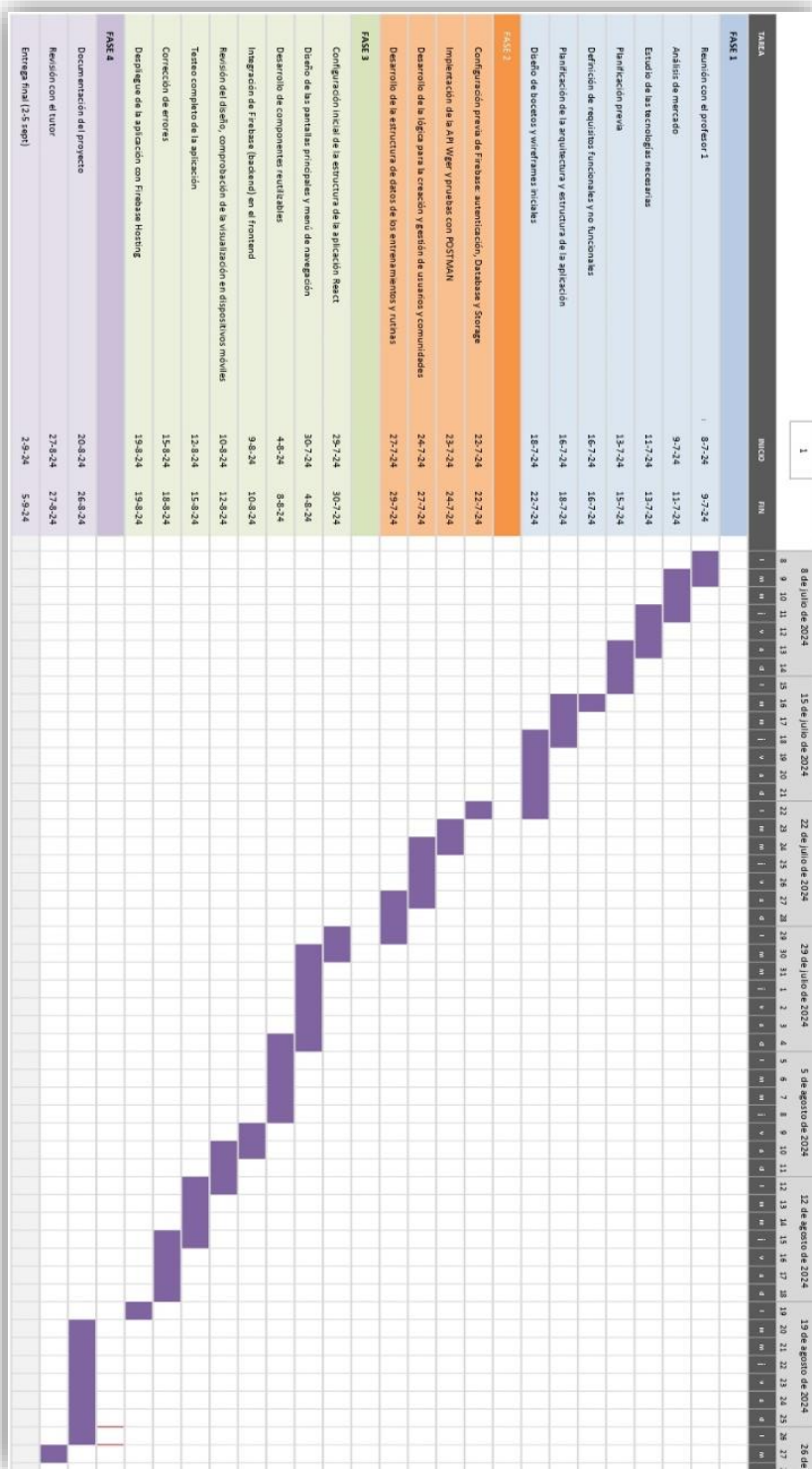


Figura 1. Diagrama de Gantt con la planificación del proyecto.

Capítulo 2. Estado del Arte

2.1 Análisis de la competencia

Para poder situar LiftLinked en el mercado se debe estudiar la competencia previamente.

En esta sección se analizará y comparará LiftLinked con las aplicaciones del mercado que más se asemejan. Entre ellas estarán Harbiz [4], Strong [5] o Hevy [6].

Se explicará brevemente cada una de las aplicaciones mencionadas, las tecnologías que emplean, su público objetivo, así como sus ventajas y limitaciones.

De esta manera, se podrá llevar a cabo una comparativa entre nuestra aplicación y el resto, ayudándonos a potenciar sus puntos fuertes y proteger los débiles.

2.1.1 Harbiz

- **Características:** Harbiz es una aplicación de fitness centrada en ofrecer a los entrenadores y nutricionistas una plataforma para gestionar sus clientes y sus progresos. Ofrece funcionalidades como la creación de planes de entrenamiento y dietas, y comunicación directa con los clientes a través de la aplicación. Su enfoque está dirigido principalmente a nutricionistas y/o entrenadores personales.
- **Público Objetivo:** Entrenadores personales, nutricionistas, y profesionales del fitness.
- **Ventajas:** Proporciona una plataforma para la gestión de clientes y planes personalizados, ideal para aquellos que buscan combinar entrenamiento y nutrición.
- **Limitaciones:** Carece de un enfoque en la comunidad o en la interacción social entre los usuarios, y su uso está más orientado hacia profesionales que hacia individuos que buscan un seguimiento personal o motivación de una comunidad.

2.1.2 Strong:

- **Características:** Strong es una aplicación dedicada exclusivamente al diseño y registro de entrenamientos o rutinas. Permite a los usuarios registrar sus rutinas de manera detallada, incluyendo repeticiones, series y pesos.
- **Público Objetivo:** Deportistas que desean un seguimiento detallado de su entrenamiento.
- **Ventajas:** Ofrece una interfaz intuitiva para registrar entrenamientos y ver el progreso a lo largo del tiempo. Es ideal para usuarios que buscan una herramienta sencilla y enfocada para el registro de ejercicios.
- **Limitaciones:** No ofrece una red social integrada ni funcionalidades de comunidad, lo que limita la interacción entre usuarios. Además, no tiene opciones para entrenadores o influencers que desean gestionar grupos o clientes. Su plan gratuito es muy limitado.

2.1.3 Hevy:

- **Características:** Hevy es una aplicación de seguimiento de entrenamientos que también incluye algunas funcionalidades sociales, como la posibilidad de compartir entrenamientos y progresos con amigos y seguidores. Ofrece características como la creación de rutinas personalizadas, seguimiento del progreso y la posibilidad de unirse a desafíos comunitarios.
- **Público Objetivo:** Usuarios individuales interesados en un registro de entrenamiento detallado, con un leve componente social.
- **Ventajas:** Combina el registro de entrenamiento con funciones sociales ligeras, permitiendo a los usuarios seguir a amigos y compartir progresos.
- **Limitaciones:** La funcionalidad social es limitada en comparación con una red social completa. No proporciona una plataforma robusta para la creación de comunidades o la gestión de clientes por parte de entrenadores.

2.2 Tecnologías Utilizadas

Las aplicaciones mencionadas anteriormente utilizan una variedad de tecnologías para ofrecer sus servicios:

- **Frontend:** La mayoría de estas aplicaciones, como Strong y Hevy, están diseñadas utilizando tecnologías frontend modernas como React Native para asegurar una experiencia de usuario fluida y responsiva tanto en dispositivos iOS como Android.
- **Backend:** Para la gestión de datos en tiempo real y la autenticación de usuarios, es común el uso de Firebase, que proporciona una solución integral para almacenamiento, base de datos en tiempo real, y autenticación segura. Además, algunas aplicaciones utilizan AWS o Google Cloud para escalabilidad y almacenamiento de grandes volúmenes de datos.

2.3 Diferenciación de LiftLinked

LiftLinked se distingue de las aplicaciones mencionadas en varios aspectos clave:

1. **Combinación de Red Social y Fitness:** A diferencia de Strong y Hevy, que ofrecen funcionalidades de registro de entrenamiento con algunas características sociales limitadas, **LiftLinked** combina una red social completa con herramientas avanzadas de fitness. Los usuarios no solo pueden seguir y compartir sus entrenamientos, sino también formar comunidades, interactuar en grupos, y acceder a contenido exclusivo de influencers.
2. **Plataforma para Influencers y Entrenadores:** **LiftLinked** ofrece funcionalidades específicas para influencers y entrenadores personales, permitiéndoles crear comunidades, gestionar sus seguidores, y proporcionar rutinas personalizadas directamente dentro de la aplicación. Esta característica no está presente en Harbiz, Strong o Hevy, lo que hace que **LiftLinked** sea única en su enfoque hacia la monetización y la gestión de seguidores.
3. **Diseño Optimizado para Dispositivos Móviles:** Aunque todas las aplicaciones mencionadas tienen versiones móviles, **LiftLinked** ha sido diseñada desde el principio para ser mobile-first, garantizando que todas las funcionalidades sean



fácilmente accesibles en un entorno de gimnasio, donde los usuarios no tienen tiempo ni paciencia para interfaces complejas.

4. **Sistema de Logros:** Para fomentar la motivación y el compromiso, **LiftLinked** incorpora un sistema de logros y gamificación, lo que añade un elemento de desafío y recompensa que va más allá del simple seguimiento del progreso. Esto es especialmente útil para mantener a los usuarios comprometidos a largo plazo.

En conclusión, aunque existen varias aplicaciones en el mercado que abordan el registro de entrenamientos y cuentan con funcionalidades sociales, **LiftLinked** se diferencia por ofrecer una plataforma que combina ambos aspectos. Además, no solo está diseñada para entrenadores, si no que permite acercar las comunidades a los influencers, retroalimentando a todos.

Capítulo 3. Análisis del problema

En esta sección se analizará el dominio de la aplicación a desarrollar, especificando los requisitos funcionales y no funcionales.

3.1 Requisitos funcionales

3.1.1 Visualización de posts

- Los usuarios deben poder visualizar los posts de los usuarios seguidos
- Los usuarios deben poder ver el entrenamiento realizado por el publicador

3.1.2 Mensajería

- Los usuarios deben poder enviar mensajes privados a otros usuarios
- Los usuarios deben poder enviar mensajes a las comunidades de las que formen parte

3.1.3 Diseño de rutinas

- El usuario debe poder dar nombre a las rutinas
- El usuario debe poder elegir el grupo muscular representante de la rutina
- El usuario debe poder seleccionar a quién quiere dirigir la rutina (para los entrenadores)
- El usuario debe poder seleccionar ejercicios
- El usuario debe poder crear ejercicios
- El usuario debe poder definir la intensidad y volumen de los ejercicios seleccionados
- El usuario debe poder editar la rutina

3.1.4 Visualización de rutinas

- El usuario debe poder visualizar un listado con sus rutinas
- El usuario debe distinguir entre rutinas propias y asignadas
- El usuario debe poder visualizar rutinas
- El usuario debe poder registrar cambios en las rutinas
- El usuario debe poder eliminar rutinas

3.1.5 Comunidades

- El usuario debe poder crear comunidades
- El usuario debe poder buscar comunidades
- El usuario debe poder suscribirse a comunidades
- El usuario debe poder interactuar con las comunidades
- El usuario debe poder visualizar un listado de las comunidades a las que se ha suscrito
- El usuario debe poder cancelar la suscripción de una comunidad



3.1.6 Perfil

- El usuario debe poder visualizar su perfil
- El usuario debe poder visualizar sus posts
- El usuario debe poder visualizar su progreso
- El usuario debe poder visualizar el número de seguidores y seguidos
- El usuario debe poder modificar sus datos de registro

3.2 Requisitos no funcionales

- La aplicación debe cargar rápida y fluidamente
- La aplicación debe ser intuitiva y sencilla de utilizar para los usuarios
- La aplicación debe ser compatible con el mayor número de dispositivos posibles
- La aplicación debe ser capaz de manejar un gran número de usuarios y peticiones sin que afecte al rendimiento de esta

3.3 Especificaciones técnicas

- Se trata de una aplicación web diseñada con el framework de Javascript [7], React.js [8] para el frontend.
- Para la creación, almacenamiento y gestión de datos se ha usado Firebase [9].
- Para la importación de ejercicios se ha usado la API de Wger [10].
- Para la seguridad, privacidad y despliegue también se ha utilizado Firebase

Capítulo 4. Solución propuesta

En esta sección se explicará la solución propuesta, tanto su estructura como su diseño y desarrollo.

4.1 Arquitectura de la aplicación

4.1.1 Arquitectura del cliente – FrontEnd

El frontend de LiftLinked se ha construido utilizando React, uno de los frameworks más conocidos y utilizados de JavaScript. React se centra en la creación de componentes reutilizables, que son bloques de construcción independientes que pueden combinarse para formar la interfaz de usuario completa. Su capacidad para manejar el DOM de manera eficiente a través de un Virtual DOM, junto con la gestión de estados y propiedades, permite que las actualizaciones de la interfaz sean rápidas y reactivas, mejorando significativamente la experiencia del usuario.

Elementos principales del frontend:

- **Componentes:** Los componentes son las piezas fundamentales de la interfaz en React. En LiftLinked, han sido utilizados para generar elementos como botones, formularios, tarjetas de perfil, y elementos más complejos como feeds de publicaciones. Son altamente reutilizables, lo que significa que el mismo componente puede ser usado en diferentes partes de la aplicación, manteniendo una apariencia y comportamiento consistentes. Además, permiten un mayor modularidad del código, haciéndolo más limpio y práctico.
- **Páginas:** Representan las vistas principales de la aplicación, en LiftLinked habrá cinco pantallas principales: como
 - Inicio (Home),
 - Diseño de rutinas (Routine Designer)
 - Mis rutinas (My Routines)
 - Comunidades (Communities)
 - Perfil (User Profile).

Cada página combina componentes con la lógica específica de esa vista, como la obtención de datos desde el backend.

- **Estilo:** Para la estética de la aplicación se han utilizado CSS [11] y Tailwind [12]. CSS permite crear hojas de estilos para cada componente o página, de forma más general, para mantener una línea estética de forma simple. Lo que evita conflictos de estilo y facilita el mantenimiento. Tailwind, por otro lado, es un framework de CSS que permite aplicar clases predefinidas directamente a los elementos de React, lo que permite un diseño rápido y más concreto, dando como resultado un diseño más personalizado a cada página.

Para asegurar una experiencia fluida y eficiente, LiftLinked utiliza **React Router DOM** para gestionar la navegación sin necesidad de recargar la página, lo que mejora la rapidez y la interacción. Además, el uso de **APIs** y **hooks de React** como `useState`, `useEffect` y `useContext` permite manejar eficazmente los estados y los efectos secundarios, como la obtención de datos o la actualización de la interfaz en tiempo real. Estas herramientas

aseguran que la aplicación sea reactiva y que los datos se sincronicen correctamente con el backend, ofreciendo una experiencia de usuario óptima y actualizada al instante.

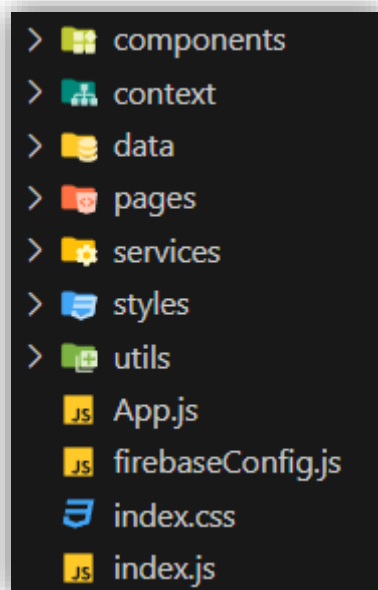


Figura 2. Esquema de la estructura de los directorios del frontend.

4.1.2 Arquitectura del servidor – BackEnd

El backend de LiftLinked está desarrollado con Firebase, una plataforma proporcionada por Google diseñada para crear aplicaciones web y móviles. Firebase ofrece un entorno integral que proporciona todas las herramientas necesarias para construir el backend de una aplicación, lo que facilita el desarrollo al tener todo centralizado en un solo lugar. Además, su interfaz es muy intuitiva y accesible, lo que la convierte en una excelente opción para desarrolladores principiantes y experimentados.

Componentes del Backend:

- **Firestore Authentication:** Gestiona la autenticación de usuarios mediante diferentes proveedores (correo electrónico y contraseña, Google, Facebook, etc.). Esta autenticación es crucial para manejar el acceso seguro a la aplicación y sus diferentes funcionalidades.
- **Firestore:** Una base de datos NoSQL en tiempo real que almacena los datos de la aplicación, como información de usuarios, publicaciones, rutinas de entrenamiento, comunidades, etc. Firestore permite realizar consultas en tiempo real, lo que es fundamental para la actualización instantánea de los feeds y los chats.
- **Storage:** Utilizado para almacenar archivos como imágenes de perfil, imágenes de publicaciones y otros archivos multimedia subidos por los usuarios. Este servicio asegura que los archivos sean accesibles de manera eficiente y segura.

4.2 Casos de uso

A continuación, se explicarán un gran abanico de casos de usos de la aplicación, que abordarán la mayoría de las situaciones de la aplicación y su consiguiente flujo.

Caso de uso - 01	Inicio de sesión
Actor	Usuario
Descripción: El usuario inicia sesión en la aplicación	
Precondiciones: El usuario debe estar previamente registrado en la aplicación	
Flujo principal: <ol style="list-style-type: none"> 1. El usuario abre la aplicación por primera vez o tras cerrar sesión. 2. El usuario introduce su correo electrónico y contraseña. 3. El sistema valida las credenciales. 4. El sistema redirige al usuario a la pantalla principal (Home). 	
Flujo alternativo: Si el usuario introduce credenciales incorrectas, el sistema muestra un mensaje de error.	
Postcondiciones: El usuario ha iniciado sesión y tiene acceso a todas las funcionalidades.	

Tabla 7. Caso de uso 01.

Caso de uso - 02	Registro
Actor	Usuario
Descripción: Un nuevo usuario se registra en la aplicación.	
Precondiciones: Acceso a Internet	
Flujo principal: <ol style="list-style-type: none"> 1. El usuario abre la aplicación y selecciona "Registrarse". 2. El usuario introduce su correo electrónico, contraseña, y otros detalles requeridos. 3. El sistema valida la información y crea una nueva cuenta de usuario. 4. El sistema redirige al usuario a la pantalla principal (Home). 	
Flujo alternativo: Si el usuario introduce información incompleta o un correo ya registrado, el sistema muestra un mensaje de error.	
Postcondiciones: El usuario tiene una nueva cuenta y está registrado en la aplicación.	

Tabla 8. Caso de uso 02.

Caso de uso - 03	Visualizar publicaciones
Actor	Usuario
Descripción: El usuario visualiza su “feed” de publicaciones en la pantalla de Home.	
Precondiciones: El usuario debe estar autenticado.	
Flujo principal:	
<ol style="list-style-type: none"> 1. El usuario accede a la pantalla Home. 2. El sistema muestra las publicaciones de los usuarios a los que sigue. 3. El usuario puede desplazarse por el “feed” para ver más publicaciones. 	
Flujo alternativo: Si el usuario no sigue a ningún otro usuario, se muestra vacío	
Postcondiciones: El usuario visualiza el contenido actualizado de su red.	

Tabla 9. Caso de uso 03.

Caso de uso - 04	Buscar y seguir usuarios
Actor	Usuario
Descripción: El usuario busca otros usuarios y decide seguirlos.	
Precondiciones: El usuario debe estar autenticado.	
Flujo principal:	
<ol style="list-style-type: none"> 1. El usuario accede a la función de búsqueda en Home. 2. El usuario introduce un nombre de usuario o “nickname”. 3. El sistema muestra los resultados de la búsqueda. 4. El usuario selecciona un perfil. 5. Se muestra la vista previa del perfil. 6. El usuario pulsa el botón “Seguir” 	
Flujo alternativo: Si no se encuentran coincidencias, el sistema muestra un mensaje de "No se encontraron resultados".	
Postcondiciones: El usuario sigue ahora al usuario mostrado y podrá ver sus publicaciones	

Tabla 10. Caso de uso 04.

Caso de uso - 05	Enviar mensaje privado
Actor	Usuario
Descripción: Un usuario envía un mensaje a otro usuario	
Precondiciones: El usuario debe estar autenticado	
Flujo principal:	
<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de mensajes. 2. El usuario selecciona un chat existente o crea uno nuevo a través de la búsqueda de usuarios. 3. El usuario escribe un mensaje y pulsa "Enviar". 4. El sistema envía el mensaje al destinatario. 	
Flujo alternativo: El usuario crea el chat tras visitar el perfil de otro usuario	
Postcondiciones: El mensaje es enviado y aparece en el chat del destinatario.	

Tabla 11. Caso de uso 05.

Caso de uso - 06	Crear rutina propia
Actor	Usuario
Descripción: El usuario crea una nueva rutina de entrenamiento personalizada.	
Precondiciones: El usuario debe estar autenticado.	
Flujo principal:	
<ol style="list-style-type: none"> 1. El usuario accede a la pantalla "Diseño de Rutinas". 2. El usuario introduce el nombre de la rutina y selecciona la parte del cuerpo. 3. El usuario añade ejercicios utilizando el botón de selección de ejercicios. 4. El usuario guarda la rutina. 	
Flujo alternativo:	
Postcondiciones: La nueva rutina se guarda en la base de datos y aparece en "Mis Rutinas".	

Tabla 12. Caso de uso 06

Caso de uso - 07	Crear rutina para un suscriptor/cliente
Actor	Administrador de comunidad
Descripción: Un administrador de comunidad crea una rutina de entrenamiento personalizada para un suscriptor	
Precondiciones: El usuario debe estar autenticado.	
Flujo principal:	
<ol style="list-style-type: none"> 1. El administrador accede a la pantalla "Diseño de Rutinas". 2. El administrador selecciona la opción "Crear para suscriptor". 3. El administrador elige un suscriptor de la lista. 4. El administrador diseña y guarda la rutina. 	
Flujo alternativo:	
Postcondiciones: La rutina se asigna al suscriptor y aparece en "Mis Rutinas" del suscriptor.	

Tabla 13. Caso de uso 07.

Caso de uso - 08	Visualización y registro de entrenamiento
Actor	Usuario
Descripción: El usuario realiza una rutina y registra su progreso.	
Precondiciones: El usuario debe tener al menos una rutina en "Mis Rutinas".	
Flujo principal:	
<ol style="list-style-type: none"> 1. El usuario accede a "Mis Rutinas" y selecciona una rutina. 2. El usuario comienza la rutina y se inicia el cronómetro. 3. El usuario completa los ejercicios, registrando repeticiones y peso. 4. El usuario finaliza y guarda la rutina. 	
Flujo alternativo:	
Postcondiciones: Los valores de la rutina se actualizan y registran.	

Tabla 14. Caso de uso 08.

Caso de uso - 09	Unirse a Comunidad
Actor	Usuario
Descripción: El usuario se une a una comunidad existente.	
Precondiciones: El usuario debe estar autenticado.	
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1. El usuario accede a la pantalla de "Comunidades". 2. El usuario busca comunidades y selecciona una. 3. El usuario selecciona "Unirse a Comunidad". 	
Flujo alternativo:	
Postcondiciones: El usuario se convierte en miembro de la comunidad.	

Tabla 15. Caso de uso 09

Caso de uso - 10	Crear Comunidad
Actor	Usuario
Descripción: El usuario crea una nueva comunidad dentro de la aplicación de la que será administrador.	
Precondiciones: El usuario debe estar autenticado.	
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1. El usuario accede a la pantalla de "Comunidades". 2. El usuario selecciona "Crear Comunidad". 3. El usuario introduce los detalles de la comunidad (nombre, descripción, imagen). 4. El usuario guarda la comunidad. 	
Flujo alternativo:	
Postcondiciones: La nueva comunidad se guarda en la base de datos y aparece en la lista de comunidades del usuario.	

Tabla 16. Caso de uso 10

Caso de uso - 11	Visualizar perfil
Actor	Usuario
Descripción: El usuario visualiza su perfil	
Precondiciones: El usuario debe estar autenticado.	
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1. El usuario accede a la pantalla de "Perfil". 2. El sistema muestra la información del perfil, los logros y el progreso del usuario. 	
Flujo alternativo:	
Postcondiciones: El usuario visualiza sus datos y estadísticas personales.	

Tabla 17. Caso de uso 11.



Caso de uso - 12	Editar perfil
Actor	Usuario
Descripción: El usuario edita su información de perfil.	
Precondiciones: El usuario debe estar autenticado	
Flujo principal: <ol style="list-style-type: none">1. El usuario accede a la pantalla de "Perfil".2. El usuario selecciona "Editar Perfil".3. El usuario modifica la información (nombre, imagen, etc.).4. El usuario guarda los cambios.	
Flujo alternativo:	
Postcondiciones: La información del perfil se actualiza en la base de datos	

Tabla 18. Caso de uso 12

Caso de uso - 13	Cerrar sesión
Actor	Usuario
Descripción: El usuario cierra la sesión en la aplicación.	
Precondiciones: El usuario debe estar autenticado	
Flujo principal: <ol style="list-style-type: none">1. El usuario accede a la pantalla de "Perfil".2. El usuario selecciona "Cerrar Sesión".3. El sistema cierra la sesión y redirige a la pantalla de inicio de sesión.	
Flujo alternativo: -	
Postcondiciones: El usuario se desconecta de la aplicación.	

Tabla 19. Caso de uso 13.

4.3 Diseño del modelo de datos

Se ha organizado en varias colecciones principales, tales como users, posts, routines, communities, y messages. Cada colección tiene documentos con campos específicos que almacenan la información relevante para cada entidad.

Las colecciones más importantes serían:

- **Users:** Contiene la información básica del perfil del usuario.
- **Posts:** Almacena los datos de cada publicación del usuario.
- **Routines:** Guarda los datos relacionados con las rutinas de entrenamiento, desde los ejercicios hasta sus valores.
- **Communities:** Gestiona la información de las comunidades dentro de la aplicación.
- **Messages:** Registra las conversaciones entre usuarios o entre comunidades, difiere el campo displayName de communityName.

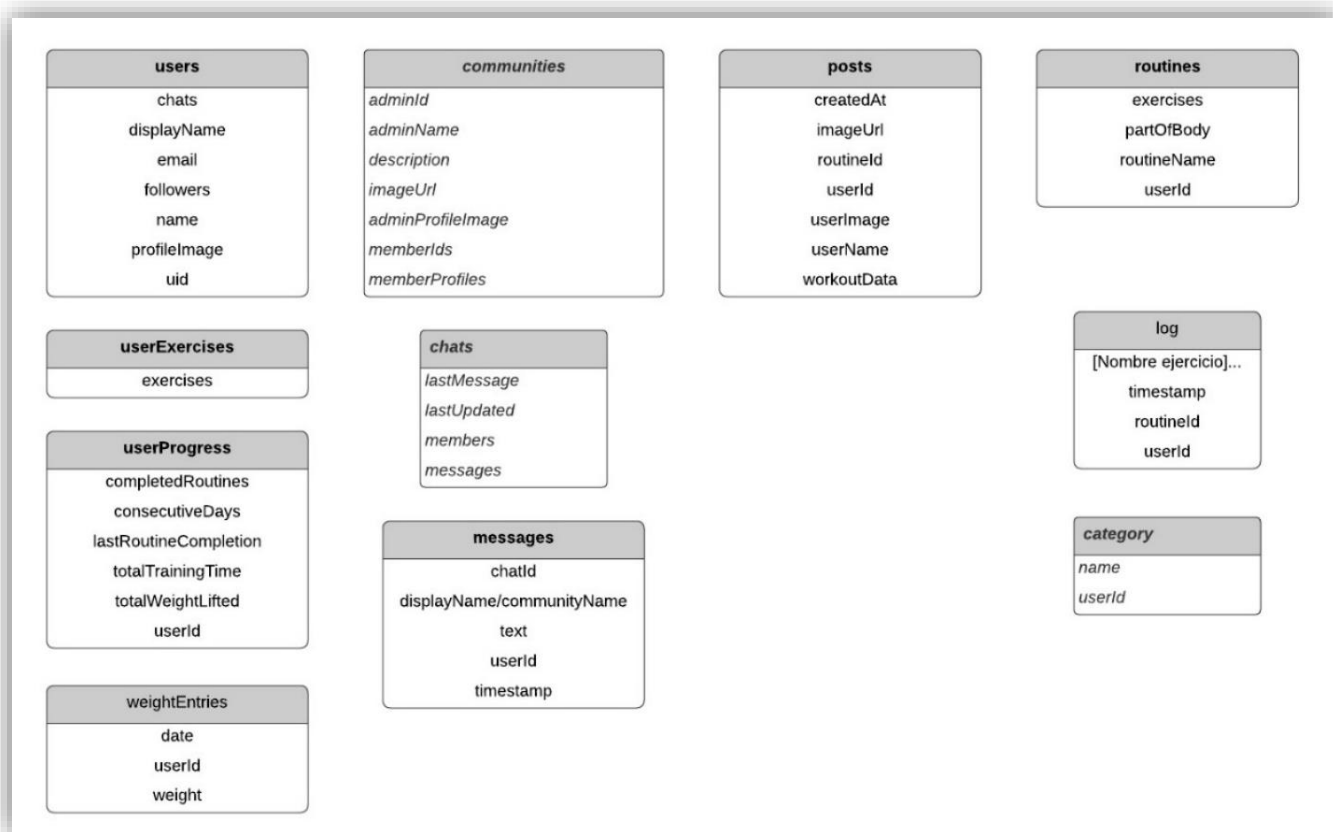


Figura 3. Modelo de datos de LiftLinked.

4.4 Diseño de la interfaz

4.4.1 Diagrama de flujo

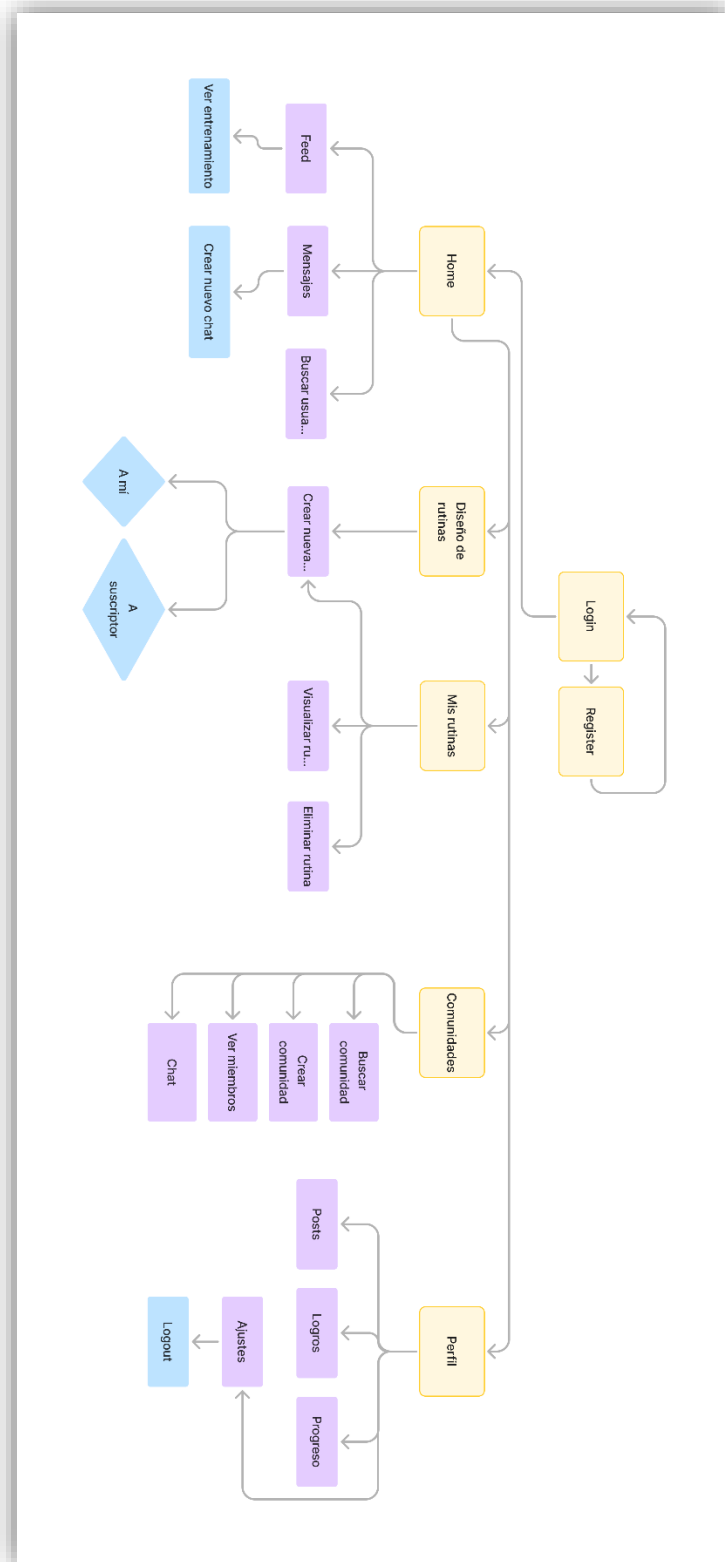


Figura 4. Diagrama de flujo de LiftLinked.

4.4.2 Prototipos

A continuación, se muestran los primeros bocetos o prototipos de la aplicación, los cuáles fueron fundamentales para plasmar las ideas iniciales y su posterior visualización, de forma que la implementación posterior fuese más simple y rápida.

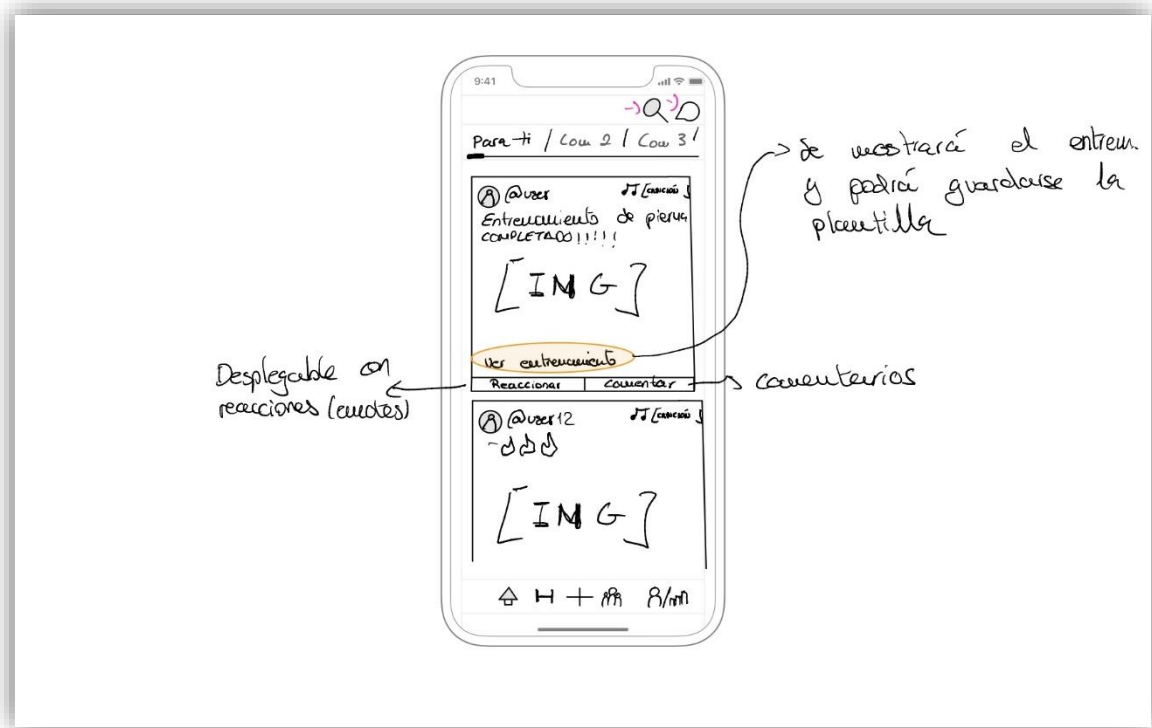


Figura 5. Prototipo de la pantalla “Home”.



Figura 6. Prototipo de la pantalla mis rutinas.

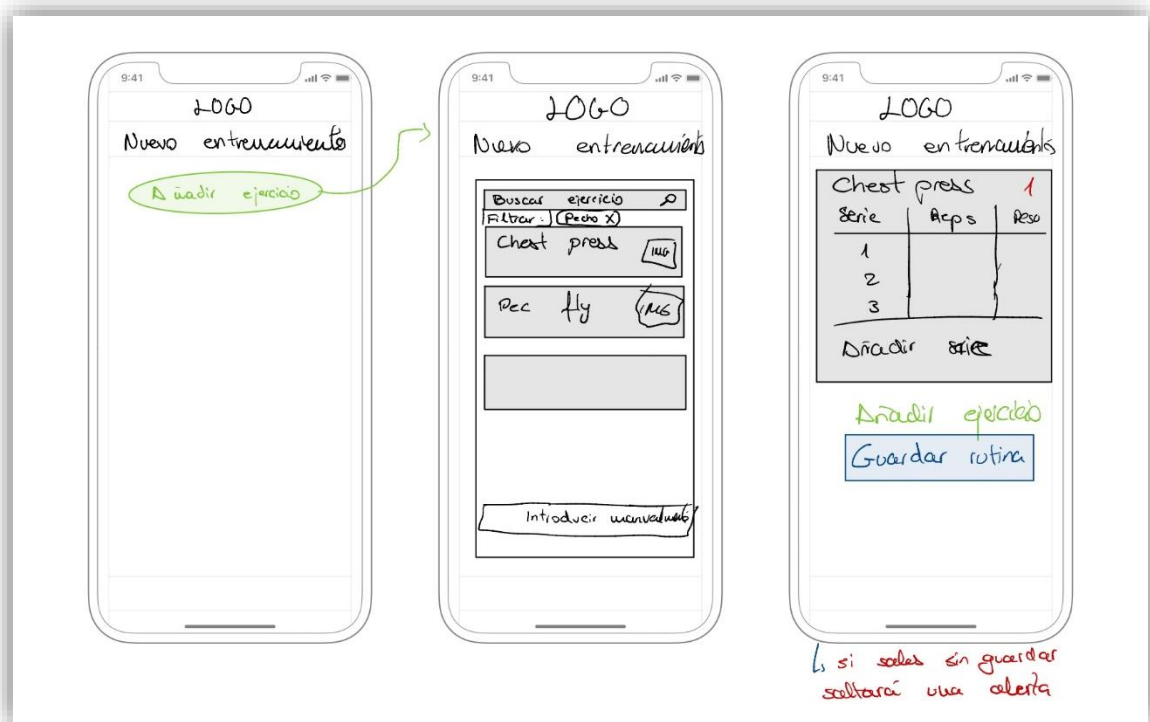


Figura 7. Prototipo de la pantalla "Diseño de rutinas"

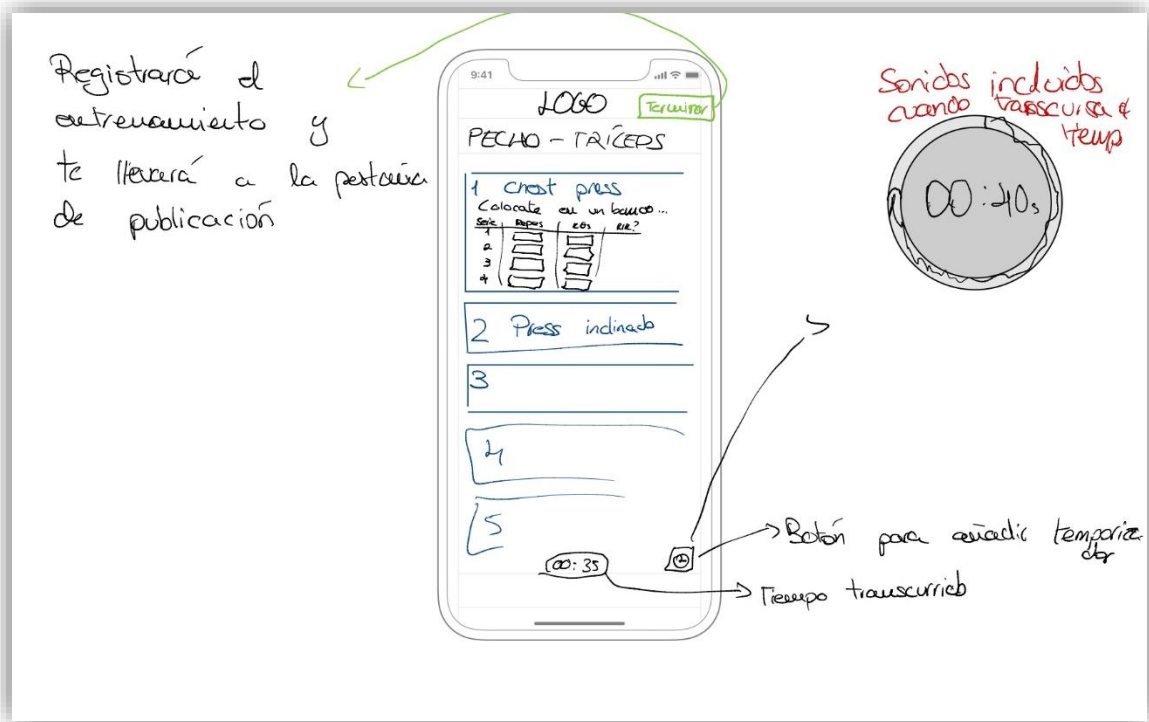


Figura 8. Prototipo de la visualización de la rutina.

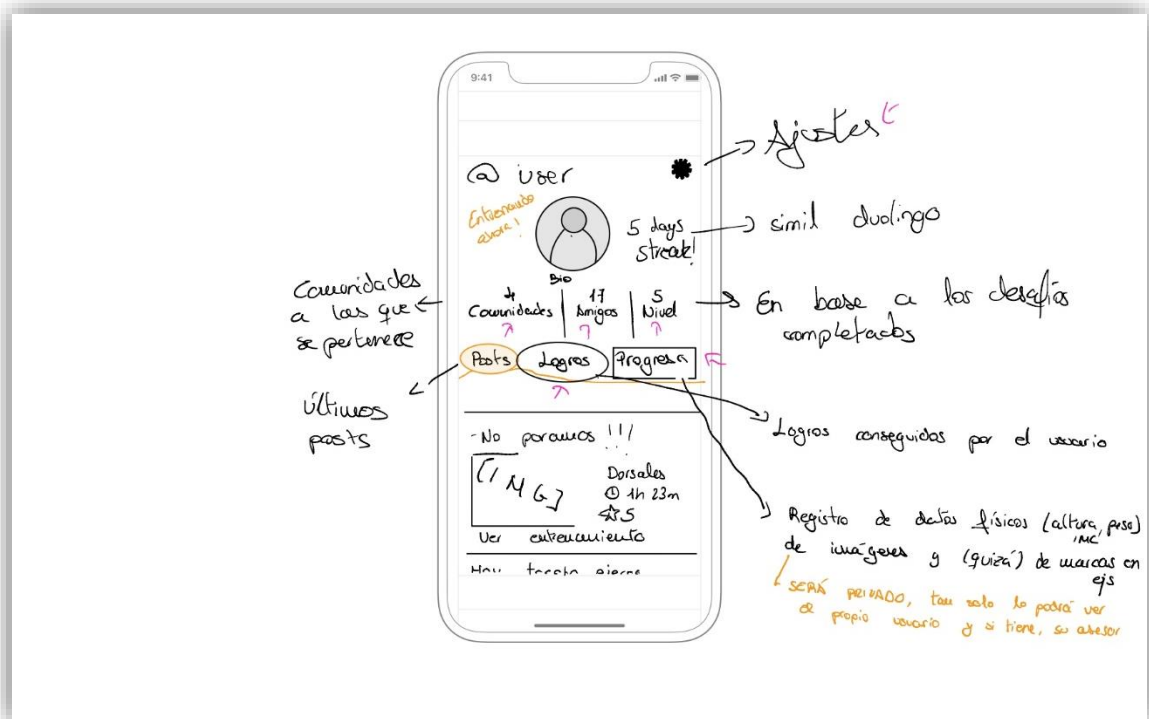


Figura 9. Prototipo de la pantalla "Perfil".

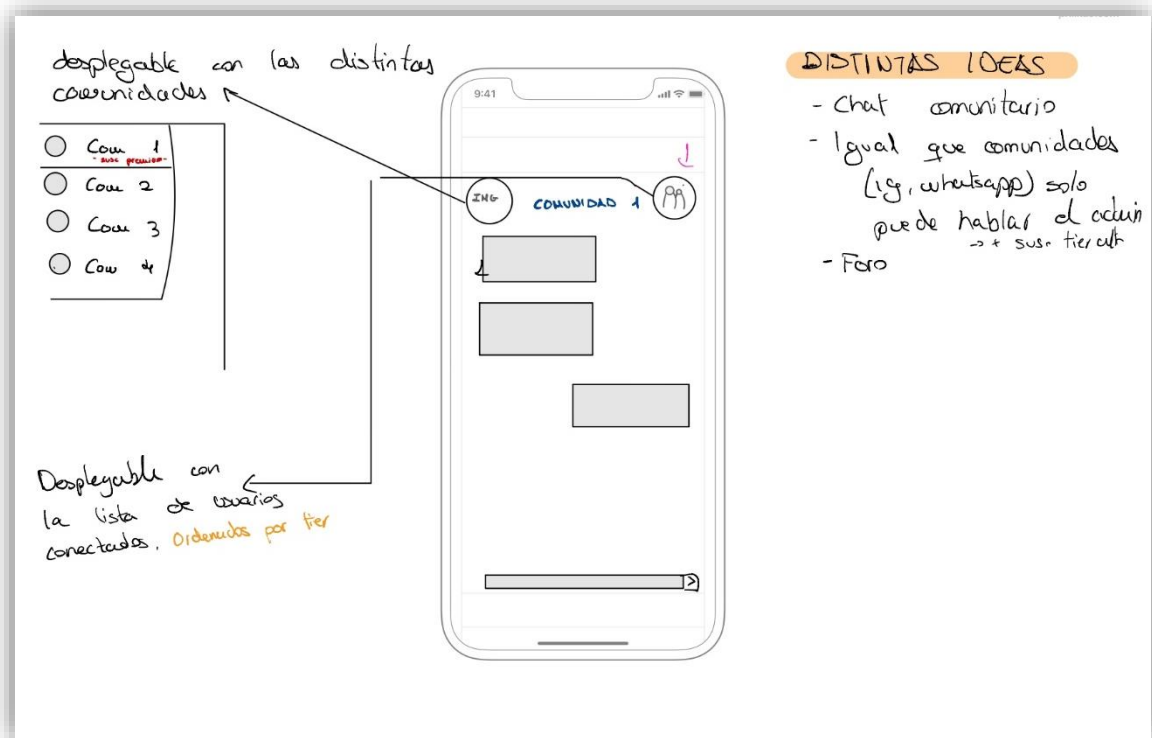


Figura 10. Prototipo de la pantalla "Comunidades".

4.5 Línea estética

4.5.1 Paleta de colores

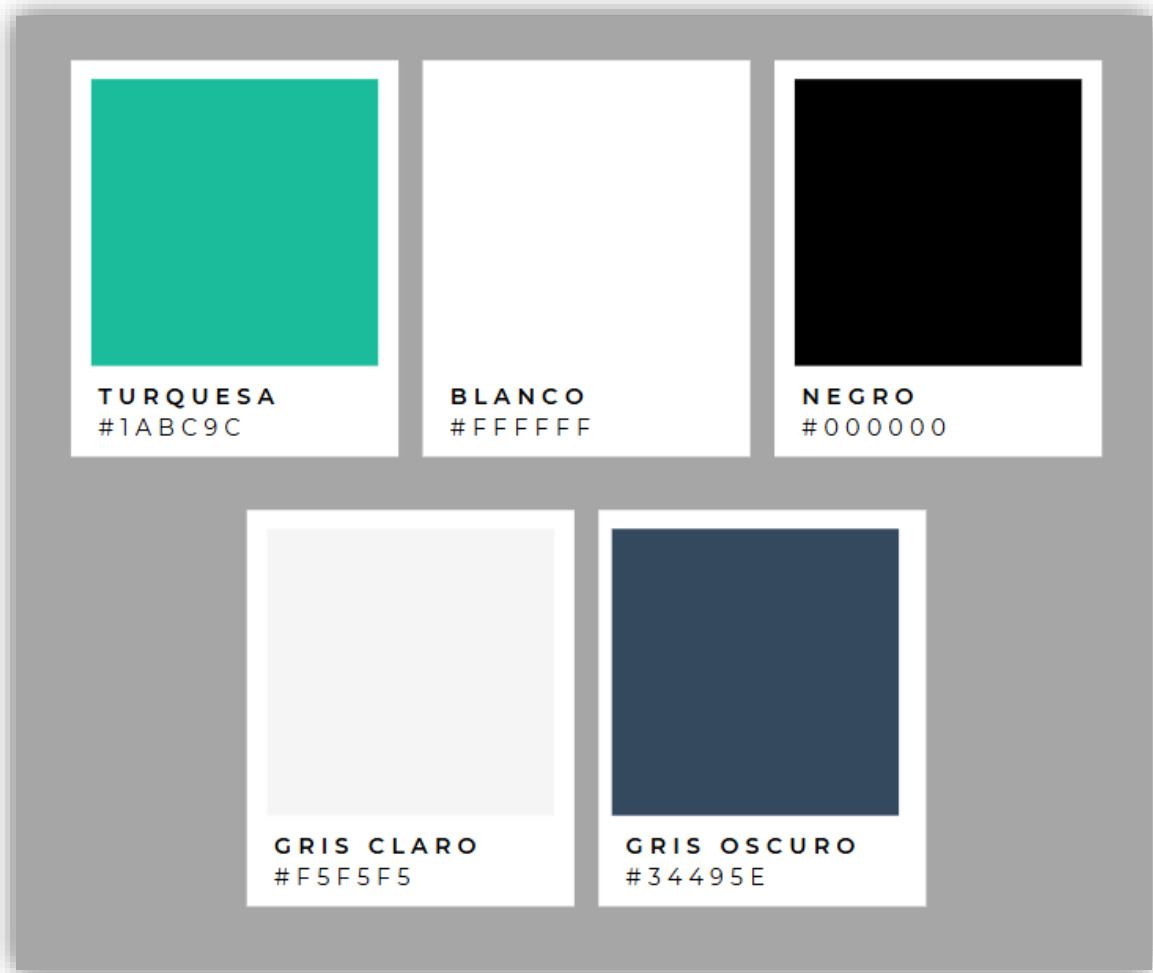


Figura 11. Paleta de colores de LiftLinked.

Turquesa: #1ABC9C

Uso y significado: Este color vibrante y refrescante se utiliza para destacar elementos clave de la interfaz, como botones, iconos y enlaces activos. El turquesa es conocido por transmitir tranquilidad, frescura y energía, atributos que son ideales para motivar a los usuarios en su camino hacia el bienestar físico. Además, es un color que resulta agradable para la vista, lo que facilita un uso prolongado de la aplicación.

Blanco: #FFFFFF

Uso y significado: El blanco es el color predominante de fondo en la aplicación, utilizado para proporcionar claridad, simplicidad y espacio visual. En el diseño de interfaces, el blanco representa pureza y enfoque, ayudando a los usuarios a concentrarse en el contenido principal sin distracciones innecesarias. Esto es especialmente importante en una aplicación de fitness donde la claridad de la información es crucial.

Negro: #000000

Uso y significado: Utilizado principalmente para el texto y detalles menores, el negro ofrece el contraste necesario para asegurar la legibilidad sobre fondos claros. Es un color clásico que aporta una sensación de profesionalismo y seriedad, adecuados para destacar información importante.

Gris Claro: #F5F5F5

Uso y significado: Este color se utiliza para diferenciar secciones secundarias y añadir contraste sutil con el fondo blanco principal. El gris claro proporciona un fondo neutro que no compete con otros elementos de la interfaz, manteniendo el diseño limpio y organizado.

Gris Oscuro: #34495E

Uso y significado: Empleado para el texto secundario o elementos que requieren un menor contraste que el negro puro, como descripciones o instrucciones. Este tono aporta una suavidad que reduce la fatiga visual y mantiene un equilibrio visual con el fondo blanco y el gris claro.

4.5.2 Tipografía

Para la tipografía de LiftLinked, se ha optado por dos fuentes complementarias que aportan tanto claridad como modernidad al diseño de la aplicación. Montserrat se ha utilizado para los títulos y encabezados debido a su estilo limpio y geométrico, que asegura una buena legibilidad y proporciona un aspecto contemporáneo. Para el cuerpo del texto y otros elementos más extensos, se ha elegido Open Sans, una fuente sans-serif que es fácil de leer en cualquier tamaño de pantalla y que mantiene una estética profesional y amigable.

4.5.3 Logotipo y marca

El diseño del logotipo de LiftLinked consistía en destacar visualmente las dos ideas clave detrás del nombre de la aplicación: "Lift" (levantar) y "Linked" (conectado). El proceso creativo comenzó con una lluvia de ideas para lograr este objetivo.

De esta lluvia de ideas obtuvimos dos conceptos clave. Utilizar una mancuerna para representar "Lift" y un enlace para representar "Linked". Tras esto comenzó el bocetaje inicial y primeros prototipos.

El resultado final es un logotipo que fusiona una mancuerna con un enlace (que sustituye el soporte de la mancuerna), combinando así la representación de fuerza y esfuerzo con la idea de conexión y comunidad.

Se han creado cuatro versiones del logotipo para asegurar su adaptabilidad a diferentes contextos y fondos. Dos de ellas son en blanco y negro, lo que permite una fácil integración. Las otras dos versiones mantienen la misma estructura, pero incorporan el color turquesa, característico de LiftLinked, sobre fondos blanco y negro. Esto garantiza

que el logotipo sea versátil, escalable y que mantenga una identidad visual fuerte y coherente en todos los usos posibles.



Figura 12. Tipografía de LiftLinked (Montserrat).

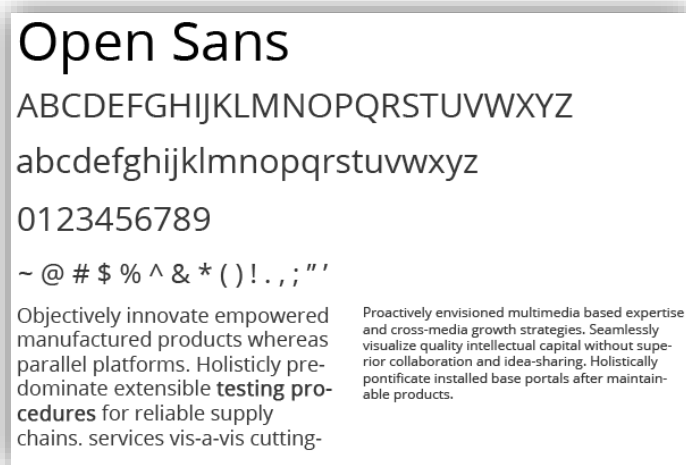


Figura 13. Tipografía de LiftLinked (Open Sans).

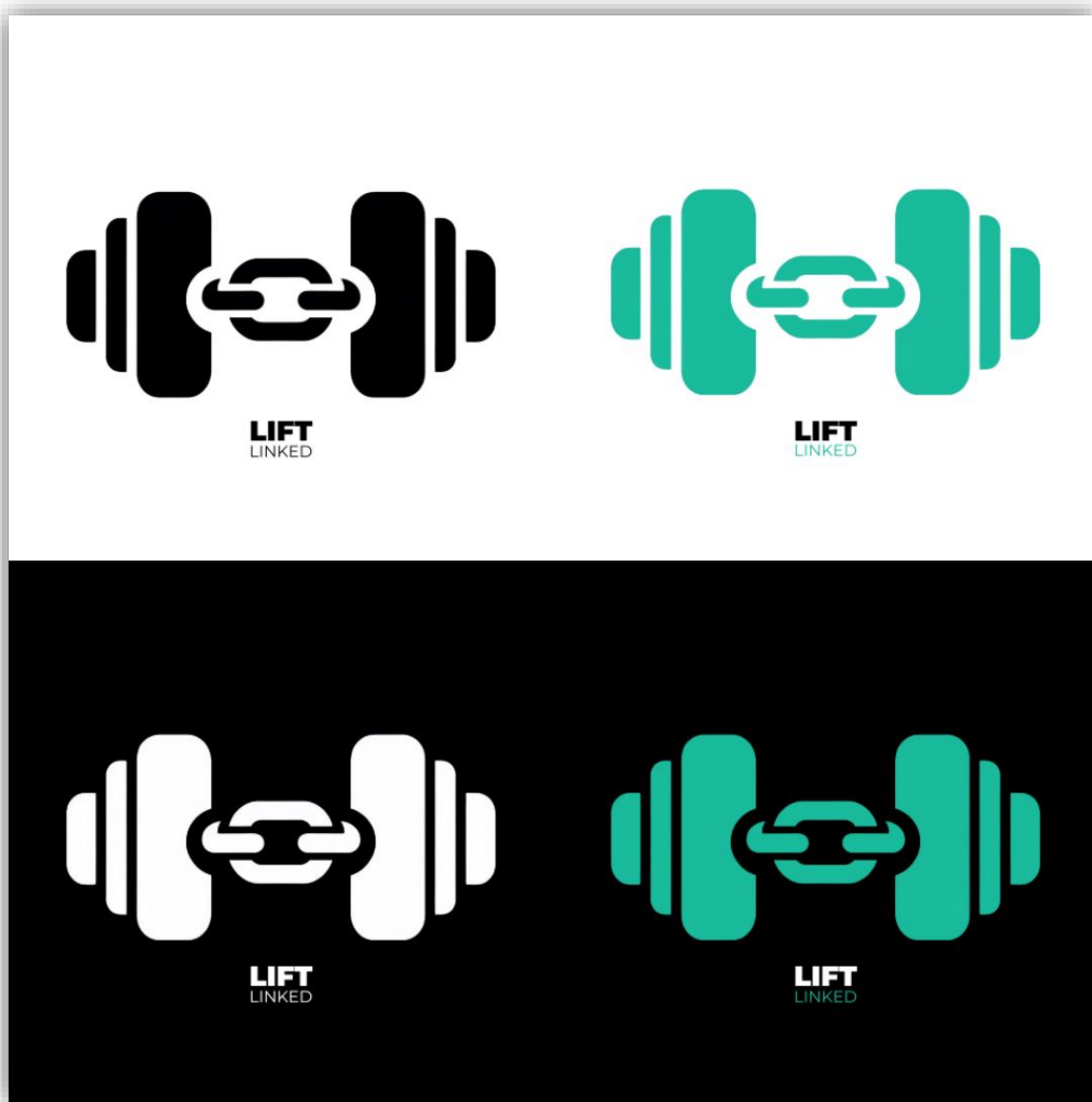


Figura 14. Logos de la aplicación LiftLinked.

Capítulo 5. Desarrollo de la solución

En este apartado se va a plantear el desarrollo de la solución propuesta y los resultados obtenidos

5.1 Tecnologías empleadas

En el desarrollo de LiftLinked, hemos utilizado una combinación de herramientas y tecnologías que optimizan tanto el proceso de desarrollo como el rendimiento de la aplicación. A continuación, se detalla cada una de las herramientas y tecnologías clave empleadas:

5.1.1 Visual Studio Code

Visual Studio Code (VS Code) [13] es un editor de código fuente desarrollado por Microsoft que se ha convertido en uno de los entornos de desarrollo más populares debido a su flexibilidad y extensibilidad. VS Code ofrece una amplia variedad de extensiones que facilitan el desarrollo en múltiples lenguajes y plataformas. Para el desarrollo de LiftLinked, se ha utilizado VS Code por su integración fluida con GitHub, sus potentes herramientas de depuración, y su capacidad para personalizar el entorno de trabajo con extensiones específicas para React y Firebase, como Prettier y ESLint, que ayudan a mantener un código limpio y consistente.

5.1.2 GitHub

GitHub [14] es una plataforma de alojamiento de código fuente y control de versiones que utiliza Git. En LiftLinked, GitHub se ha empleado para gestionar el código fuente del proyecto, facilitando el trabajo colaborativo entre desarrolladores y el control de versiones. La integración de GitHub con VS Code ha permitido un flujo de trabajo eficiente, donde se pueden realizar commits, crear ramas y resolver conflictos de manera ágil. Además, GitHub Actions se ha utilizado para automatizar procesos de CI/CD (Integración Continua y Despliegue Continuo), asegurando que cada cambio en el código se prueba y se despliega adecuadamente.

5.1.3 Firebase

Firebase es una plataforma de desarrollo de aplicaciones móviles y web proporcionada por Google que ofrece una variedad de servicios backend, como bases de datos en tiempo real, almacenamiento en la nube, y autenticación de usuarios. Para LiftLinked, Firebase ha sido esencial, ya que permite gestionar la autenticación de usuarios, almacenar y sincronizar datos en tiempo real a través de Firestore, y almacenar archivos multimedia con Firebase Storage. La integración de Firebase con React se ha realizado mediante el uso del SDK de Firebase, lo que permite una gestión simplificada y segura del backend desde el frontend.

5.1.4 React

React es una biblioteca de JavaScript para construir interfaces de usuario, especialmente de aplicaciones web de una sola página (SPA). LiftLinked utiliza React debido a su capacidad para crear componentes reutilizables y su enfoque en la gestión eficiente del DOM, lo que resulta en una experiencia de usuario rápida y reactiva. React también facilita el modularidad del código, permitiendo un desarrollo más limpio y mantenible.

5.1.5 *Create React App (CRA)*

Create React App (CRA) [15] es una herramienta que permite configurar un proyecto React con todas las configuraciones y dependencias necesarias con un solo comando. CRA ha sido utilizado en LiftLinked para inicializar el proyecto de manera rápida y efectiva, estableciendo una estructura de carpetas estándar y configurando un entorno de desarrollo optimizado sin necesidad de configuración manual. Esto ha permitido a los desarrolladores centrarse más en el desarrollo de la aplicación en sí y menos en las configuraciones del entorno.

5.1.6 *React Router v6 (RRD@6)*

React Router v6 (RRD@6) [16] es una biblioteca que permite la navegación dinámica en aplicaciones React. Para LiftLinked, React Router v6 ha sido fundamental para gestionar la navegación entre las distintas páginas de la aplicación sin recargar la página completa. Esto permite una transición fluida y una mejor experiencia de usuario, especialmente en dispositivos móviles, que son el enfoque principal de la aplicación. Con RRD@6, la navegación es más intuitiva y flexible, utilizando hooks como useNavigate y useParams para manejar rutas dinámicas y parámetros de URL.

5.1.7 *React Icons*

React Icons [17] es una biblioteca que proporciona un conjunto de iconos listos para usar en proyectos React. En LiftLinked, se han utilizado React Icons para mejorar la interfaz de usuario, proporcionando iconos consistentes y estéticamente agradables que son fáciles de integrar y personalizar. Esto ha permitido una representación visual clara y accesible de diferentes funcionalidades y acciones dentro de la aplicación, mejorando la experiencia de usuario y la usabilidad.

5.1.8 *Postman*

Postman es una herramienta de colaboración para el desarrollo de APIs. En LiftLinked, Postman ha sido utilizado principalmente para probar y validar las solicitudes a la API de Wger, así como para asegurarse de que las integraciones con Firebase y otras APIs funcionen correctamente. Postman permite enviar peticiones HTTP y observar las respuestas, lo que facilita el desarrollo y la depuración de las integraciones API antes de implementarlas en la aplicación final.

5.2 *API Wger*

A pesar de la existencia de distintas APIs útiles para este proyecto, finalmente se ha utilizado la API de Wger [18], que proporciona datos sobre ejercicios de musculación, incluyendo descripciones detalladas, categorías musculares, imágenes y equipos necesarios. Se trata de una API muy completa teniendo en cuenta que es gratuita y libre.

Para integrar la API, realizamos peticiones GET desde nuestro frontend en React cuando el usuario accede al selector de ejercicios. Utilizamos hooks como useEffect para gestionar estas llamadas a la API y actualizar el estado de la aplicación con la información recibida. Una vez que los datos se obtienen, se presentan de manera intuitiva y organizada, permitiendo a los usuarios filtrar ejercicios según el grupo muscular o el equipo necesario.

Además, hemos implementado un sistema de caché local para almacenar temporalmente los datos obtenidos de la API, minimizando el número de llamadas a la API y mejorando



así la eficiencia y velocidad de la aplicación. Esta integración asegura que los usuarios tengan acceso rápido y fácil a una variedad de ejercicios, facilitando el proceso de diseño de rutinas y promoviendo un entrenamiento más efectivo y variado.

5.3 Firebase

Como se ha mencionado en el punto 4.1 hemos utilizado Firebase para todo el backend de LiftLinked. Sus funciones principales han sido la autenticación, el servicio de base de datos y el almacenamiento para imágenes.

5.4 Pantallas finales

A continuación, se presentan las pantallas de LiftLinked, siguiendo el flujo de la aplicación y detallando cada elemento que las compone, así como la interacción entre ellas.

LiftLinked se compone de cinco pantallas principales:

- Home
- Diseño de rutinas
- Mis rutinas
- Comunidades
- Perfil

Estas se comunicarán entre sí, siguiendo unas pautas lógicas dentro de la aplicación. Además, para facilitar la accesibilidad y el flujo del usuario la aplicación cuenta con una barra de navegación “Navbar” en la parte inferior con un icono para cada una de las páginas mencionadas.

Previamente, antes de acceder a las pantallas principales, el usuario deberá iniciar sesión y si no está registrado, registrarse. Habrá una pantalla encargada para ambas funciones.

Una vez registrado o loggeado, se redirigirá a Home.

5.4.1 Home

Home es la pantalla principal de la aplicación, contendrá la feed del usuario, es decir, un listado con las publicaciones/posts de los usuarios a los que se sigue. Además, contará con dos botones, uno para buscar usuarios por su “nickname” y otro para acceder a los mensajes del usuario.

Los posts se buscan en la BBDD de Firebase y se muestran ordenados por fecha, en orden descendente (últimos posts primero).

El diseño de cada post se ha realizado mediante un componente *PostItem.js*, el cuál devuelve la imagen del post, con el nombre de usuario del publicador y su foto de perfil, así como un botón para mostrar el entrenamiento que ha realizado.

Al pulsar el icono del perfil se muestra una vista previa del perfil del usuario, que permite seguir (o dejar de seguir) y enviar un mensaje al usuario.

Al pulsar el botón de mostrar entrenamiento se muestra una vista previa del entrenamiento realizado por el usuario publicador del post, y se nos da la opción de copiar la rutina y guardarla en nuestra colección de rutinas.

5.4.2 Buscar usuarios

Esta pantalla permite la búsqueda de usuarios por su Nick. Si se pulsa sobre uno de los perfiles nos redirige a la vista previa del mismo.

5.4.3 Mensajes:

La pantalla de mensajes contendrá un listado con los chats existentes del usuario, así como una barra de búsqueda y un botón para crear chats nuevos.

Para la creación de chats nuevos se reutiliza la pantalla Buscar Usuarios, pero esta vez, al ser llamado desde Mensajes crea el chat con el usuario seleccionado y te redirige al mismo.

5.4.4 Diseño de rutinas

Esta pantalla permitirá el diseño de las plantillas de entrenamiento o rutinas del usuario, y si este es administrador de una comunidad, también de sus suscriptores.

La página se compone de varios elementos:

Una caja de texto para introducir el nombre de la rutina.

Un desplegable para seleccionar la parte del cuerpo a trabajar.

Botones para diseñar la rutina para uno mismo o para un suscriptor (en caso de contar con una comunidad y ser el administrador de la misma)

Un botón para añadir ejercicios:

Este botón abre un modal que habilita la selección de ejercicios.

Para disponer de suficientes ejercicios se ha usado la API de Wger que permite un uso básico gratuito. Trabajamos los ejercicios realizando una petición GET a la API del

nombre y la parte del cuerpo focalizada, esta devolvería un listado de ejercicios ordenados alfabéticamente.

Además, se pueden crear ejercicios propios, mediante un botón dentro de la selección de ejercicios, que se incluyen en el listado de ejercicios. Existe la opción de mostrar únicamente los ejercicios creados por el usuario.

La funcionalidad del selector de ejercicios es bastante simple, se seleccionan (clicando en cada uno) los ejercicios que se quieren añadir y una vez seleccionados todos los deseados, se pulsaría el botón Añadir Seleccionados que hará lo propio.

Una vez añadidos, se creará una tabla por ejercicio, con los siguientes índices: Sets, Reps (Repeticiones), Peso y RIR¹. Se podrán añadir tantas/series en cada ejercicio como el usuario desee, así como eliminar el ejercicio añadido de la rutina.

Por último, el usuario establecerá los valores para los distintos parámetros a modo de objetivo a completar cuando realice el entrenamiento.

5.4.5 *Mis rutinas*

Esta pantalla mostrará el listado de las rutinas del usuario, tanto las creadas por el mismo como las asignadas por un entrenador (admin de una comunidad).

Además, tendrá un botón para crear una rutina nueva que le redirigirá a la página de diseño de rutinas.

Cada rutina del listado contendrá lo siguiente:

- Título de la rutina
- Parte del cuerpo involucrada
- Botón para acceder a la rutina
- Botón para eliminar la rutina

Al acceder a una rutina se mostrará una vista previa de la misma y no será hasta que se pulse el botón “Comenzar rutina” que no comenzará el registro de esta.

Una vez pulsado comenzará un cronómetro que registrará el tiempo durante la sesión.

El usuario verá los valores previos de cada ejercicio, es decir, de la sesión anterior y si no la hubiera, los valores pautados al diseñar la rutina. De forma que podrá seguir un progreso lineal. También podrá modificar estos valores para registrar nuevos datos.

Además, podrá añadir o eliminar series en cada ejercicio.

Esta página también cuenta con un botón para añadir un temporizador, de forma que el usuario pueda medir y controlar los descansos entre series.

Por último, existe un botón para registrar la rutina y en caso de que el usuario haya añadido una imagen o video, se creará un post con la rutina anclada. En caso contrario, simplemente se registrará el entrenamiento y se actualizarán los valores o cambios en la rutina.

5.4.6 Comunidades

Esta pantalla mostrará y/o conducirá a todo el contenido relacionado con la funcionalidad homónima.

La pantalla principal está compuesta por:

- El nombre de la comunidad y, si no se ha seleccionado ninguna, mostrará el texto “Seleccione una comunidad”
- Dos menús desplegables
 - Menú lateral izquierdo
 - Mostrará las comunidades a las que se está suscrito.
 - Un botón para crear comunidades
 - Un botón para buscar comunidades
 - Menú lateral derecho
 - Mostrará un listado con los miembros de la comunidad y permitirá navegar a sus perfiles.
- El chat de la comunidad seleccionada

5.4.7 Perfil

Esta pantalla mostrará el perfil del usuario y contendrá lo siguiente:

- Imagen de perfil
- Nickname
- Email
- Número de seguidores y seguidos
- Listado de posts del usuario
- Listado de logros del usuario
- Seguimientos del progreso del usuario
- Botón que redirigirá a las opciones del perfil
 - Permitirá modificar los datos del usuario, así como su foto de perfil.
 - Permitirá cerrar la sesión del usuario.

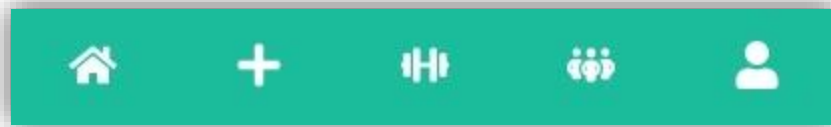


Figura 15. Barra de navegación de la aplicación (Navbar).

Login

Email

Password

Login

[¿No tienes cuenta? Registrate](#)

Figura 16. Pantalla de inicio de sesión.

Register

Nickname

Name

Email

Password

Elegir archivo No se ha selec... ningún archivo

Register

Figura 17. Pantalla de registro.

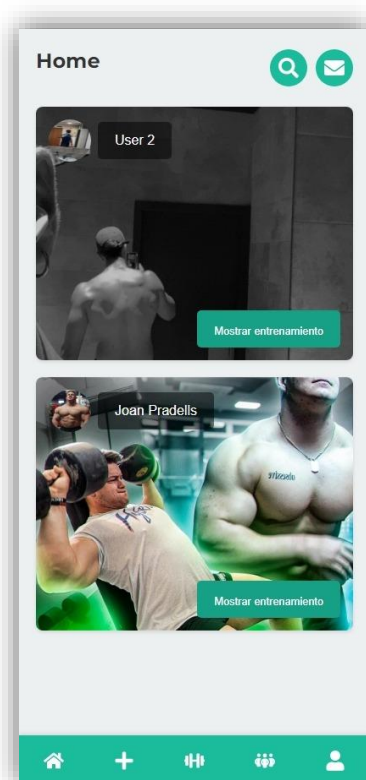


Figura 19. Pantalla Home.

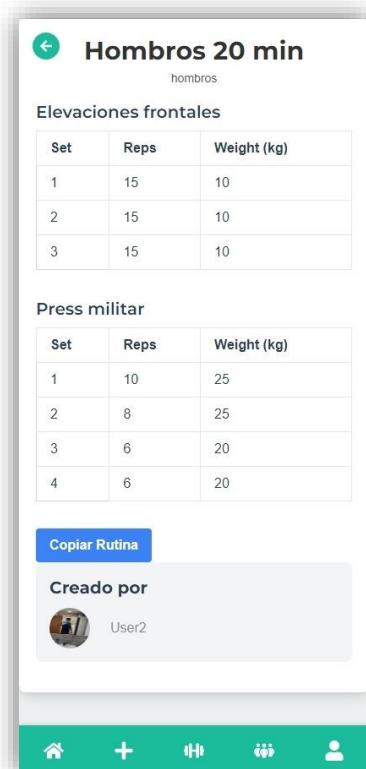


Figura 18. Pantalla de la visualización del entrenamiento enlazado a un post.

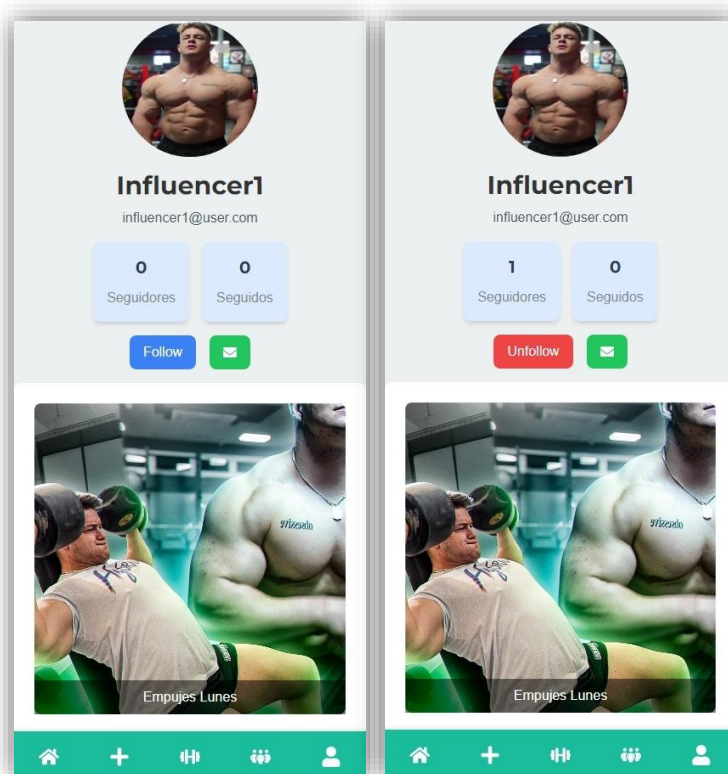


Figura 20. Pantalla de vista previa de perfil.

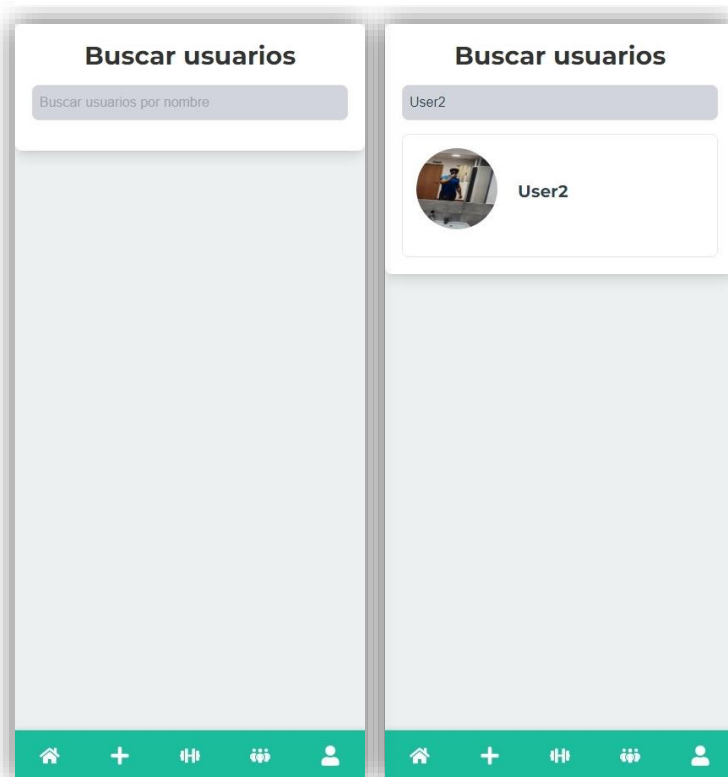


Figura 21. Vista previa de pantalla "Buscar usuarios"

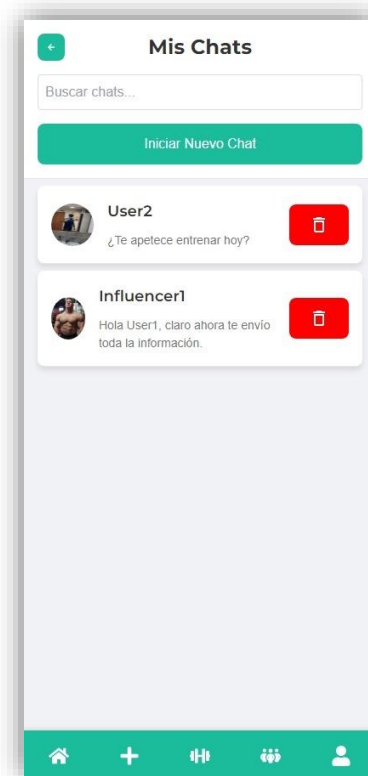


Figura 22. Vista previa de la pantalla "Mis chats"

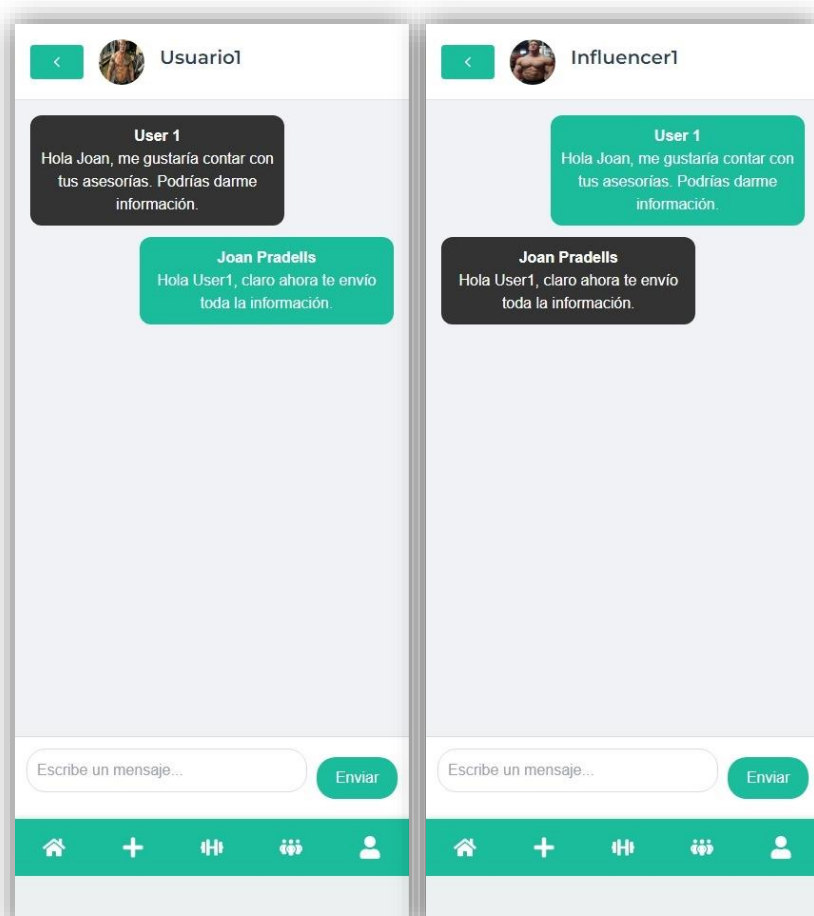


Figura 23. Ejemplo de chat privado entre usuarios.

Diseñador de Rutinas

Nombre de la Rutina:

Parte del Cuerpo:

Asignar Rutina A:

Ejercicios:

Figura 24. Campos del diseñador de rutinas.

Asignar Rutina A:

Seleccionar Suscriptor:

Figura 25. Selección de usuario para la rutina.

Seleccionar Ejercicios

Filtrar por nombre

Piernas

Mostrar solo mis ejercicios

Barbell Hip Thrust

Curl cuádriceps

Curl femoral

Extensión de cuádriceps

Crear Ejercicio Propio Añadir Seleccionados Cerrar

Figura 26. Pop up que muestra el listado de ejercicios.

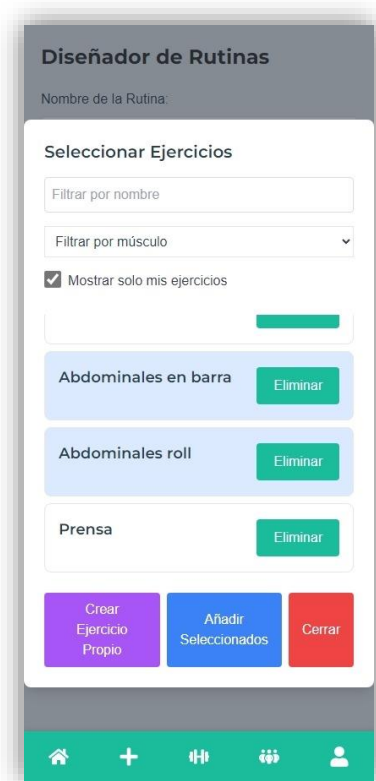


Figura 27. Opción para ver tan sólo los ejercicios propios del usuario.



Figura 28. Pop up para crear un ejercicio propio.



Figura 30. Ejercicios añadidos a la rutina en desarrollo.

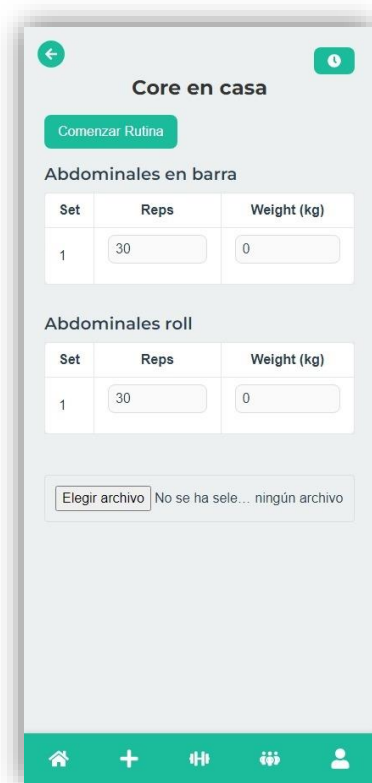


Figura 29. Vista previa de una rutina.

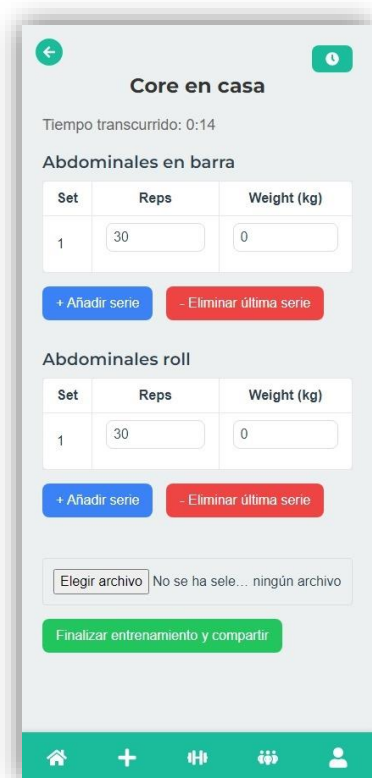


Figura 32. Registro de una rutina.



Figura 31. Pop up para crear un temporizador.

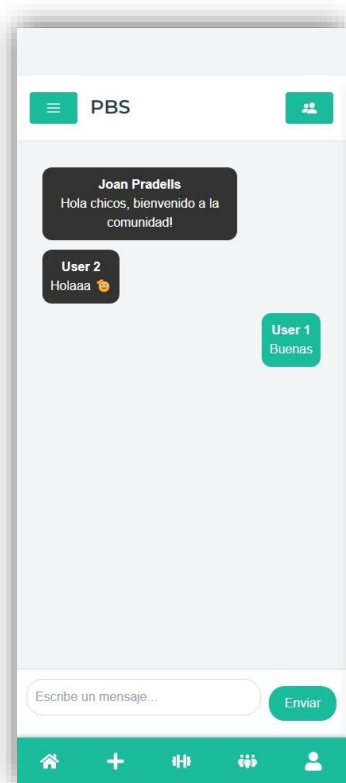


Figura 33. Chat comunitario.

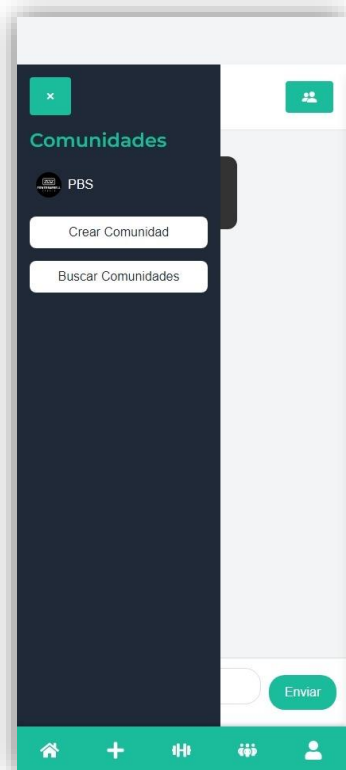


Figura 34. Menú desplegable con el listado de comunidades.

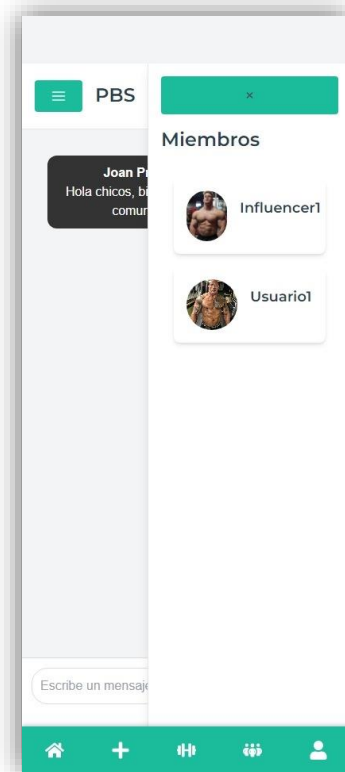


Figura 35. Menú desplegable con el listado de miembros de la comunidad seleccionada.



Figura 36. Pop up para crear una comunidad.



Figura 38. Pop up para buscar comunidades.



Figura 37. Vista previa del perfil de una comunidad.

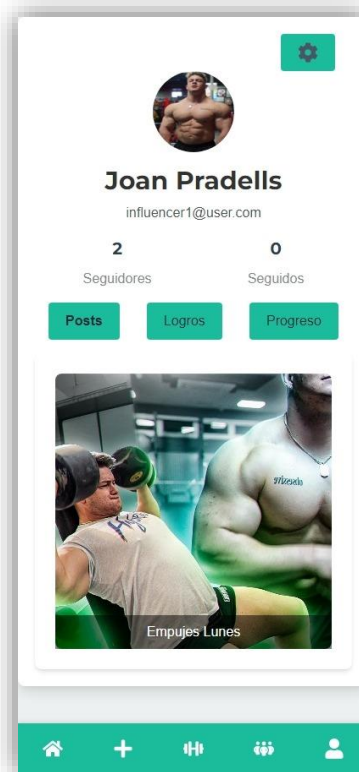


Figura 40. Pantalla "Perfil"

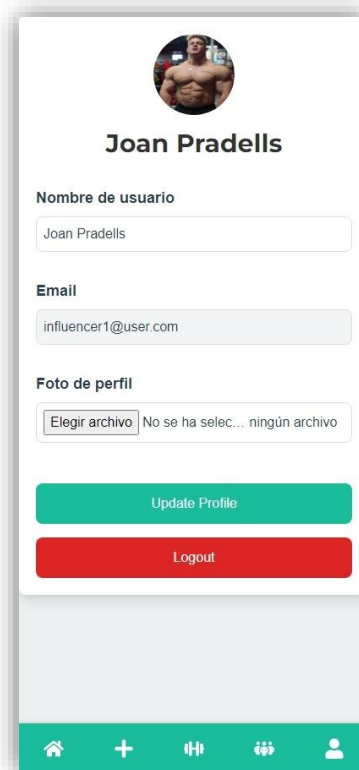


Figura 39. Opciones del perfil.

Capítulo 6. Conclusiones y trabajo futuro

6.1 Conclusiones

Si recordamos el objetivo principal de la aplicación y observamos los resultados obtenidos podemos afirmar que se ha solucionado el problema planteado. Se ha logrado construir una aplicación web, diseñada para dispositivos móviles que combina el registro de entrenamientos con una red social orientada a fanáticos del fitness y la musculación. Su usabilidad es sencilla e intuitiva. A pesar de las posibles diferencias entre las primeras ideas y el resultado final, el producto final cumple los objetivos establecidos y ofrece una solución al problema planteado. Sin embargo, tiene varias líneas de mejora que se expondrán en el siguiente apartado.

Haber completado este trabajo ha supuesto más de lo que imaginaba antes de empezar con él.

Para empezar, he aprendido tecnologías nuevas como React y Firebase, que siempre me habían llamado la atención, pero nunca profundicé en ellas. Me ha ayudado a mejorar mis habilidades en cuanto a la gestión y distribución del tiempo. El trabajo final es realmente el primer proyecto en el que nos encontramos “solos”, sin desmerecer a nuestros tutores, pero en el que somos nosotros los que tenemos que asegurarnos de plantear un problema, buscar y diseñar una solución, plantear y dividir el trabajo y asegurarnos de cumplirlo en los plazos establecidos. Es por ello, que entiendo ahora el sentido de realizar este proyecto al terminar el grado, dándonos la oportunidad para evolucionar las aptitudes obtenidas a lo largo de nuestra vida académica e integrando los conocimientos conseguidos durante el mismo

6.2 Asignaturas relacionadas

En este proyecto convergen varias de las asignaturas cursadas durante el grado.

La más destacable, Tecnologías Web. En ella aprendimos las bases y fundamentos del desarrollo web, así como las tecnologías HTML, CSS y Javascript. Tratándose de una aplicación web, esta asignatura ha estado más que involucrada.

Otra asignatura que destacar es IoT (Internet of Things) que directamente no tiene mucho que ver con el trabajo, pero en ella conocimos y aprendimos a usar la herramienta POSTMAN. Esta ha sido muy útil para el diseño del backend y el uso de la API.

Cabe destacar, BBDD donde aprendimos los conocimientos necesarios para iniciarnos en el mundo de los datos y APLICUSA (Aplicaciones y usabilidad) donde aprendimos a usar Github, el cual ha sido necesario para parte del despliegue de la aplicación.

6.3 Costes del trabajo

A continuación, se detallan los costes estimados del desarrollo de LiftLinked, considerando tanto el tiempo invertido en cada fase del proyecto como los recursos necesarios. El coste por hora se ha establecido en 30€ para reflejar el valor del trabajo técnico especializado en el desarrollo de software.

<i>Fase</i>	<i>Tarea</i>	<i>Horas</i>	<i>Coste</i>
<i>Fase 1: Definición y Planificación</i>	Reunión con el profesor	2	60€
	Análisis de mercado	10	300€
	Estudio de tecnologías necesarias	10	300€
	Planificación previa	6	180€
	Definición de requisitos funcionales y no funcionales	8	240€
	Planificación de arquitectura y estructura	12	360€
	Diseño de bocetos y wireframes iniciales	12	360€
<i>Fase 2: Implementación Básica del Backend</i>	Configuración inicial de Firebase	8	240€
	Implementación de la API Wger y pruebas con Postman	15	450€
	Desarrollo de la lógica para la gestión de usuarios y comunidades	20	600€
<i>Fase 3: Diseño y Desarrollo del Frontend</i>	Desarrollo de la estructura de datos para entrenamientos y rutinas	18	540€
	Configuración inicial de React y CRA	10	300€
	Diseño de las pantallas principales y menú de navegación	20	600€
	Desarrollo de componentes reutilizables	25	750€
	Integración de Firebase en el frontend	15	450€

	Revisión del diseño y pruebas de visualización móvil	12	360€
	Testeo completo y corrección de errores	18	540€
	Despliegue de la aplicación con Firebase Hosting	8	240€
Fase 4: Cierre de Proyecto	Documentación del proyecto	15	450€
	Revisión con el tutor	4	120€
	Entrega final	1	30€
Total		249	7470€

Tabla 20. Cálculo de costes del trabajo.

El mayor coste del proyecto corresponde a las horas invertidas en el desarrollo de la aplicación. El trabajo se ha llevado a cabo con una dedicación promedio de 3 a 4 horas diarias, ajustando el esfuerzo según la carga de trabajo y disponibilidad. Este horario flexible ha permitido avanzar en el proyecto de manera continua, incluso durante fines de semana y días festivos. La tarifa establecida de 30€ por hora es una estimación justa para el nivel de experiencia y el trabajo especializado realizado en el desarrollo de LiftLinked.

6.3.1 Costes materiales

A pesar de que la mayor parte de los costes corresponde al desarrollo de la aplicación también tenemos que destacar algunos costes materiales.

- **Ordenador:** Se ha utilizado un ordenador de alta gama necesario para desarrollar y probar la aplicación de manera eficiente. Un equipo de este tipo es fundamental para ejecutar las herramientas y entornos de desarrollo modernos, como Visual Studio Code, navegadores para pruebas, y emuladores móviles. El coste estimado de un ordenador adecuado para estas tareas ronda los **750€**.
- **Curso de React de Fernando Herrera en Udemy:** Dado que React es el framework principal utilizado para el desarrollo frontend de LiftLinked, se ha invertido en formación específica para asegurar el correcto manejo de la herramienta. El curso de Fernando Herrera en Udemy es uno de los más completos y actualizados disponibles, proporcionando una base sólida y conocimientos avanzados sobre React. El coste del curso es de **15€**.

6.3.2 Costes indirectos

No se incluye ningún gasto indirecto ya que las reuniones y comunicaciones han sido de forma remota.

6.4 Trabajo futuro

Aún cumplidos los objetivos principales, se proponen nuevas funcionalidades o mejoras para la aplicación LiftLinked que, por cuestiones de tiempo, no se han podido implementar.

6.4.1 *Diseño de rutinas con IA: Creación de un asistente que permita crear rutinas mediante IA, tras especificarle los grupos musculares y parámetros esenciales del entrenamiento de fuerza (Intensidad, volumen, ...)*

6.4.2 *BBDD de ejercicios propia: Aunque usamos la API de Wger, lo ideal sería crear una base de datos propia con ejercicios diseñados por nosotros.*

6.4.3 *Programas premium y free: Dado que la aplicación está diseñada para trabajar con influencers y profesionales del sector, sería un gran punto a favor crear distintos planes de pago y/o gratuitos para los consumidores. Habilitando un mayor número de funcionalidades a los usuarios de pago y limitando el uso a los usuarios gratuitos.*

6.4.4 *Edición de rutinas: Actualmente la aplicación no permite la modificación de una rutina existente, se tendría que crear una nueva con los mismos valores. Es por ello por lo que se debería crear esta función*

6.4.5 *Comunidades privadas: En el caso de comunidades pequeñas (Ej/ Gimnasio local) sería recomendable habilitar comunidades privadas a las que solo se pudiera acceder mediante invitación, para asegurar orden y control sobre la comunidad.*

6.4.6 *Aumento del catálogo de logros: Aunque existe la funcionalidad de gamificación, por cuestiones de tiempo apenas se han diseñado logros y metas. Por lo que sería recomendable crear una gran base de logros personalizados para los usuarios. Así como poder diseñar logros personalizados para tu comunidad.*

6.4.7 *Desarrollo nativo: Aunque el diseño web permite una mayor escalabilidad dado que se puede acceder desde “cualquier” dispositivo, un desarrollo nativo para dispositivos móviles favorecería el correcto funcionamiento y diseño de la aplicación. Además, permitiría trabajar sin conexión, lo cual es inviable ahora mismo.*

6.4.8 *Registro de dietas: Además de los entrenamientos, crear una funcionalidad para pautar dietas y comidas abriría aún más el mercado para la aplicación y captaría más clientes del sector y de fuera de éste.*

6.4.9 *Mejora general del diseño*

Bibliografía

- [1] American Council on Exercise, "Fitness Training Program Guidelines," Disponible en: <https://www.acefitness.org/education-and-resources/lifestyle/blog/6658/how-to-create-a-well-balanced-fitness-training-program>, 2021. [Online].
- [2] García, M., & López, S., "El impacto de los influencers en el mundo del fitness y la musculación," *Revista de Cultura Fitness*, vol. 15, no. 3, pp. 45-60, 2021.
- [3] Scrum, "¿Qué es Scrum?," Disponible en: <https://www.scrum.org/resources/blog/que-es-scrum>
- [4] Harbiz App, "Guía oficial de Harbiz para entrenadores y nutricionistas," Disponible en: <https://www.harbizapp.com/guia>, 2019. [Online].
- [5] Strong App, "Manual de usuario de Strong: Registro y seguimiento de entrenamientos," Disponible en: <https://www.strongapp.io/manual>, 2020. [Online].
- [6] Hevy App, "Hevy: Una plataforma para el seguimiento y la comunidad fitness," Disponible en: <https://www.hevyapp.com>, 2021. [Online].
- [7] Mozilla Developer Network, "JavaScript Guide," Disponible en: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>, 2022. [Online].
- [8] FaceReact, "React: A JavaScript library for building user interfaces," Disponible en: <https://reactjs.org/>, 2021. [Online].
- [9] Google LLC, "Firebase Documentation," Disponible en: <https://firebase.google.com/docs>, 2021. [Online].
- [10] Wger Workout Manager, "Wger API Documentation," Disponible en: <https://wger.de/en/software/api>, 2021. [Online].
- [11] Mozilla Developer Network, "CSS: Cascading Style Sheets," Disponible en: <https://developer.mozilla.org/en-US/docs/Web/CSS>, 2022. [Online].
- [12] Tailwind Labs, "Tailwind CSS Documentation," Disponible en: <https://tailwindcss.com/docs>, 2021. [Online].
- [13] Microsoft Corporation, "Visual Studio Code Documentation," Disponible en: <https://code.visualstudio.com/docs>, 2022. [Online].
- [14] GitHub, Inc., "GitHub Documentation," Disponible en: <https://docs.github.com/en>, 2022. [Online].
- [15] Facebook Inc., "Create React App: Set up a modern web app by running one command," Disponible en: <https://create-react-app.dev/>, 2021. [Online].
- [16] React Training, "React Router v6 Documentation," Disponible en: <https://reactrouter.com/docs/en/v6>, 2021. [Online].
- [17] React Icons, "React Icons Documentation," Disponible en: <https://react-icons.github.io/react-icons/>, 2022. [Online].



[18] Postman, Inc., "Postman Documentation," Disponible en:
<https://learning.postman.com/docs/>, 2022. [Online].



ANEXOS

Anexo I. Enlace a la aplicación LiftLinked

[React App \(tfg-gym-c4291.firebaseio.com\)](https://tfg-gym-c4291.firebaseio.com)