



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Una aplicación para planificación y gestión de  
entrenamientos de gimnasio. Desarrollo del Frontend

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Fierrez Ruipérez, Jesús

Tutor/a: Letelier Torres, Patricio Orlando

CURSO ACADÉMICO: 2023/2024

*A mis padres y hermana, cuyo esfuerzo han sido  
la base de mi camino.*



# Agradecimientos

---

Este TFG no habría sido posible en primer lugar sin la ayuda de mi compañero Juan Pablo Bueno fuentes, que ha sido el encargado de realizar la parte *backend* de la aplicación.

Por otra parte, me gustaría hacer hincapié en nuestras compañeras del grado de BBAA de la UPV, gracias a Andrea Palomo, Joanna López y Noemí Ortega, por hacer del diseño de esta aplicación.

Por último y no menos importante, a Patricio el tutor de este TFG, por ser el guía de todo este proceso.



# Resumen

---

En un contexto donde el *fitness* y el bienestar personal están ganando cada vez más importancia, *Social Lift* surge como una aplicación innovadora que busca revolucionar la forma en que las personas planifican y gestionan sus rutinas de entrenamiento. Basada en la tendencia hacia métodos de aprendizaje más activos y dinámicos, *Social Lift* ofrece una plataforma interactiva que permite a los usuarios crear, compartir y seguir rutinas de ejercicios personalizadas.

El objetivo de este TFG es desarrollar y validar la idea de negocio asociada a *Social Lift*, extendiendo sus funcionalidades para crear un producto atractivo y funcional. El proyecto se enmarca en *Start.inf*, el espacio de emprendimiento de la ETSINF, y seguirá el método *Lean Startup*. Esto incluye la evaluación de la idea de negocio, la puesta en marcha del proyecto y su desarrollo a través de al menos dos MVP (Minimum Viable Products) con experimentos con *early adopters*. El producto se desarrollará como una aplicación móvil multiplataforma.

El proyecto de emprendimiento ha sido realizado por dos estudiantes: Juan Pablo Bueno Fuentes, encargado del backend, y yo, encargado del frontend. Además, el resto de tareas asociadas al proyecto se han realizado conjuntamente.

**Palabras clave:** *Desarrollo de software, Lean Startup, emprendimiento.*

# Abstract

---

In a context where fitness and personal well-being are gaining more and more importance, *Social Lift* emerges as an innovative application that seeks to revolutionize the way people plan and manage their training routines. Based on the trend towards more active and dynamic learning methods, *Social Lift* offers an interactive platform that allows users to create, share and follow personalized exercise routines.

The objective of this TFG is to develop and validate the business idea associated with *Social Lift*, extending its functionalities to create an attractive and functional product. The project is part of *Start.inf*, the entrepreneurship space of the ETSINF, and will follow the *Lean Startup* method. This includes the evaluation of the business idea, the implementation of the project and its development through at least two MVP with experiments with early adopters. The product will be developed as a cross-platform mobile app.

The entrepreneurship project has been carried out by two students: Juan Pablo Bueno, in charge of the backend, and me, in charge of the frontend. In addition, the rest of the tasks associated with the project have been carried out jointly.

**Keywords:** *Software development, Lean Startup, entrepreneurship.*

# Tabla de contenidos

---

## Contenido

1.	Introducción .....	11
1.1	Motivación .....	11
1.2	Objetivos .....	13
1.3	Estructura de la memoria .....	14
2.	Evaluación de la Idea de Negocio .....	15
2.1	Resumen de <i>Social Lift</i> .....	15
2.2	Clientes .....	16
2.3	Estudio de mercado y competidores .....	17
2.3.1	<i>Efort App</i> .....	17
2.3.2	<i>Strava</i> .....	18
2.3.3	<i>Hevy</i> .....	19
2.3.4	Gymshark training and fitness .....	20
2.4	Tabla comparativa de competidores .....	21
2.5	Proyección de ingresos y gastos .....	23
2.6	Análisis DAFO .....	26
2.7	<i>Lean Canvas</i> .....	27
2.8	Conclusiones de la evaluación .....	28
3.	Tecnologías utilizadas .....	29
3.1	<i>Flutter</i> .....	30
3.2	Visual Studio Code .....	30
3.3	<i>Figma</i> .....	31
4.	Desarrollo de la solución .....	32
4.1	Metodología .....	32
4.1.1	Sprints .....	33
4.1.2	Herramientas de apoyo para la planificación y el seguimiento .....	35
4.2	Requisitos .....	36
4.3	Diseño .....	41
4.3.1	Diseño de la IGU .....	42
4.3.2	Arquitectura <i>Frontend</i> .....	43
4.3.3	Diagrama de clases .....	44
4.3.4	Diagrama de <i>widgets</i> de <i>Flutter</i> .....	45

4.4	Programación.....	46
4.4.1	Aplicación de patrones .....	46
4.4.2	<i>Refactorizaciones</i> .....	49
4.4.3	Desafíos de programación .....	51
4.5	Pruebas .....	54
4.5.1	<i>Widget Testing</i> .....	54
4.5.2	Pruebas de Aceptación .....	56
5.	Cronología del TFG .....	58
5.1	Desarrollo del primer MVP.....	60
5.1.1	Primer Experimento .....	62
5.2	Desarrollo del segundo MVP .....	67
5.2.1	Segundo Experimento .....	69
5.2.2	Dedicación .....	74
6.	Conclusiones y trabajo futuro .....	75
6.1	Conclusiones .....	75
6.2	Trabajo Futuro.....	76
7.	Referencias .....	77
	ANEXOS.....	78
	ANEXO A: GUIA DE USO .....	78
1.	Feed de Rutinas .....	78
2.	Calendario de Rutinas.....	80
3.	Perfil de Usuario .....	83
4.	Ranking.....	85
	ANEXO B: OBJETIVOS DE DESARROLLO SOSTENIBLE.....	86



# Índice de figuras

Figura 1. Esquema de una rutina de gimnasio .....	12
Figura 2. Logo de Social Lift.....	15
Figura 3. Ventanas Effort App.....	17
Figura 4. Ventanas Strava .....	18
Figura 5. Ventanas Hevy.....	19
Figura 6. Ventanas Gymshark .....	20
Figura 7. Gráfico Resultado trimestral acumulado .....	25
Figura 8. Diseño de la Pantalla Calendario .....	31
Figura 9. Mock-up Pantalla Calendario .....	31
Figura 10. Workflow tareas Sprints de desarrollo.....	33
Figura 11. Tarea Calendario Rutinas .....	33
Figura 12. Plantilla Prueba de aceptación Tarea Calendario de rutinas .....	34
Figura 13. Tablero Kanban .....	35
Figura 14. Diagrama CU – Características Social Lift .....	36
Figura 15. De izquierda a derecha. Mock-up y Diseño pantalla Nueva Rutina.....	42
Figura 16. Arquitectura Frontend Social Lift .....	43
Figura 17. Diagrama de Clases Social Lift .....	44
Figura 18. Diagrama de Componentes Pantalla Calendario .....	45
Figura 19. Diagrama Patrón Fachada VideoService.....	46
Figura 20. Código Patrón Fachada VideoService.....	47
Figura 21. Patrón Provider .....	48
Figura 22. Diagrama Patrón Singleton.....	48
Figura 23. Patrón Singleton .....	48
Figura 24. Clase Serie .....	49
Figura 25. Clase Serie tras la refactorización .....	50
Figura 26. Clase SerieAdapter. Nueva implementación de los mappings .....	50
Figura 27. Pantalla Calendario .....	52
Figura 28. Primer requisito: Requisito de los markers.....	53
Figura 29. CardRutina.....	55
Figura 30. Test CardRutina .....	55
Figura 31. Plan de Pruebas de Aceptación Calendario Rutinas .....	56
Figura 32. Escala de Prioridad .....	58
Figura 33. Línea temporal del proyecto .....	59
Figura 34. Estado Tareas del Sprint 1 al comienzo .....	60
Figura 35. Tablero Kanban Sprint 1 .....	60
Figura 36. Tareas Sprint 2 .....	61
Figura 37. Gráfico ¿Con qué frecuencia realizas entrenamientos en el gimnasio?.....	62
Figura 38. Gráfico ¿Con qué frecuencia realizas entrenamientos en el gimnasio? .....	63
Figura 39. Gráfico ¿Dónde apuntas tu progreso diario en los diferentes ejercicios de tus rutinas?.....	64
Figura 40. Gráfico ¿Conoces alguna aplicación útil para gestionar tus entrenamientos de gimnasio?.....	64
Figura 41. Gráfico ¿Cuál es tu fuente de información para añadir nuevos ejercicios en tus rutinas?.....	65
Figura 42. Gráfico ¿Qué impresión tienes de las pantallas de la aplicación? .....	65

Figura 43. Gráfico ¿Consideras que la información presentada en las pantallas es clara y concisa? .....	66
Figura 44. Gráfico ¿Cómo calificarías tu nivel de satisfacción general con el estilo de las pantallas de la aplicación? .....	66
Figura 45. Sprint 3.....	67
Figura 46. Sprint 4.....	68
Figura 47. Validación del requisito Feed de rutinas.....	70
Figura 48. Validación requisito Verificación de los usuarios.....	71
Figura 49. ¿Qué tan intuitiva encuentras la interfaz de usuario de la aplicación? .....	71
Figura 50. ¿Te resulta fácil encontrar las funciones que necesitas dentro de la aplicación? .....	72
Figura 51. Navegar por las rutinas.....	78
Figura 52. Buscador.....	79
Figura 53. Pantalla calendario.....	80
Figura 54. Rutinas .....	80
Figura 55. Nueva rutina.....	81
Figura 56. Nueva rutina y añadir ejercicios .....	82
Figura 57. Rutina Detalles .....	82
Figura 58. Perfil de Usuario- Datos Morfológicos/Datos Ejercicios.....	83
Figura 59. Perfil Usuario - Plantillas .....	84
Figura 60. Ranking.....	85

# Índice de tablas

---

Tabla 1. Comparativa de las funcionalidades de los competidores .....	22
Tabla 2. Ingresos anuales .....	23
Tabla 3. Gastos Anuales.....	24
Tabla 4. Resultados Anuales.....	24
Tabla 5. Matriz DAFO.....	26
Tabla 6. Lean Canvas.....	27
Tabla 7. Tabla Ejemplo Conversiones .....	51
Tabla 8. Tabla dedicación horas (Desglose) del proyecto .....	74
Tabla 9. Tabla dedicación horas (Genérico) del proyecto .....	74

# 1. Introducción

---

## 1.1 Motivación

El auge del interés por el deporte y el *fitness* ha provocado un crecimiento exponencial en la cantidad de personas que frecuentan gimnasios en todo el mundo. Según la Asociación Internacional de Salud, Raqueta y Clubes Deportivos (IHRSA, por sus siglas en inglés) <sup>1</sup>, más de 230 millones de personas a nivel mundial son miembros de gimnasios y esta cifra sigue incrementándose a un ritmo impresionante [1]. Este aumento no es solo una tendencia pasajera; es un reflejo de un cambio global hacia un estilo de vida más saludable y activo.

Un aspecto particularmente interesante es el crecimiento del interés en el *fitness* entre los jóvenes. La Federación Internacional de Culturismo y Fitness (IFBB) <sup>2</sup> informa que el interés en el *fitness* ha crecido en un 22% anual entre aquellos menores de 30 años [2]. Esta tendencia indica que las nuevas generaciones están cada vez más comprometidas con su bienestar físico reconociendo la importancia de mantenerse en forma desde una edad temprana. Esto no solo mejora su salud física, sino que también contribuye a su bienestar mental y emocional.

La evolución tecnológica también ha jugado un papel crucial en el mundo del *fitness*. Hoy en día muchos usuarios no solo siguen un plan de entrenamiento, sino que también buscan herramientas avanzadas para monitorear su progreso.

Muchos son los usuarios que utilizan técnicas rudimentarias como libretas, en las que apuntan las diferentes marcas de los ejercicios que realizan en ese entrenamiento, u otras no tan rudimentarias como las "Notas" de su teléfono. Bajo mi propia experiencia, como usuario habitual de gimnasios, sufro de esta técnica, yo mismo apunto mis marcas en las "Notas" de mi teléfono. Estos apuntes están basados en los diferentes ejercicios que realizas durante tu rutina de entrenamiento, representando así la serie (en cada ejercicio se realiza un número de series concretas), en cada serie se utiliza un peso y se realizan unas determinadas repeticiones con ese peso como se ilustra en la Figura 1.

---

<sup>1</sup> <https://www.healthandfitness.org/>

<sup>2</sup> <https://ifbb.com/>

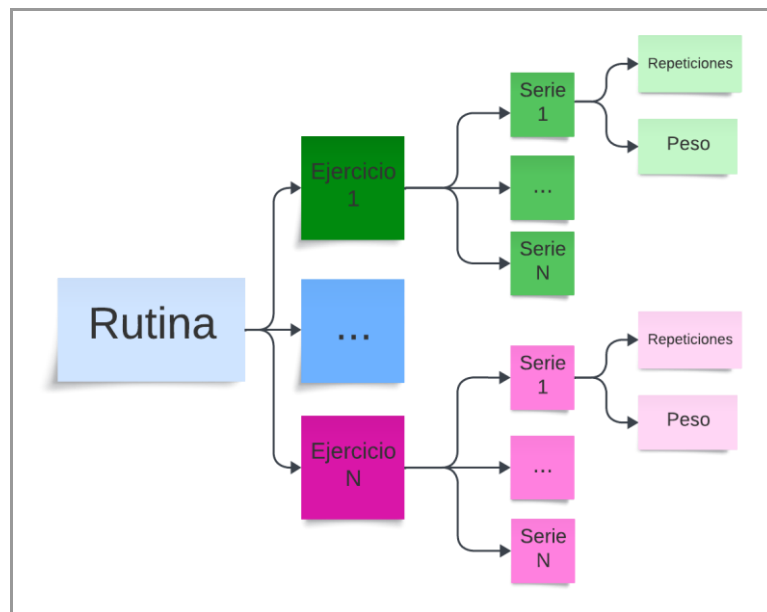


Figura 1. Esquema de una rutina de gimnasio

Con el creciente número de personas que adoptan un estilo de vida fitness la demanda de soluciones más sofisticadas y tecnológicas se hace cada vez más evidente. Aplicaciones avanzadas y dispositivos de seguimiento pueden ofrecer beneficios significativos como análisis de progreso recomendaciones personalizadas basadas en los datos recopilados acceso a información actualizada y una comunidad que se apoya mutuamente y comparte logros lo cual es esencial para mantener la motivación a largo plazo.

El *fitness* digital no solo se trata de registrar ejercicios se trata de crear una experiencia integral que fomente el crecimiento personal y el bienestar holístico. Las innovaciones tecnológicas están transformando cómo entrenamos, proporcionándonos herramientas que hacen que nuestros esfuerzos sean más efectivos y eficientes.

Existen diferentes aplicaciones o plataformas que nos ayudan a plasmar nuestro progreso, evitando así el uso de las técnicas anteriormente mencionadas, como pueden ser *Efort App*<sup>3</sup>, *Hevy*<sup>4</sup>, *Gymshark training and fitness*<sup>5</sup>, etc. Estas herramientas no solo ayudan a los usuarios de gimnasios a plasmar su progreso, si no que aportan funcionalidades como rutinas de entrenamiento personalizadas o diferentes estadísticas de progreso.

Estas herramientas son de gran utilidad para los usuarios, pero el grado de dificultad de uso puede llegar a ser alto en algunos casos y no están enfocadas socialmente como aplicaciones para establecer vínculos entre los diferentes usuarios que las utilizan.

De esta problemática nace la idea de *Social Lift*, una aplicación móvil multiplataforma que aporta las funcionalidades de sus competidores, implementado interfaces de usuario sencillas y un uso bastante simple, además de ofrecer una interacción con diferentes usuarios y crear relaciones entre ellos para que la comunidad del *fitness* crezca. En su desarrollo se empleará la metodología *Lean Startup* para asegurar que el proyecto avanza en la dirección más apropiada en cada fase, y además técnicas de otras metodologías ágiles que faciliten su gestión.

<sup>3</sup> <https://efortcoach.com/>

<sup>4</sup> <https://www.hevyapp.com/>

<sup>5</sup> <https://central.gymshark.com/article/the-gymshark-training-app-for-ios-and-android>

## 1.2 Objetivos

El propósito de este proyecto de emprendimiento es desarrollar un producto que facilite, apoye y motive a la comunidad *fitness* global en sus entrenamientos y planificación, a través de una aplicación móvil llamada *Social Lift*. Para lograr este objetivo y en el marco de este TFG se han establecido metas específicas:

- Evaluar la viabilidad financiera del proyecto.
- Crear una aplicación con una usabilidad sencilla y una interfaz llamativa.
- Crear ciertas funcionalidades que monitoricen el progreso de los usuarios en las rutinas de ejercicios.
- Enfocarlo a una red social, creando un *feed* de rutinas para que los usuarios puedan compartir sus rutinas y logros con los demás usuarios que siguen y así poder crear una estrecha comunidad *fitness*.
- Realizar una gestión clara y ágil usando aplicaciones de gestión de proyectos.
- Validar el producto con un entorno real, realizando dos experimentos con *early adopters*.

Además de los objetivos relacionados con el proyecto, también hay metas personales vinculadas a este TFG:

- Mejorar y profundizar en la aplicación práctica de técnicas, metodologías y herramientas aprendidas durante la carrera.
- Explorar y dominar nuevos lenguajes de programación asociados al desarrollo de este producto.
- Adquirir experiencia en el desarrollo completo de un producto de *software*, desde su concepción hasta su implementación.
- Profundizar en el proceso de establecimiento de una empresa centrada en el desarrollo de *software* y tecnología.

### 1.3 Estructura de la memoria

Este trabajo expone la generación de una idea de negocio junto con el posterior desarrollo *frontend* de una aplicación móvil que se materializa esta idea en un producto software.

La exposición de todo este proceso se ha distribuido de la siguiente forma:

- **Capítulo 1. Introducción:** Sintetiza toda la información relativa a la motivación y los objetivos del proyecto, así como, la estructura de la memoria.
- **Capítulo 2. Evaluación de la idea de negocio:** Muestra la documentación que ha ayudado a definir la idea de negocio incluyendo las técnicas empleadas para desarrollarla, análisis del mercado en el que se espera implantar y estudio económico que muestra su viabilidad.
- **Capítulo 3. Tecnologías utilizadas:** Define todo lo relativo a las tecnologías utilizadas tanto para el desarrollo del proyecto, como para su diseño.
- **Capítulo 4. Desarrollo de la solución:** En este capítulo final se detalla todo lo relacionado con el desarrollo de la solución, abarcando aspectos como los procesos de gestión, la especificación de requisitos, el desarrollo de la aplicación y las pruebas.
- **Capítulo 5. Cronología del proyecto:** Explica la cronología del proyecto, desarrollado mediante dos MVP en 2 Sprints de desarrollo cada uno. Al final de cada MVP, se realizan experimentos con *early adopters* para validar las funcionalidades y obtener retroalimentación.
- **Anexo:** Incluye una guía de usuario de la aplicación y la relación del proyecto con los ODS (Objetivos de Desarrollos Sostenible) <sup>6</sup>.

---

<sup>6</sup> <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

## 2. Evaluación de la Idea de Negocio

---

En este capítulo se ilustra toda la documentación que ha ayudado a definir la idea de negocio incluyendo todas las técnicas empleadas para desarrollarla, análisis de mercado, estudio de la viabilidad económica, etc.

### 2.1 Resumen de *Social Lift*

*Social Lift*, cuyo logo se muestra en la Figura 2, es una innovadora aplicación móvil multiplataforma desarrollada para facilitar y ayudar a la comunidad *fitness* a registrar sus entrenamientos de manera fácil, sencilla y efectiva, proporcionando gran variedad de ejercicios y rutinas para adaptarse a las necesidades de cada usuario, ya sean principiantes o expertos. Cuenta con un *feed* (foro) integrado para la integración con diferentes usuarios de la aplicación, los cuales podrán compartir sus rutinas de entrenamiento al igual que guardar y acceder a rutinas compartidas por otros miembros de la comunidad de *Social Lift*, promoviendo la colaboración e intercambio de conocimientos en el entorno del *fitness* construyendo una red de información sólida.



Figura 2. Logo de *Social Lift*

Con *Social Lift*, podrás crear o añadir rutinas personalizadas propias o de usuarios que sigues en la aplicación. Además, tendrás acceso a un *feed* donde podrás ver las rutinas de los usuarios que sigues y usarlas en tus entrenamientos. También es posible subir videos de las series que realizas en cada rutina, los cuales podrán ser analizados por usuarios verificados, quienes te darán retroalimentación sobre la correcta ejecución de los ejercicios.

Podrás especificar el gimnasio donde realizas tus rutinas y planificar tu semana de entrenamiento. La aplicación proporciona diferentes estadísticas personalizadas basadas en tu progreso según tus registros de peso, progreso, metas y objetivos, permitiéndote crear un seguimiento efectivo y detallado de tu rendimiento.

En resumen, *Social Lift* es una aplicación que promueve y facilita la organización de tus rutinas en gimnasio con el fin de simplificar, mejorar tus entrenamientos y crear la mejor comunidad fitness del mundo.



## 2.2 Clientes

*Social Lift* tiene un mercado objetivo diverso, tanto en términos de quienes pagarían por el producto como en quienes lo usarían. Aquí hay una descripción de ambos:

Clientes potenciales (quienes pagarían por el producto):

- **Usuarios Individuales *Premium*:** Aquellos que buscan funcionalidades avanzadas y están dispuestos a pagar por una suscripción premium que desbloquee características adicionales, como análisis detallados de progreso, acceso ilimitado a rutinas y ejercicios premium, y participación en competiciones exclusivas.
- **Gimnasios y Centros Deportivos:** Establecimientos que deseen mejorar la experiencia de sus miembros ofreciendo una plataforma digital integrada para seguimiento de entrenamientos, gestión de rutinas y competiciones internas. Estos establecimientos podrían pagar por una suscripción empresarial y, a su vez, podrían incluir el costo de la aplicación en las tarifas de membresía.
- **Entrenadores Personales y Profesionales del Fitness:** Aquellos que deseen ampliar su alcance y ofrecer servicios de asesoramiento personalizado a través de la plataforma. Pueden optar por una suscripción especial que les permita utilizar la función de coach y acceder a herramientas adicionales para la gestión de clientes.
- **Marcas de *Fitness* y Empresas Relacionadas:** Empresas que buscan aprovechar la plataforma como un canal de marketing para promocionar sus productos y servicios relacionados con el fitness. Pueden pagar por publicidad en la aplicación o por colaboraciones estratégicas para promover sus productos entre los usuarios.

Usuarios potenciales (quienes usarían el producto):

- **Entusiastas del *Fitness* y Culturistas:** Personas apasionadas por el fitness que buscan mejorar su rendimiento, seguir rutinas personalizadas y conectarse con otros miembros de la comunidad para compartir conocimientos y experiencias.
- **Principiantes en el *Fitness*:** Individuos que están dando sus primeros pasos en el mundo del *fitness* y necesitan orientación, motivación y seguimiento para establecer y alcanzar sus objetivos de acondicionamiento físico de manera segura y efectiva.
- **Atletas y Deportistas:** Personas que buscan optimizar su entrenamiento y mejorar su rendimiento en su disciplina deportiva específica mediante el seguimiento de su progreso, la participación en competiciones y la colaboración con entrenadores y compañeros de equipo.
- **Usuarios Sociales:** Aquellos que disfrutan de la interacción social y la competencia amistosa, y que encuentran valor en la participación en desafíos y competiciones dentro de la comunidad de la aplicación. Esto puede incluir personas que no son necesariamente atletas, pero que están interesadas en mantenerse activas y saludables de una manera divertida y motivadora.

## 2.3 Estudio de mercado y competidores

El estudio de mercado es fundamental para comprender el entorno competitivo en el que el proyecto se encuentra. Para ello, se han estudiado productos similares que se encuentren actualmente en el mercado que puedan ser competidores, en los cuales se han definido las funcionalidades y servicios básicos que ofrecen cada uno de estos para ponerlo en comparación *Social Lift*.

Con esta información se pueden extraer las funcionalidades básicas que tendría que tener la aplicación en este sector, al igual que aspectos útiles que diferenciaría a *Social Lift* para hacerla más atractiva a los usuarios.

Las aplicaciones a comparar son las siguientes:

### 2.3.1 Efort App



Efort App <sup>7</sup> es una aplicación móvil simple diseñada para ayudar a registrar entrenamientos físicos y hacer seguimiento. Dispone de registro de peso y medidas corporales, creación de rutinas y estadísticas.

Se pueden registrar entrenamientos donde se añaden ejercicios y los organiza según la semana. Esta rutina de entrenamiento se puede asociar a un día del calendario.

Se puede registrar periódicamente el peso y ejercicios de trac. Cada uno de estos registros se utiliza para sacar un gráfico de estadísticas a medida del tiempo para ver el progreso, como ilustra la Figura 3.

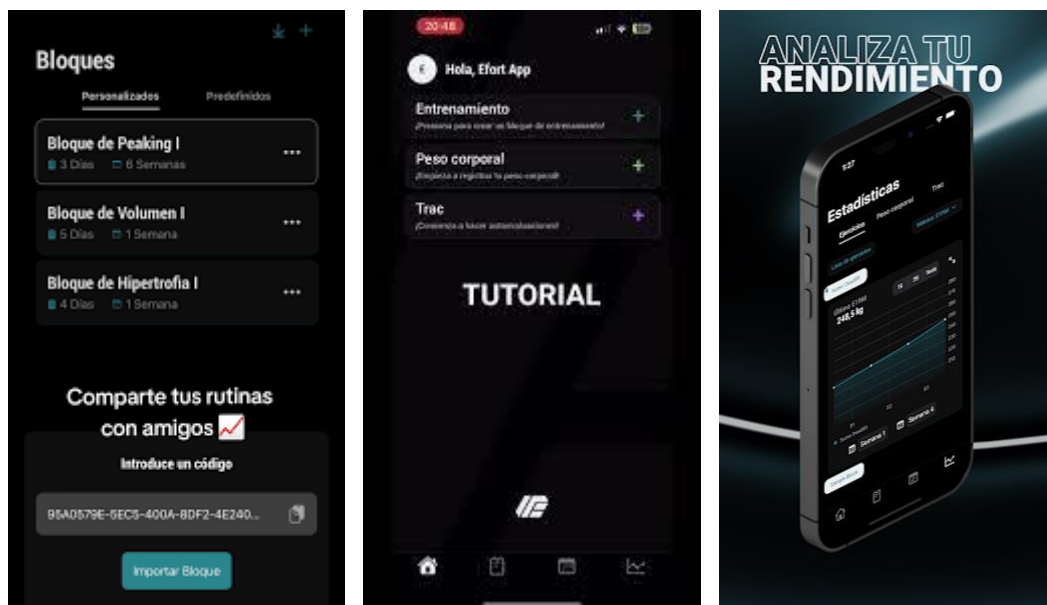


Figura 3. Ventanas Efort App

<sup>7</sup> <https://efortcoach.com/>

### 2.3.2 Strava



*Strava*<sup>8</sup> es una red social basada en Internet y GPS enfocada a deportistas como pueden ser ciclistas y corredores y una aplicación de seguimiento GPS deportiva. La red la gestiona una compañía del mismo nombre con sede en San Francisco [3].

Los usuarios de *Strava* pueden registrar sus actividades físicas utilizando dispositivos GPS como relojes deportivos o teléfonos inteligentes, y luego cargar los datos a la plataforma para analizar su rendimiento, compartir experiencias con la comunidad de *Strava* y recibir retroalimentación sobre su entrenamiento, todo ello ilustrado en la Figura 4. Además, *Strava* ofrece características sociales que permiten a los usuarios seguir a otros atletas, dar "me gusta" y comentar en actividades de otros, así como unirse a clubes y eventos locales o globales.

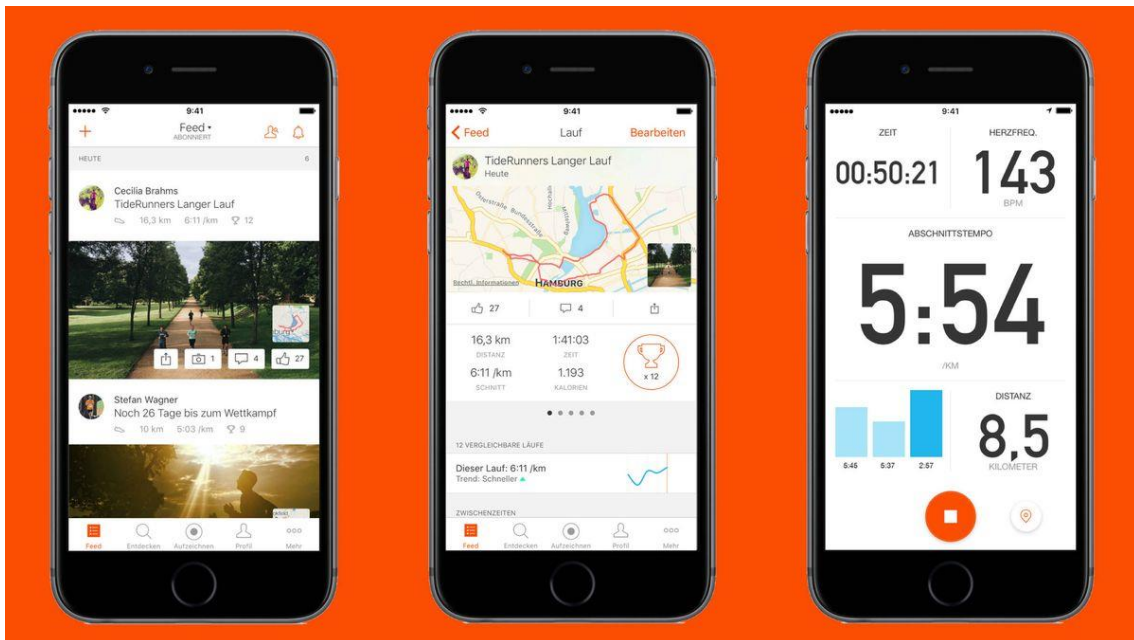


Figura 4. Ventanas Strava

<sup>8</sup> <https://www.strava.com/?hl=es-ES>

### 2.3.3 Hevy



Hevy<sup>9</sup> es una aplicación móvil centrada en el levantamiento de pesas en la que es posible registrar rutinas de entrenamiento y hacer seguimiento de estos. Ofrece una biblioteca de ejercicios y técnicas de entrenamiento [4].

En la aplicación puedes registrar rutinas y agregar ejercicios predeterminados proporcionados por la aplicación y también asociar el tiempo que se tarda en hacer los ejercicios.

La aplicación proporciona una red social para compartir rutinas y publicaciones de distintos usuarios y ver sus estadísticas de ejercicios, como se puede observar en la Figura 5.

Se pueden registrar medidas corporales que se desbloquean con una versión de pago, y ofrece un panel de configuración para seleccionar el idioma, conexión con *Google fit*<sup>10</sup> y con *Strava*, ocultar las opciones de red social y cambiar información sobre el perfil de usuario.

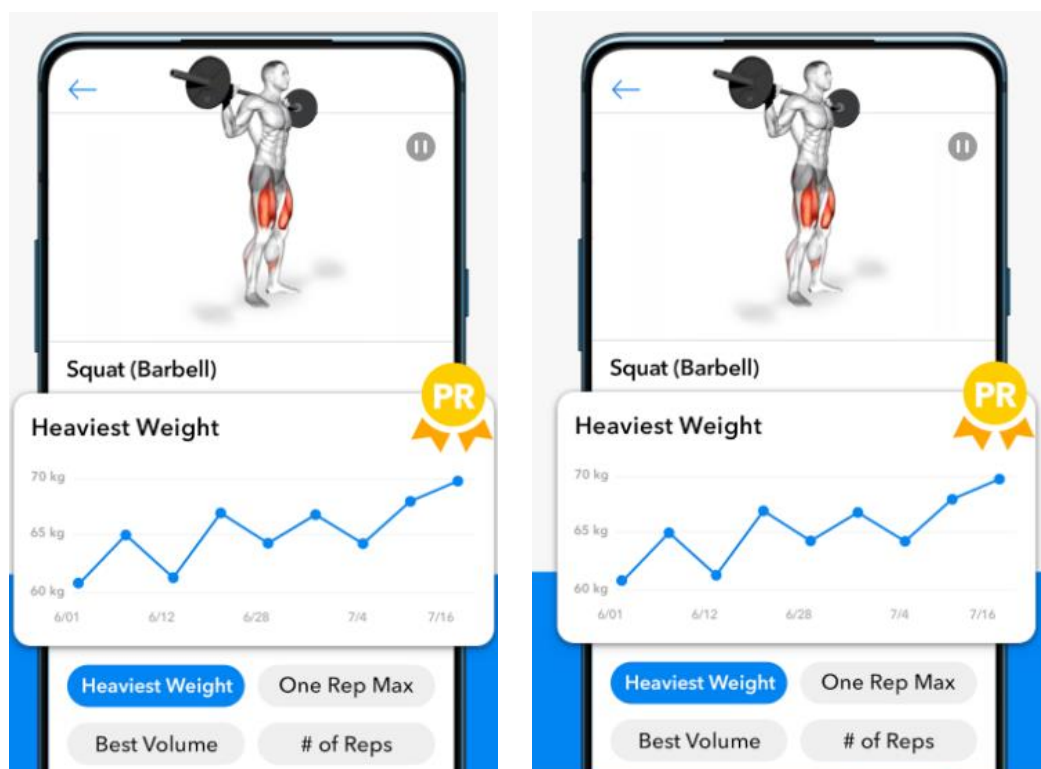


Figura 5. Ventanas Hevy

<sup>9</sup> hevy.com

<sup>10</sup> <https://www.google.com/fit/>

### 2.3.4 Gymshark training and fitness<sup>11</sup>



Aplicación móvil que ofrece una variedad de programas de entrenamiento diseñados por entrenadores y deportistas profesionales, así como seguimiento de progreso.

Ofreciendo rutinas variadas según el tipo de entrenamiento que se desea realizar y diferentes planes de entrenamiento.

Dando la opción de personalizarlos a tu gusto y mostrando diferentes estadísticas sobre tu progreso, como ilustra la Figura 6.

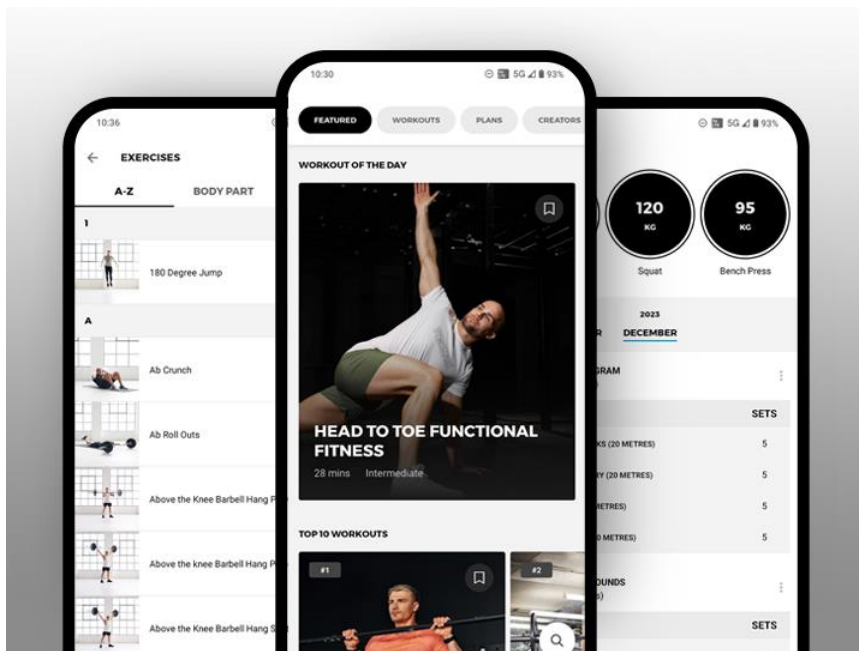


Figura 6. Ventanas Gymshark

<sup>11</sup>

[https://play.google.com/store/apps/details/Gymshark\\_Training\\_Fitness\\_App?id=com.gymshark.fitness&hl=es](https://play.google.com/store/apps/details/Gymshark_Training_Fitness_App?id=com.gymshark.fitness&hl=es)

## 2.4 Tabla comparativa de competidores

Tras el análisis se ha desarrollado una tabla (Tabla 1) en la que se compara las diferentes funcionalidades claves que ofrecen cada una de las aplicaciones competidoras y muestra si la aplicación dispone de esta, si lo realizan a medias o si por el contrario no ofrecen ningún servicio parecido.

	<b>Effort App</b>	<b>Strava</b>	<b>Hevy</b>	<b>Gymshark Training and fitness</b>	<b>SocialLift</b>
Registro de actividades deportivas	Sí	Sí	Sí	Sí	Sí
Calendario Interactivo	Sí	No	No	No	Si
Estadísticas de progreso de ejercicios	Sí	Sí	Sí	Sí	Sí
Estadísticas personales	No	No	Sí	No	Sí
Proporcionar rutinas	No	Sí	Sí	Sí	Sí
Seguimiento de usuarios	No	Sí	Sí	No	Sí
Registro de ejercicios	Sí	Sí	Sí	Sí	Sí
Registro de rutinas/entrenamiento	Sí	Sí	Sí	Sí	Sí
Proporción de Rutinas predeterminadas	No	Sí	Sí	Sí	No
Foro Usuarios	No	Sí	No	Sí	Sí
Multiplataforma	Sí	Sí	Sí	Sí	Sí
Anuncios de actividades	No	No	Sí	No	No
Vinculación entre Gimnasios/Lugares	No	Sí	No	No	Sí
Rankings	No	Sí	No	No	Sí
Publicación audiovisual	No	Sí	Sí	No	Sí
Integración con inteligencia artificial	No	No	No	No	En un futuro
Personalización de entrenamiento	Sí	Sí	Sí	Sí	Sí
Coaching online	No	No	No	No	Sí
Competiciones <i>online</i>	No	Sí	No	No	Sí

Suscripciones Premium	No	Sí	Sí	No	Sí
Suscripciones Gratuitas	Sí	Sí	Sí	Sí	Sí

Tabla 1. Comparativa de las funcionalidades de los competidores

*Social Lift* se propone cubrir todas las funcionalidades básicas que ofrecen los diferentes competidores, además de introducir herramientas innovadoras como el coaching online y las competiciones y rankings para motivar a los usuarios. También se planea la futura integración de inteligencia artificial, que podría incluir la recomendación de rutinas o un *chatbot* que ofrezca consejos y advertencias sobre los ejercicios.

A diferencia de *Strava* o *Hevy*, que ofrecen rutinas predeterminadas, *Social Lift* busca crear una comunidad sólida donde los usuarios aporten su experiencia y compartan rutinas directamente entre ellos.

Además, *Social Lift* pretende funcionar como una red social, creando una comunidad amplia en el mundo del fitness mediante un feed en el que los usuarios pueden interactuar.

Para fomentar la interacción, las rutinas podrán vincularse con los gimnasios donde se realicen. Por último, pero no menos importante, un aspecto clave de nuestra aplicación es el Calendario Interactivo, que permitirá a los usuarios planificar y gestionar sus rutinas. Esta característica es fundamental, ya que, según la Tabla 1, solo uno de nuestros competidores la ofrece actualmente.

## 2.5 Proyección de ingresos y gastos

Esta proyección económica es una estimación simple y básica de ingresos y gastos en los 3 primeros años del ciclo de vida del producto. El estudio económico se ha realizado teniendo en cuenta los ingresos y gastos trimestrales durante estos 3 años.

Las diferentes fuentes de ingreso del producto son las siguientes:

- **Suscripciones de *Coaching Online*:** como su nombre indica el usuario paga por un servicio de *coaching* (entrenamiento) *online* a través de la aplicación. Este servicio está basado en un pago recurrente del usuario cada trimestre del año. El precio es de 7€ por trimestre.
- **Publicidad de marcas de la comunidad *fitness*:** es una fuente de ingreso de la que dispone el producto para que las marcas relacionadas con la comunidad *fitness*, publiciten sus productos a cambio de una cuota trimestral recurrente de 1500€.

Para estas dos fuentes de ingresos se han estimado los usuarios y las marcas, respectivamente, que adquieran estos servicios en la aplicación según el aumento de usuarios que van al gimnasio y las marcas *fitness* que existen ahora mismo, como se puede ver en la Tabla 2.

Es de vital importancia destacar que el análisis se ha hecho trimestralmente pero que en lo que concierne a la memoria solo se muestra un resumen con los datos relativos al último trimestre de cada año.

Ingresos	Año 1 / T4	Año 2 / T4	Año 3 / T4
Suscripcion de Coaching Online	3485	10025	50231
Publicidad de marcas de la comunidad fitness	2	4	5
Ingresos Anuales en T4			
Suscripcion Coaching x 7 €	24,395 €	70,175 €	351,617 €
Publicidad de marcas x 1500€	3,000 €	6,000 €	7,500 €
<b>Total Ingresos</b>	<b>27,395 €</b>	<b>76,175 €</b>	<b>359,117 €</b>

Tabla 2. Ingresos anuales

Con respecto a los gastos, durante el primer año, mostrados en la Tabla 3, se han tenido en cuenta los relacionados con la Infraestructura *Cloud* (infraestructura en la nube), ordenadores, los gastos relacionados con los sueldos del CEO, CTO y los desarrolladores de mantenimiento, un director de marketing, personal administrativo, campañas de marketing, internet, electricidad, alquiler y la gestoría.

Durante el segundo año, se han mantenido la mayoría de los gastos, menos la inversión en ordenadores, pero ha aumentado el coste de la gestoría y de lo relativo a electricidad, internet y alquiler.

Durante el tercer año, se han realizado un aumento de la inversión en las campañas de marketing, aumentos del sueldo a diferentes empleados, contratación de un desarrollador junior y por ende el aumento de gastos relacionados con el internet, electricidad, alquiler y gestoría.





El resumen de esta proyección se ilustra en la Figura 7 relatando de forma más visual lo expuesto anteriormente. Con esta gráfica podemos observar que el Punto de Equilibrio está en el primer trimestre del segundo año, donde los ingresos cubren los gastos de años anteriores y se empieza a obtener liquidez económica.

Además, se puede observar que, a partir del año 2, los ingresos aumentan exponencialmente.

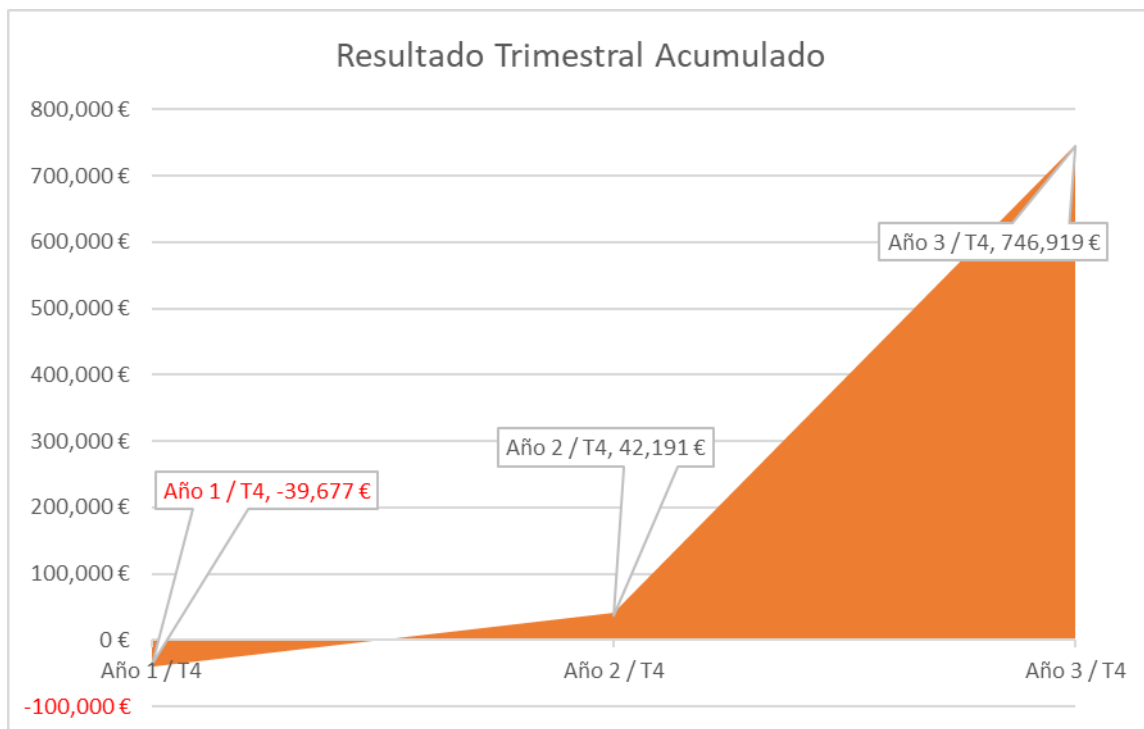


Figura 7. Gráfico Resultado trimestral acumulado

## 2.6 Análisis DAFO

El Análisis DAFO es una herramienta de estudio de la situación de una empresa, institución, proyecto o persona, analizando sus características internas (Debilidades y Fortalezas) y su situación externa (Amenazas y Oportunidades) en una matriz cuadrada. Es una herramienta para conocer la situación real en que se encuentra una organización, empresa o proyecto, y planear una estrategia a futuro [5].

El objetivo principal de la realización de la matriz DAFO, ilustrada en la Tabla 5, es representar de manera más clara cuáles son los puntos a favor y en contra de la idea de negocio y con ello saber dónde orientar futuras estrategias.

<b>DEBILIDADES</b> <ul style="list-style-type: none"><li>• Aplicación desconocida, esto implica un gran coste humano en entrometerse de lleno en el nicho de mercado.</li><li>• Poca experiencia en gestión empresarial.</li></ul>	<b>AMENAZAS</b> <ul style="list-style-type: none"><li>• Gran competencia de aplicaciones similares.</li><li>• Aplicaciones similares asentadas en el mundo del <i>fitness</i>.</li></ul>
<b>FORTALEZAS</b> <ul style="list-style-type: none"><li>• Novedades tecnológicas como <i>machine learning</i> o <i>inteligencia artificial</i>.</li><li>• No implica gran inversión económica.</li><li>• Buen conocimiento del público objetivo.</li></ul>	<b>OPORTUNIDADES</b> <ul style="list-style-type: none"><li>• Nicho de mercado en pleno auge.</li><li>• Interés de la aplicación a gente joven.</li><li>• Enfoque de la aplicación como una red social que pueda crear una amplia comunidad en el mundo del <i>fitness</i>.</li></ul>

Tabla 5. Matriz DAFO

Como podemos observar en la Tabla 5, las principales debilidades presentadas por nuestra idea de negocio son las relacionadas con la gestión empresarial, dada la poca experiencia que tenemos, y el alto coste que implica darse a conocer en su nicho de mercado, ya que al principio seremos una competidor desconocido. Por el contrario, destacamos diversas fortalezas, estas están relacionadas con las novedades tecnológicas, la poca inversión económica y el conocimiento del público objetivo.

Durante la evaluación de la idea de negocio también hemos encontrado amenazas debido a la gran competencia con productos *software* similares y ya asentados en el mundo del *fitness*. Sin embargo, existen diferentes oportunidades, como que será un producto *software* que se implantará en un nicho de mercado que hoy en día está en pleno auge y creciendo de manera exponencial y el enfoque de la aplicación como una red social que pueda crear una gran comunidad en el mundo del *fitness*.

## 2.7 Lean Canvas

La idea de negocio en un proyecto de emprendimiento software se refiere a la concepción original y fundamental que inspira la creación de un nuevo producto software. Esta idea es la base del desarrollo del proyecto y su principal función es resolver un problema o satisfacer las necesidades específicas de un nicho de mercado. Algunos elementos claves que afectan al desarrollo de la idea son: el problema a resolver, los clientes, la viabilidad económica, la infraestructura y la innovación entre muchos otros.

Para desarrollar la idea de negocio hemos empleado la técnica *Lean Canvas* que es una herramienta visual utilizada para describir y analizar un modelo de negocio de manera rápida y clara para el cliente. Un *Lean Canvas* se divide en 9 secciones: clientes, problema, proposición de valor, solución, canales, ingresos, costos, métricas y ventaja competitiva.

El resultado de la utilización de esta técnica se muestra en la Tabla 6 siendo esta la versión actual de este documento, aunque podría ser sometida a cambios en un futuro a medida que se desarrolle el producto.

Destacar la sección número 9 del *Lean Canvas* “Ventaja Competitiva”, en esta sección se explican las diferentes características de la idea de negocio que nos diferencia de la competencia. Cabe mencionar que estas características dejarán de ser una ventaja competitiva cuando la competencia las implemente en su producto.

<b>7 Costos</b> Los costes principalmente son de carácter personal, tanto desarrolladores como directores de proyectos o personal administrativo. Por otro lado, hay gastos en publicidad y marketing para así captar más usuarios. Por último, gastos en gestión, servicios e infraestructuras <i>cloud</i> y material personal como ordenadores		<b>6 Ingresos</b> Modelo de suscripciones premium a los clientes para acceder a funcionalidades premium como coach de personal experto de gimnasios directo publicidad de marcas relacionadas con el mundo del fitness dentro de la aplicación		
<b>2 Problema</b> Los usuarios que realizan actividades para ejercitarse carecen de una forma de registro de rutinas eficiente que les ayude a hacer seguimiento.	<b>4 Solución</b> Una aplicación móvil que permite a los usuarios registrar eficientemente rutinas de ejercicios para hacer seguimiento constante, incorporando también una red social que permite a los usuarios compartir sus logros y rutinas para ayudar a gente inexperta	<b>3 Proposición De valor</b> La aplicación proporcionará un sistema de registro y planificación de entrenamientos, así como una comunicación constante entre diferentes usuarios que quieran compartir sus rutinas y el soporte de diferentes Coach que ayuden a los usuarios a mejorar y progresar.	<b>9 Ventaja competitiva</b> Un foro entre los usuarios en el que los mismos podrán compartir sus rutinas y así poder ayudar a otros usuarios a la hora de realizar sus ejercicios, así como diferentes formas de registrar y visualizar tu progreso.  La realización de diferentes competiciones y desafíos virtuales donde los usuarios compiten entre ellos en diferentes categorías ofreciendo premios virtuales para motivar las participaciones.  Mentorías en la que usuarios experimentados y validados por la aplicación (Coach) asesoran a otros usuarios inexpertos y podrán recibir una bonificación por ello	<b>1 Clientes</b> Nuestros clientes serían distintos gimnasios interesados en saber y planificar las rutinas de sus clientes.  Nuestros early adopters son gimnasios cercanos a nosotros.  Cabe hacer una distinción entre usuarios y clientes.  Nuestros clientes serán los gimnasios, pero hay que tener en cuenta a los usuarios de los gimnasios, que no son clientes, pero sí usuarios directos de la aplicación.
	<b>8 Métricas</b> Número de Usuarios registrados, frecuencia de uso de la aplicación, monitoreo de las iteraciones entre usuarios, nivel de participación en la comunidad, porcentaje de centros deportivos sincronizados en la aplicación respecto al total			

Tabla 6. *Lean Canvas*

## 2.8 Conclusiones de la evaluación

*Social Lift* como idea de negocio no solo ofrece innovación en el mercado creciente de la comunidad *fitness* y las aplicaciones destinadas a la gestión de rutinas de entrenamiento, gracias a los estudios realizados se han podido definir las características concretas que se van a implementar y que marca una propuesta de valor y un camino a seguir muy claro, conciso y con respecto a la competencia.

Con el *Lean Canvas* y el estudio de nuestros competidores, hemos podido detectar que el mercado actual, este tipo de aplicaciones no ofrece enfoques relacionados con la interacción de usuarios (enfoque de red social), además, también, se ha detectado que nuestros competidores ofrecen una usabilidad que a veces no es sencilla ni intuitiva, esto se convierte en una característica importante a implementar para poder distinguirse de ellos.

Por otro lado, en el Análisis DAFO hemos podido comprobar que las debilidades y las amenazas detectadas no son igual de importantes que las oportunidades y las fortalezas que hemos detectado en nuestro proyecto.

Por último, con el estudio de la proyección económica, hemos podido comprobar que la inversión inicial mínima que requiere este proyecto es de 32903€ y que la recuperación de la misma se alcanzaría en el primer trimestre del segundo año. Además, podemos concluir que el beneficio acumulado crece exponencialmente a partir del año 2 de vida del producto.

En definitiva, nos encontramos con un proyecto con gran capacidad de generar ingresos y con una gran solvencia económica, así como una gran magnitud de usuarios consumidores de la aplicación, todo esto junto con las ventajas comentadas anteriormente hace que *Social Lift* sea un proyecto viable.

### 3. Tecnologías utilizadas

---

En este capítulo, se describirán y explicarán de manera detallada las diversas tecnologías empleadas durante el desarrollo del producto, abarcando tanto las tecnologías relacionadas con el desarrollo del Frontend como aquellas vinculadas con el diseño de la aplicación en su totalidad. Se ofrecerá una visión completa del proceso de selección y utilización de estas tecnologías, explicando el razonamiento detrás de cada decisión tomada.

Inicialmente, tras definir el objetivo principal del proyecto y decidir que se desarrollaría una aplicación móvil multiplataforma, se exploraron múltiples ideas sobre las tecnologías, entornos y lenguajes de programación Frontend adecuados para la creación de *Social Lift*. Este proceso de exploración fue fundamental para asegurar que se seleccionara la opción más eficiente y adecuada para el proyecto.

Entre las opciones consideradas, una de las propuestas más destacadas fue el uso de *React Native*, un *framework* (marco de trabajo) muy popular que inicialmente se consideró una excelente opción debido a la experiencia previa con este en proyectos anteriores. La familiaridad con *React Native* presentaba la ventaja de eliminar la necesidad de un período de aprendizaje extenso, lo cual podría haber acelerado el desarrollo inicial. Sin embargo, a pesar de estas ventajas, la experiencia existente con *React Native* también llevó a su descarte. El equipo de desarrollo tenía el objetivo de aprender y utilizar un nuevo *framework* de desarrollo más moderno y prometedor, lo cual motivó la búsqueda de alternativas.

Después de investigar y analizar otros *frameworks* disponibles en el mercado, se tomó la decisión de optar por *Flutter*, un *framework* de código abierto desarrollado por Google. *Flutter* se destacó como una opción ideal para el desarrollo de aplicaciones móviles multiplataforma debido a sus numerosas ventajas y características innovadoras. Además, una influencia significativa en la elección de *Flutter* fue su creciente popularidad y presencia en redes sociales, donde numerosos desarrolladores compartían sus aplicaciones y experiencias positivas con este *framework*. La existencia de una gran y activa comunidad de usuarios que respaldan y contribuyen al desarrollo de *Flutter* fue un factor decisivo que aportó seguridad y confianza en la elección de esta tecnología para el proyecto.

De esta manera, se concluye que la decisión de utilizar *Flutter* no solo fue motivada por sus capacidades técnicas y su potencial innovador, sino también por la sólida comunidad de soporte y recursos disponibles, que garantizaban un entorno de desarrollo robusto y con amplias posibilidades de aprendizaje y crecimiento.

### 3.1 Flutter



*Flutter*<sup>12</sup> es un *framework* de código abierto desarrollado por Google que se utiliza para crear aplicaciones móviles, web y de escritorio de alta calidad con una sola base de código. Utiliza el lenguaje de programación *Dart*, que fue también creado por Google, y ofrece un conjunto completo de herramientas y bibliotecas para el desarrollo de aplicaciones multiplataforma [6].

Su rendimiento rápido se debe a su motor de renderizado *Skia*, que compila el código a código nativo, asegurando una experiencia de usuario fluida en todas las plataformas. La UI declarativa y personalizable, junto con la función de *Hot Reload*, que es una característica que permite a los desarrolladores ver los cambios en el código de su aplicación casi instantáneamente, sin tener que reiniciar la aplicación por completo, facilita la creación e iteración rápida de interfaces de usuario complejas.

Además, *Flutter* cuenta con una amplia comunidad y respaldo activo de Google, lo que garantiza un soporte sólido, actualizaciones regulares y un crecimiento continuo. La elección de *Flutter* se justifica por su eficiencia en el desarrollo, su rendimiento óptimo y su capacidad para crear interfaces de usuario altamente personalizadas y atractivas, todo ello respaldado por una comunidad activa y un sólido soporte de Google.

### 3.2 Visual Studio Code



En todo proceso de desarrollo de una aplicación, se necesita un IDE (Entorno de desarrollo integrado) para poder llevarlo a cabo.

Elegir *Visual Studio Code*<sup>13</sup> para desarrollar *Social Lift* es una excelente opción por varias razones:

En primer lugar, *Visual Studio Code* es un editor de código altamente versátil y potente, con soporte integrado para una amplia gama de lenguajes de programación, incluido *Dart*, el lenguaje utilizado en *Flutter*, el *framework* de desarrollo elegido para *Social Lift*. Además, *Visual Studio Code* ofrece una interfaz de usuario intuitiva y personalizable, junto con una amplia variedad de extensiones y complementos que mejoran la productividad del desarrollador, como el soporte para depuración, control de versiones, etc.

También, debido a la excelente experiencia con *Visual Studio Code* en proyectos anteriores nos hizo decantarnos para utilizarlo como IDE.

---

<sup>12</sup> <https://flutter.dev/>

<sup>13</sup> <https://code.visualstudio.com/>

### 3.3 Figma

En todo desarrollo Frontend, es esencial emplear alguna tecnología para el diseño del producto.

En nuestro proyecto, el diseño del Frontend de la aplicación fue realizado con la colaboración de estudiantes del Grado de Bellas Artes de la Universidad Politécnica de Valencia (UPV). Dado que habíamos trabajado juntos en proyectos similares anteriormente, decidimos realizar el diseño del Frontend de esta aplicación de manera conjunta.

Por todo ello y a elección de ellas, debido a su experiencia en el diseño de interfaces graficas de usuarios, decidieron que se usara *Figma* para el diseño.

*Figma*<sup>14</sup> es una herramienta de diseño de interfaz de usuario basada en la web que ofrece colaboración en tiempo real para equipos de diseño y desarrollo. Permite a los usuarios crear, colaborar y prototipar diseños de manera eficiente en un entorno único y accesible desde cualquier dispositivo con conexión a Internet. Sus beneficios incluyen la capacidad de trabajar simultáneamente en un proyecto, compartir fácilmente diseños con otros usuarios, realizar comentarios y revisiones en tiempo real, y mantener una sola fuente de verdad para todos los activos de diseño, lo que facilita la colaboración y la iteración eficaz durante el proceso de diseño de productos digitales.

Cabe destacar que en la especificación de requisitos se realizaban mockups (Figura 9) que luego se ponían en común con nuestras compañeras de BBAA para que ellas fueran mediante *Figma* las que le dieran un mejor y más óptimo diseño (Figura 8).

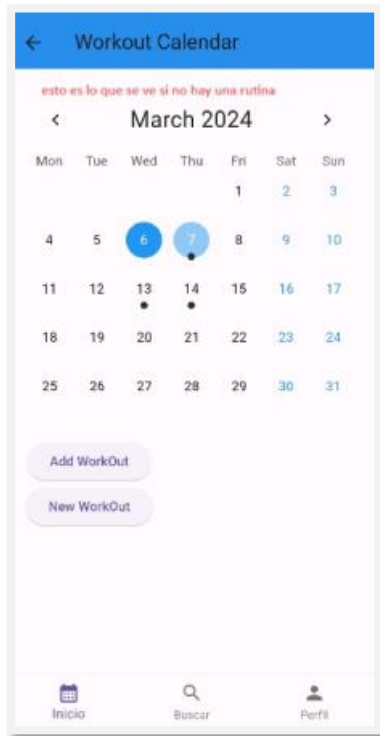


Figura 9. Mock-up Pantalla Calendario



Figura 8. Diseño de la Pantalla Calendario

<sup>14</sup> <https://www.figma.com>



## 4. Desarrollo de la solución

---

### 4.1 Metodología

Hemos elegido adoptar la metodología *Lean Startup* [7] para llevar a cabo nuestra idea propuesta. En este enfoque, formulamos hipótesis sobre el negocio y las validamos a medida que avanzamos en el desarrollo del producto.

Establecemos objetivos claros que nos guían hacia la creación de versiones iniciales del producto, conocidas como *Minimum Viable Product* (MVP) [8], que incluyen las funciones esenciales para probar estas hipótesis. Estas versiones iniciales son sometidas a pruebas con un grupo de usuarios *Early adopters* (clientes potenciales), cuya retroalimentación es fundamental para validar nuestras hipótesis y orientar el desarrollo del producto en la dirección correcta.

Este proceso se considera un ciclo, donde repetimos estas iteraciones en cada etapa de evolución del producto, ajustándose continuamente a las necesidades y observaciones obtenidas durante su desarrollo.

También, se ha optado por incorporar distintas técnicas de otras metodologías ágiles, siguiendo el manifiesto ágil.

Hemos optado por incorporar prácticas de la metodología *Scrum* [9] que se basa en principios de transparencia, inspección y adaptación. Las prácticas incorporadas son las siguientes:

- **Rol de *Product Owner*:** Representa los intereses del cliente y del negocio, define los requisitos del producto y prioriza el *backlog* del producto. En este caso y debido a que somos usuarios habituales de gimnasios nosotros éramos nuestros propios *POs*.
- **Rol de *Scrum Master*:** Facilita el proceso *Scrum*, ayuda al equipo a eliminar obstáculos y garantiza que se sigan los principios y prácticas de *Scrum*. En este caso era un papel que llevábamos a cabo nosotros, también.
- ***Sprints*:** en lo que concierne a este proyecto hemos definido 5 *Sprints*, un *Sprint* inicial, en el que se evalúa todo lo respectivo a la idea de negocio y 4 *Sprints* de desarrollo de la aplicación, estos *Sprints* de desarrollo tienen una duración de 3 semanas cada uno.
- **Reuniones de retrospectiva del *Sprint*:** Reunión al final del *Sprint* donde el equipo reflexiona sobre su desempeño y busca formas de mejorar en el próximo *Sprint*.
- ***Backlog*:** Una lista priorizada de todos los elementos de trabajo que podrían ser necesarios en el producto final.

Esta combinación de *Lean Startup* y *Scrum* nos permite adaptarnos rápidamente a medida que obtenemos retroalimentación del mercado y evolucionamos nuestra idea de negocio en cada iteración.

### 4.1.1 Sprints

Durante un *Sprint*, un equipo de desarrollo se compromete a completar un conjunto definido de tareas, también conocidas como historias de usuario, en un período de tiempo fijo, que generalmente varía entre una y cuatro semanas.

En el caso de este proyecto se han definido 5 *Sprints* para su desarrollo. El primer Sprint o Sprint 0 se llevó a cabo para el desarrollo de la idea de negocio y la formación en las tecnologías.

Los cuatro siguientes *Sprints* están enfocados al desarrollo del producto y las tareas en cada uno de los siguen el siguiente *workflow* (flujo de trabajo):

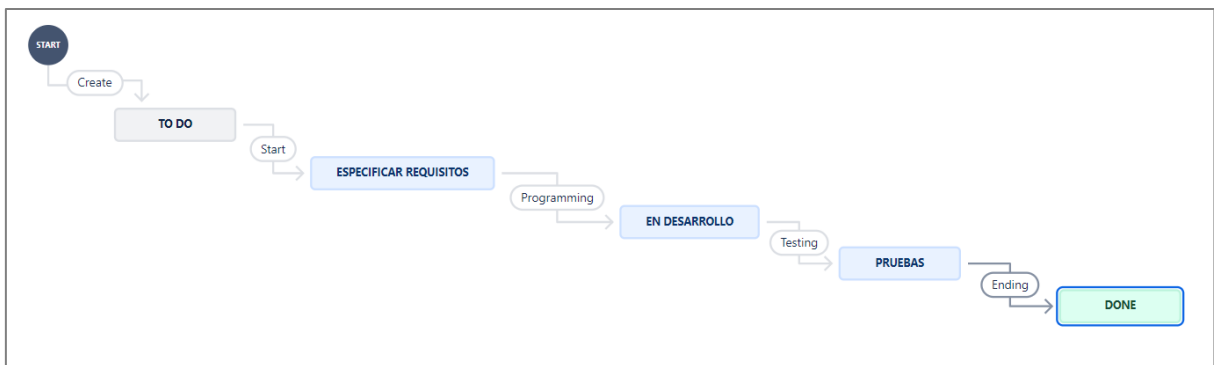


Figura 10. Workflow tareas Sprints de desarrollo

- **To Do:**
  - o En este estado es donde se añaden las Tareas o UTs (Unidades de trabajo) del tablero.
- **Especificar Requisitos:**
  - o En este estado se llevaba a cabo la especificación de requisitos, esta especificación se realiza mediante una breve explicación o un *mock-up* sencillo y que explique la funcionalidad (Figura 11).

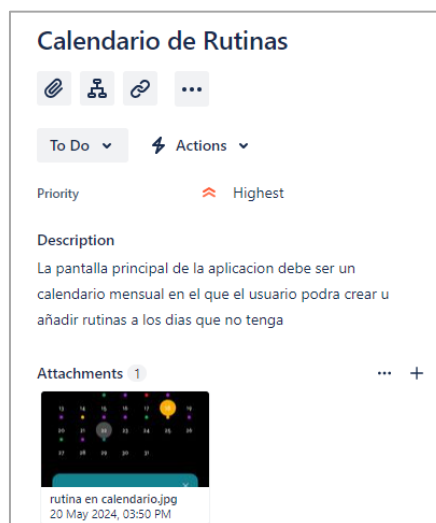


Figura 11. Tarea Calendario Rutinas

- **En Desarrollo:**
  - o Tras definir los requisitos de esa tarea, pasa a el estado de En Desarrollo y en este paso se llevan a cabo todos los procesos relativos a la programación de la misma.
- **Pruebas:**
  - o Estado siguiente a En Desarrollo, la tarea pasa al estado de Pruebas en la que, tras la especificación de un plan de pruebas previo, se lleva a cabo el plan de pruebas de la tarea, como el mostrado en la Figura 12. Si el plan de pruebas ha sido exitoso la tarea se pasa al estado de Done, en caso contrario, se vuelve al estado de En Desarrollo y se avisa a su responsable para que la revise.

Calendario de rutinas			
Jesús Fierrez	Se mostrará un calendario, en el que en el día correspondiente, si el usuario tiene una rutina asignada, debe aparecer en ese día		
22-03-24			
PASOS	PRUEBAS REALIZADAS	RESULTADO	COMENTARIOS
1	Se debe poder visualizar un calendario mensual	OK	
2	Se podrá navegar entre los días y los meses	OK	
3	Si y solo si el usuario tiene rutina asociada a ese día, tras pulsar un día se debe mostrar la rutina asociada	OK	
4	Si y solo si el usuario no tiene una rutina asociada a ese día, tras pulsar en el día se deben mostrar un boton "Crear Rutina" y "Añadir Rutina"	OK	
5	Para que el usuario sepa que tiene una rutina asociada a ese día, debe aparecer un punto de cualquier color debajo del día correspondiente en el que tiene una rutina	NOK	No aparece o no se ve el punto

Figura 12. Plantilla Prueba de aceptación Tarea Calendario de rutinas

- **Done:**
  - o Tras haber realizado las pruebas, si no hay ninguna fallida, la tarea pasa a estado Done y se da por finalizada.

### 4.1.2 Herramientas de apoyo para la planificación y el seguimiento

Para un proyecto de desarrollo de software basado en prácticas ágiles es necesario contar con herramientas de planificación y seguimiento del mismo.

Muchas son las conocidas para este ámbito, como por ejemplo *Trello*<sup>15</sup>, *Worki*<sup>16</sup> o *Jira*<sup>17</sup>. En este proyecto, se ha optado por utilizar Jira, debido a su gran fama y versatilidad.

*Jira* es perfecta para el desarrollo ágil porque permite una planificación flexible, fomenta la colaboración entre equipos y *stakeholders* (personas interesadas), facilita la retroalimentación continua, se adapta fácilmente a cambios en los requisitos y prioridades del proyecto, y proporciona una visibilidad clara del progreso del trabajo en tiempo real. En resumen, es una herramienta que se alinea perfectamente con los principios y prácticas ágiles, lo que la convierte en una opción ideal para equipos que adoptan esta metodología.

En nuestro caso, muchas son las funcionalidades de *Jira* que hemos explotado para la gestión de nuestro proyecto como pueden ser: la creación de Épicas, Tareas, Subtareas, Incidencias, la creación de *Sprints*, creación de *workflows* personalizados o el uso de tableros Kanban, como el mostrado en la Figura 13, en entre otras muchas.

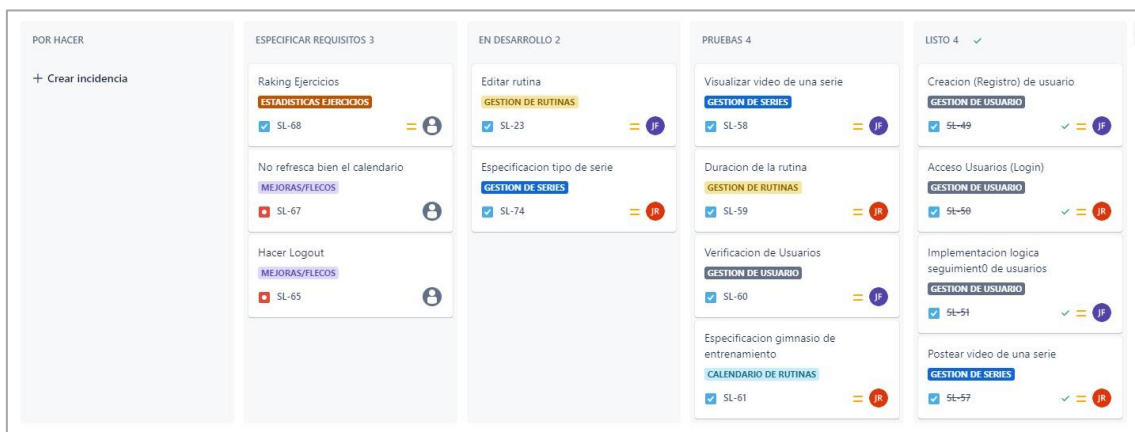


Figura 13. Tablero Kanban

<sup>15</sup> <https://trello.com>

<sup>16</sup> <https://cliente.tuneupprocess.com/web>

<sup>17</sup> <https://www.atlassian.com/es/software/jira>

## 4.2 Requisitos

Después de definir la metodología, se desarrolló la idea a través del *Lean Canvas*, donde comenzó a tomar forma.

Posteriormente, se llevó a cabo un estudio de mercado para analizar las funcionalidades de nuestros principales competidores y, basándonos en este estudio, se identificaron tanto las características principales de nuestros competidores como las nuestras, las cuales nos distinguirá de ellos.

Una vez analizadas nuestras características y las de nuestros competidores, se actualizó el *Lean Canvas* para plasmar de manera objetiva y con una mejor perspectiva las ventajas de nuestro producto y tras ello se comenzó a desarrollar proyección económica con el fin de estudiar la viabilidad del producto.

Como último paso, se construyó un Diagrama de Casos de Uso, ilustrado en la Figura 14, para tener una mejor y más clara perspectiva de las características de nuestro producto.

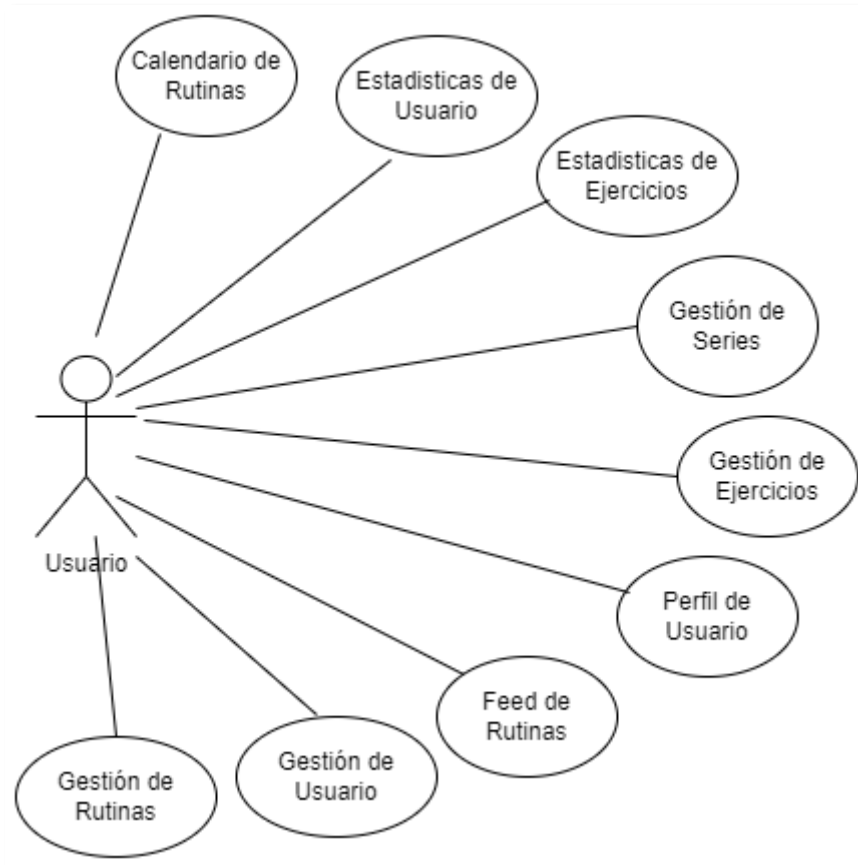


Figura 14. Diagrama CU – Características Social Lift

Los requisitos de una aplicación son las especificaciones funcionales y no funcionales que describen qué debe hacer la aplicación y cómo debe comportarse para satisfacer las necesidades de sus usuarios y *stakeholders*. Cada uno de los Requisitos Funcionales de nuestra aplicación están representados mediante Casos de Uso en la Figura 14.

Los requisitos pueden incluir aspectos como funcionalidades específicas que debe tener la aplicación, características de rendimiento, compatibilidad con diferentes dispositivos o sistemas operativos, seguridad, usabilidad y cualquier otro aspecto relevante para el desarrollo y funcionamiento de la aplicación.

Estos requisitos se recopilan y documentan durante la fase de análisis y diseño del proyecto y sirven como guía para el desarrollo, prueba e implementación de la aplicación.

- **Caso de Uso:** “Calendario de rutinas”.

Consiste en:

- Crear o Añadir una rutina: el usuario deberá podrá añadir una rutina guardada o asociar una nueva que ha creado.
- Navegar entre días y meses en el calendario.
- Visualizar las rutinas asociadas a un día.
- Eliminar rutina: el usuario debe poder eliminar la rutina asociada a un día esto no implica borrarla, solo quitarla de su día correspondiente.

Como **requisitos no funcionales** podemos destacar los siguientes:

- La navegación entre los días y meses debe de ser fluida.
- Las rutinas deben ser representadas mediante un punto de color personalizable debajo del día.
- El calendario y las rutinas se tiene que poder visualizar correctamente tanto en modo oscuro como en modo claro de la aplicación.

- **Caso de Uso:** “Gestión de Rutinas”.

Consiste en:

- Crear una rutina: este requisito funcional implica la creación explícita de la rutina, añadiéndole nombre, tipo, ejercicios, etc.
- Eliminar rutina: el usuario debe tener que poder eliminar sus rutinas-
- Editar rutina.
- Publicar rutina: el usuario es libre de poder publicar sus rutinas que realiza en un feed de rutinas.
- Buscar rutina: en un buscador correspondiente.
- Guardar rutina: el usuario podrá guardar rutinas que los usuarios publican en el feed.

Como **requisito no funcional** más importante de esta característica podemos destacar:

- El usuario podrá asociar un color personalizado para cada rutina que crea.

- **Caso de Uso:** “Gestión de Ejercicios”.

Consiste en:

- Crear Ejercicio: el usuario podrá crear ejercicios personalizados a su gusto.
- Borrar Ejercicio.
- Editar Ejercicio.
- Añadir ejercicios a una rutina.
- Buscar ejercicios: en un buscador correspondiente.

Como **requisito no funcional** podemos destacar que el usuario podrá asociar un icono según el grupo muscular afectado a cada ejercicio que crea.

- **Caso de Uso:** “Gestión de Series”.

En esta característica el usuario podrá controlar toda la gestión y las acciones que se pueden llevar a cabo con las series que se realizan en cada uno de los ejercicios, crear una serie, editarla, borrarla, postear un video de una serie y visualizarlo. En cuanto a los **requisitos no funcionales** podemos destacar los siguientes:

- El tiempo de posteo del video de una serie tiene que ser razonable y su visualización óptima.
- El usuario no podrá añadir más de un video a la serie.
- Solo se podrán ver los videos posteados por ti y los de las rutinas publicadas por usuarios a los que sigues,

- **Caso de Uso:** “Perfil de Usuario”.

Consiste en:

- Seguimiento de usuarios: el usuario podrá ver los usuarios que sigue y que le siguen.
- Ver rutinas publicadas: el usuario debe tener un listado de las rutinas que ha publicado.
- Ver rutinas guardadas: al igual que las publicadas, pero con las guardadas.
- Ver rutinas creadas: listado con las rutinas que has creado.
- Ver estadísticas de ejercicios: mediante gráficos y rankings.
- Ver perfiles externos.
- Postear imagen de perfil.
- Ver estadísticas corporales: mediante gráficos.

Como **requisitos no funcionales (RNF)** podemos destacar los siguientes:

- El usuario solo podrá ver las rutinas publicadas de aquellos usuarios a los que sigue.
- El posteo de la imagen debe ser en un tiempo razonable y con una calidad óptima.

- **Caso de Uso:** “Estadísticas de Usuario”.

Consiste en:

- Editar medidas corporales.
- Editar Peso.
- Editar IMC (Índice de Masa Corporal).
- Editar Altura.



- **Caso de Uso:** “Estadísticas Ejercicios”.

En relación a esta característica se definen dos requisitos funcionales muy genéricos, no obstante, mediante el transcurso de proyecto estos pueden ser especificados de una manera más concreta:

- Crear estadísticas por ejercicio: cada ejercicio tiene una gráfica de progreso de resultados.
- Ver ranking de ejercicios.

- **Caso de Uso:** “*Feed* Rutinas”.

Esta característica representa un muro en el que los usuarios podrán publicar sus rutinas a todos sus seguidores, de esta manera los usuarios podrán verlas, buscarlas y guardarlas para poder usarlas en un futuro o introducirlas en su plan de entrenamiento. Este *feed* tiene también una función para poder buscar usuarios. En cuanto a los requisitos no funcionales de esta característica podemos destacar:

- La actualización de las rutinas publicadas debe ser en tiempo real y constante.
- Se deben eliminar rutinas cuya fecha de publicación sea muy antigua, mayor a un mes. En el caso de las eliminadas se podrán buscar mediante el buscador.

- **Caso de Uso:** “Gestión de Usuario”.

Este caso un Usuario no registrado podrá registrarse en la aplicación y un Usuario ya registrado que podrá iniciar sesión en la misma y seguir a otros usuarios.

## 4.3 Diseño

El diseño en un proyecto de software se refiere al proceso de crear una estructura y planificación detallada para el desarrollo de un sistema. Involucra tomar los requisitos y especificaciones del software y traducirlos en un conjunto de componentes y módulos que trabajarán juntos para cumplir con los objetivos del proyecto.

Con la toma de requisitos especificada en el apartado anterior el siguiente paso es realizar el diseño de nuestro sistema software tanto a nivel de idea como a nivel de diseño de la estructura software del *frontend*, así como diseño a nivel de comportamiento del mismo.

En este capítulo nos centraremos en tres tipos de diseño concretos:

- El diseño de la IGU.
- El diseño de la Arquitectura Frontend.
- El diseño de los componentes de *Flutter*.

### 4.3.1 Diseño de la IGU

El diseño de la interfaz gráfica de usuario se llevó a cabo mediante un trabajo multidisciplinario entre desarrolladores y diseñadoras del grado de Bellas Artes de la UPV. Este trabajo conjunto se ha desarrollado desde las etapas iniciales del proyecto hasta la culminación del mismo.

Los desarrolladores realizábamos mockups de muy baja calidad, como se puede observar en la Figura 15, o a mano y se los entregábamos a las diseñadoras, quienes los rehacían utilizando *Figma*. Este enfoque permitió iterar rápidamente sobre los conceptos de diseño y asegurar que la interfaz fuera tanto atractiva como funcional.

Este trabajo multidisciplinario se basó en experiencias previas de colaboración con estas diseñadoras, lo que facilitó una comunicación fluida y efectiva. Además, el uso de *Figma* facilitó la comunicación visual y la alineación entre los equipos, garantizando que los diseños añadieran requisitos funcionales atractivos para añadir a ciertas características.

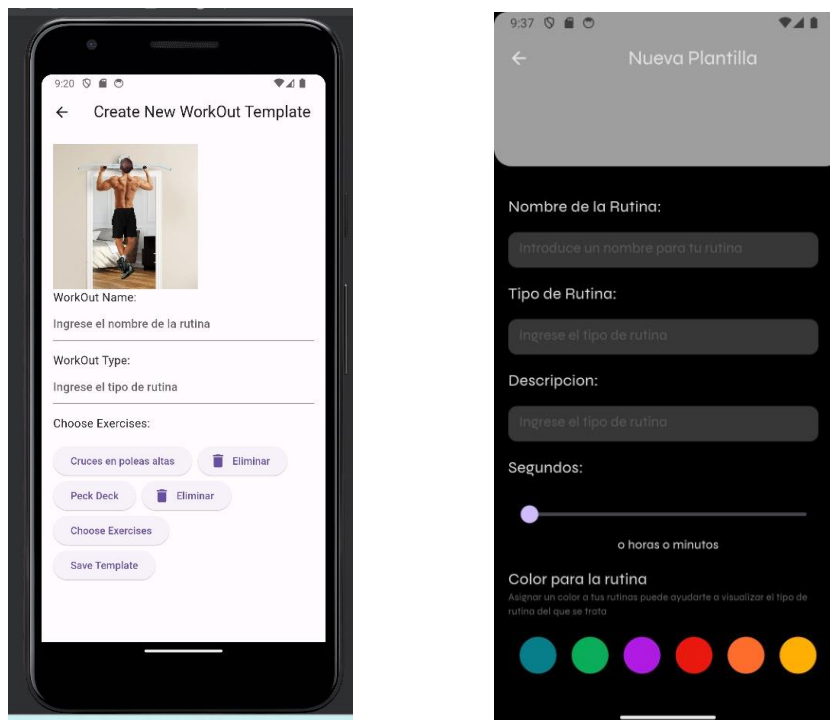


Figura 15. De izquierda a derecha. Mock-up y Diseño pantalla Nueva Rutina

### 4.3.2 Arquitectura *Frontend*

La arquitectura *frontend* de un proyecto desempeña un papel fundamental en la creación de experiencias de usuario fluidas y atractivas. Es la base sobre la cual se construyen las interfaces de usuario interactivas con las que los usuarios finales interactúan directamente. En esta sección, explicaremos en detalle cómo se estructura y organiza el *frontend* de nuestro proyecto para garantizar un rendimiento óptimo, una mantenibilidad sólida y una experiencia de usuario excepcional.

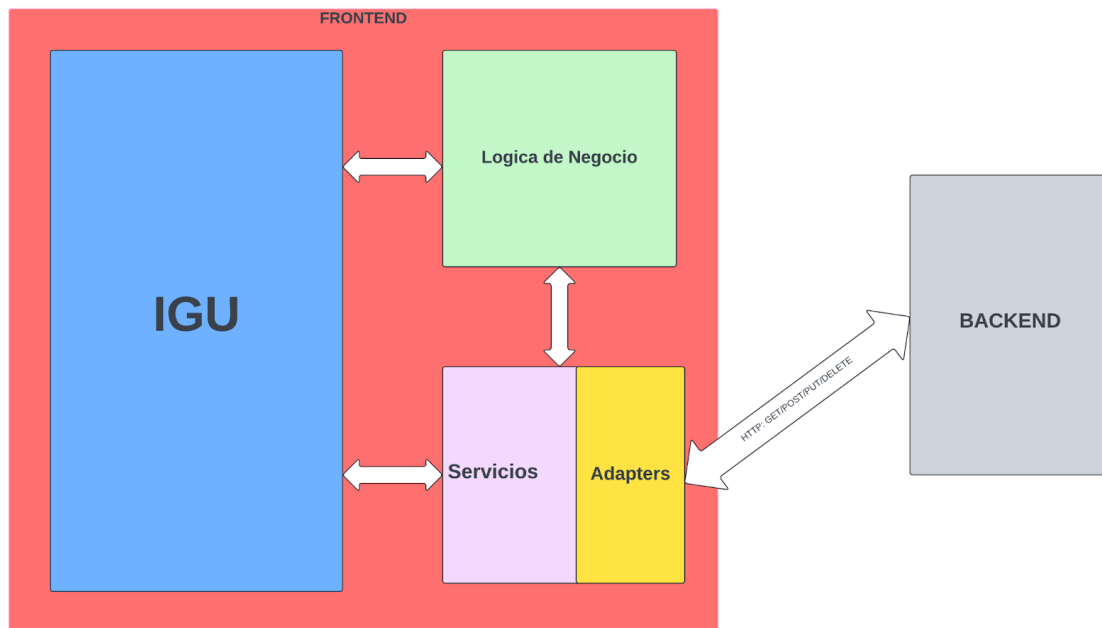


Figura 16. Arquitectura Frontend Social Lift

En este caso la arquitectura global, mostrada en la Figura 16, está basada en una arquitectura Cliente-Servidor, en la que el cliente es el *frontend* de la aplicación y el servidor el *backend*.

La arquitectura *frontend* de *Social Lift* está basada en una Arquitectura basada en Capas, este enfoque separa la aplicación en distintas capas:

- **Capa de Presentación o IGU:** en ella se encuentran todos los widgets (componentes gráficos) relacionados con cada una de las interfaces graficas de la aplicación.
- **Capa de Logica de Negocio:** es la responsable de la implementación y gestión de la funcionalidad central de la aplicación, separando las preocupaciones del dominio de negocio de otras capas del sistema, como la presentación y el acceso a datos. Esto facilita la mantenibilidad, escalabilidad y robustez del software.
- **Capa de Servicios:** el *frontend*, en este caso el cliente, realiza peticiones HTTP (GET, POST, PUT y DELETE) al servidor a través de la Capa de Servicios, cuya implementación lleva implícita una subcapa denominada *Adapters*, que es la encargada de realizar los mapeos correspondientes de los modelos de datos entre las dos partes de la arquitectura, tanto peticiones del *frontend* como respuestas del *backend*.

### 4.3.3 Diagrama de clases

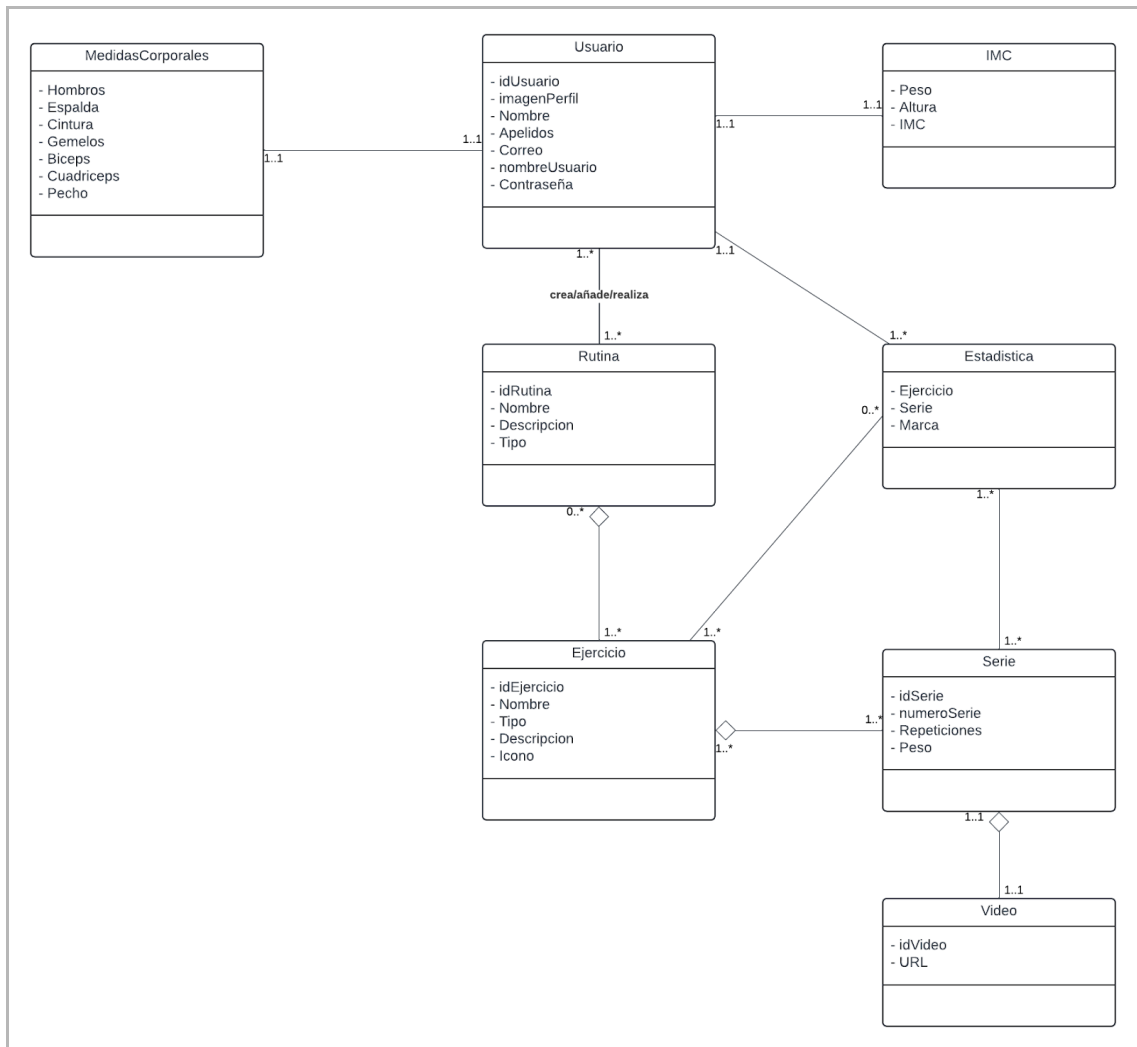


Figura 17. Diagrama de Clases Social Lift

Los diagramas (Figura 17) de clases son una herramienta crucial en el proceso de desarrollo de software porque ayudan a visualizar y planificar la estructura del sistema, facilitan la comunicación entre equipos y contribuyen a una implementación más sólida y mantenible del software.

#### 4.3.4 Diagrama de *widgets* de Flutter

Flutter utiliza *widgets* como bloques de construcción fundamentales para diseñar y construir interfaces de usuario (UI) de manera declarativa y eficiente.

En Flutter, un *widget* es un elemento de construcción visual que describe una parte de la interfaz de usuario. Prácticamente todo en Flutter es un *widget*, desde los componentes más básicos como texto y botones hasta *layouts* completos y aplicaciones enteras. Los *widgets* no solo controlan la apariencia de la aplicación, sino también su comportamiento y estructura.

Para esta sección mostraremos un sencillo diagrama de componentes (*widgets*), en la Figura 18, de una de las ventanas de nuestra aplicación como es la ventana del calendario, este diagrama de componentes es a gran escala, ya que no podemos realizarlo explícitamente (mostrando todos los *widgets*) ya que sería difícil de entender y demasiado extenso.

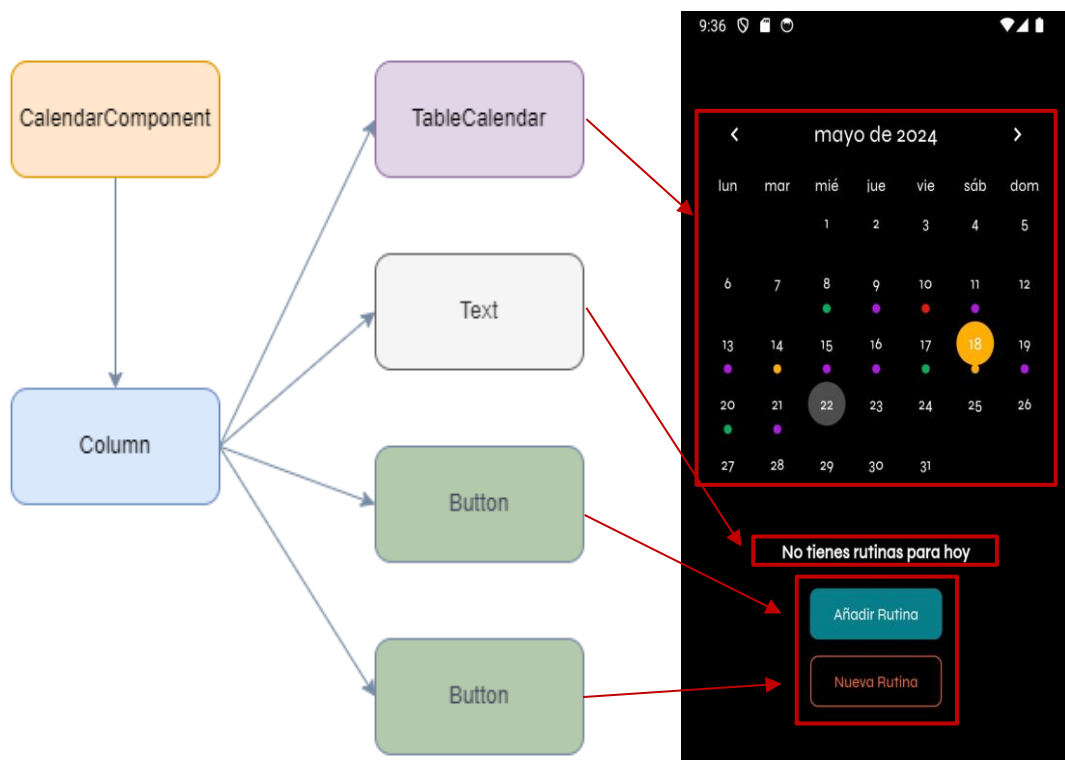


Figura 18. Diagrama de Componentes Pantalla Calendario

## 4.4 Programación

### 4.4.1 Aplicación de patrones

La aplicación de patrones de diseño en el desarrollo de software conlleva una serie de beneficios, que incluyen la reutilización de soluciones probadas, la mejora de la calidad del código, la flexibilidad y adaptabilidad, la facilitación de la comunicación y la promoción de la escalabilidad. Estos beneficios ayudan a construir sistemas robustos, mantenibles y eficientes que cumplen con los requisitos del negocio de manera efectiva.

Debido al contexto que nos ofrece *Social Lift*, como el hecho de la explicación en esta memoria de un desarrollo *frontend* y la adaptabilidad de uno patrones concretos a este tipo de desarrollo, hace que los siguientes patrones sean los que mejor se adaptan, hasta el momento, a este desarrollo *frontend*.

#### ❖ Patrón Fachada

Es un patrón de diseño estructural que proporciona una interfaz simplificada a una biblioteca, un *framework* o cualquier otro grupo complejo de clases.

Una *fachada*, cuyo diagrama se muestra en la Figura 19, es una clase que proporciona una interfaz simple a un subsistema complejo que contiene muchas partes móviles. Una fachada puede proporcionar una funcionalidad limitada en comparación con trabajar directamente con el subsistema. Sin embargo, tan solo incluye las funciones realmente importantes para los clientes.

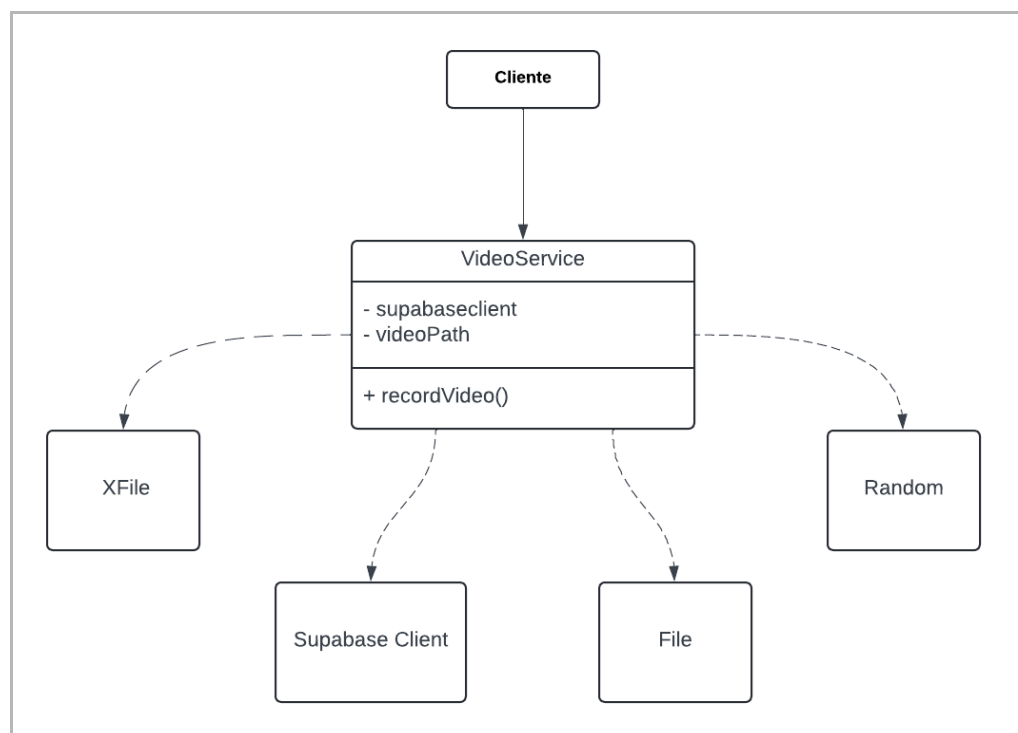


Figura 19. Diagrama Patrón Fachada VideoService

```

class VideoService {
    final supabaseclient = Supabase.instance.client;
    String videoPath = 'public/';

    Future<String> recordVideo() async {
        String randomNumber = '_${Random().nextInt(900000) + 100000}.mp4';
        videoPath += '${UserLogged.getUserLogged()!.nombreUsuario}/${randomNumber}';

        final picker = ImagePicker();
        final XFile? pickedFile = await picker.pickVideo(source: ImageSource.camera);
        if(pickedFile != null){
            final filePath = File(await pickedFile.path);

            final String path = await supabaseclient.storage.from('video').upload(
                videoPath,
                filePath,
                fileOptions: const FileOptions(cacheControl: '3600', upsert: false),
            );
        }

        String publicUrl = supabaseclient
            .storage
            .from('video')
            .getPublicUrl(videoPath);

        return publicUrl;
    }
}

```

Figura 20. Código Patrón Fachada VideoService

Esta implementación ha servido para realizar una fachada de cuatro subsistemas como se puede observar en la Figura 20. Esta fachada agrupa los siguientes subsistemas: *XFile*, *Supabase Client*, *File* y *Random* y su función es grabar un video a través de la cámara, almacenarlo en un *storage* de nuestro servidor de base de datos y devolvernos su URL, para posteriormente visualizarlo o asignarla a la entidad correspondiente.

#### ❖ Patrón *Provider* o *State Management*

Este patrón se utiliza para gestionar el estado de la aplicación de manera eficiente y reactiva.

El patrón *Provider*, mostrado en la Figura 21, es ampliamente utilizado en *Flutter* debido a su simplicidad y eficacia. Permite compartir y gestionar el estado de manera sencilla en toda la aplicación, lo que facilita la actualización de la interfaz de usuario en respuesta a cambios en el estado.

En este caso esta implementación se ha llevado a cabo para compartir o modificar el estado del usuario mediante la interfaz de usuario.



```
class UserState extends ChangeNotifier {
  Usuario? _user;

  Usuario? get user => _user;

  void setUser(Usuario user) {
    _user = user;
    notifyListeners();
  }
}
```

Figura 21. Patrón Provider

### ❖ Patrón Singleton

Es un patrón de diseño creacional que nos permite asegurarnos de que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia, ilustrado en la Figura 22 y 23.

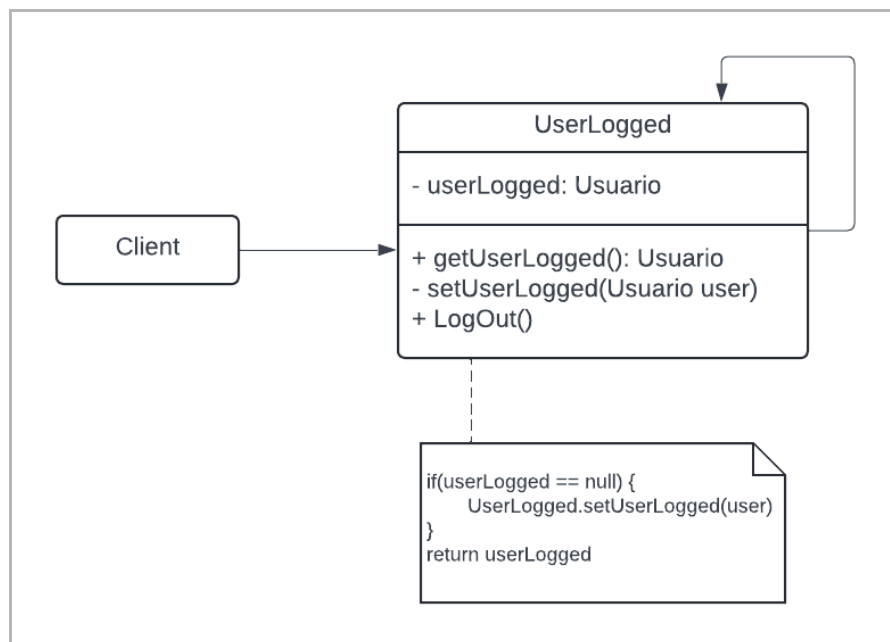


Figura 22. Diagrama Patrón Singleton

```
class UserLogged {
  static Usuario? userLogged;

  static Usuario? getUserLogged() => userLogged;

  static void setUserLogged(Usuario? user) => userLogged = user;

  static void Logout() => userLogged = null;
}
```

Figura 23. Patrón Singleton

En este caso, este patrón, ha sido aplicado debido a la necesidad de tener un acceso global al usuario logueado en la aplicación, en este caso, tanto el patrón *Provider* explicado anteriormente como este proveen una instancia global a la misma variable, pero el patrón *Provider* solo se puede usar en aquellos casos en los cuales existe un contexto, es decir, solo se puede usar en los métodos de construcción de las IGUs, en la renderización de los elementos de la interfaz.

Existía una necesidad de acceder al usuario *logueado* desde la capa de servicios de la aplicación y la solución a esto fue la creación de un patrón *Singleton*.

#### 4.4.2 Refactorizaciones

Refactorizar es el proceso de mejorar la estructura interna de un código existente sin alterar su comportamiento externo. Su objetivo principal es hacer que el código sea más comprensible, mantenible y extensible, lo cual modifica futuras correcciones y modificaciones

La refactorización más importante que se ha llevado a cabo en este proyecto es en los *mappings* (mapeos) de las peticiones y respuestas al *backend*. Tras una primera implementación los métodos mapeadores como la lógica de negocio de la aplicación se implementan en una misma clase como muestra en la siguiente Figura:

```
class Serie {
    int? idSerie;
    int numeroSerie;
    double peso;
    String tipoPeso;
    int numeroRepeticiones;
    Ejercicio? ejercicio;
    String? video;

    Serie({
        this.idSerie = 0,
        required this.numeroSerie,
        required this.peso,
        required this.tipoPeso,
        required this.numeroRepeticiones,
        required this.ejercicio,
        this.video
    });

    factory Serie.fromJson(Map<String, dynamic> json){
        return Serie(
            idSerie: json['idSerie'],
            numeroSerie: json['numeroSerie'],
            peso: json['peso'],
            tipoPeso: json['tipoPeso'],
            numeroRepeticiones: json['numeroRepeticiones'],
            ejercicio: json["ejercicio"] != null && json["ejercicio"] != ""
                ? Ejercicio.fromJson(json["ejercicio"])
                : null,
            video: json['video']
        ); // Serie
    }

    Map<String, dynamic> toJson() {
        final Map<String, dynamic> data = new Map<String, dynamic>();
        data['idSerie'] = this.idSerie;
        data['numeroSerie'] = this.numeroSerie;
        data['peso'] = this.peso;
        data['tipoPeso'] = this.tipoPeso;
        data['numeroRepeticiones'] = this.numeroRepeticiones;
        data['ejercicio'] = ejercicio?.toJson();
        data['video'] = this.video;
        return data;
    }
}
```

Figura 24. Clase Serie

Tras el estudio de una nueva implementación de estos *mappings* más adecuada, se llevó a cabo la creación de unas clases denominadas *Adapters*, como se muestra en la Figura 26, para cada una de estas entidades de nuestra lógica de negocio.

```
class Serie {  
  
    int? idSerie;  
    int numeroSerie;  
    double peso;  
    String tipoPeso;  
    int numeroRepeticiones;  
    Ejercicio? ejercicio;  
    String? video;  
  
    Serie({  
        this.idSerie = 0,  
        required this.numeroSerie,  
        required this.peso,  
        required this.tipoPeso,  
        required this.numeroRepeticiones,  
        required this.ejercicio,  
        this.video  
    });  
  
}
```

Figura 25. Clase Serie tras la refactorización

```
class SerieAdapter implements IAdapter{  
    ExerciseAdapter exerciseAdapter = ExerciseAdapter();  
  
    @override  
    Serie fromJson(Map<String, dynamic> json){  
        return Serie(  
            idSerie: json['idSerie'],  
            numeroSerie: json['numeroSerie'],  
            peso: json['peso'],  
            tipoPeso: json['tipoPeso'],  
            numeroRepeticiones: json['numeroRepeticiones'],  
            ejercicio: json["ejercicio"] != null && json["ejercicio"] != ""  
                ? exerciseAdapter.fromJson(json["ejercicio"])  
                : null,  
            video: json['video']  
        );  
    }  
  
    @override  
    Map<String, dynamic> toJson<T>(T objeto) {  
        if (objeto is Serie) {  
            final Map<String, dynamic> data = new Map<String, dynamic>();  
            data['idSerie'] = objeto.idSerie;  
            data['numeroSerie'] = objeto.numeroSerie;  
            data['peso'] = objeto.peso;  
            data['tipoPeso'] = objeto.tipoPeso;  
            data['numeroRepeticiones'] = objeto.numeroRepeticiones;  
            data['ejercicio'] = exerciseAdapter.toJson(objeto.ejercicio);  
            data['video'] = objeto.video;  
            return data;  
        } else {  
            throw ArgumentError('El parámetro debe ser una instancia de Serie');  
        }  
    }  
  
}
```

Figura 26. Clase SerieAdapter. Nueva implementación de los mappings

Esta refactorización sigue el principio de responsabilidad única al separar la lógica de serialización/deserialización de la clase principal. Como resultado, cada clase tiene una responsabilidad clara y bien definida, lo que facilita su comprensión y mantenimiento. Además, esta refactorización promueve la reutilización del código al permitir que la lógica de mapeo sea compartida con otras clases si es necesario. En resumen, este enfoque mejora la modularidad, la reutilización y la facilidad de mantenimiento del código.

Esta refactorización se ha llevado a cabo para todas las entidades de la lógica de negocio y *mappings* de este proyecto.

#### 4.4.3 Desafíos de programación

Sumergirse en un desarrollo software siempre es un desafío, implica el desarrollo lógico de soluciones eficientes, reutilizables y mantenibles.

Una implementación con estos tres rasgos es difícil de conseguir y aplicarlos es todo un desafío, no obstante, se suele conseguir en cierta manera.

En un desarrollo siempre hay partes del mismo y requisitos que son más difíciles de implementar y a los cuales se le dedica más tiempo. En el caso del desarrollo *frontend* de esta aplicación son algunos los desafíos que hemos tenido que abordar.

##### ❖ Desarrollo del calendario de rutinas

En la implementación del calendario de rutinas se pueden diferenciar dos grandes retos.

Por una parte, podemos destacar uno de ellos, la **asignación de las rutinas** en sus días correspondientes, esta implementación consiste en obtener las rutinas de nuestro servidor de base de datos y mapear cada una de ellas en su día correspondiente.

Fue un desarrollo largo y tedioso por el manejo de las fechas entre el *backend* y el *frontend*, tras largos intentos de desarrollos fallidos, debido a que las rutinas no aparecían en su día correspondiente, se tuvo que realizar una personalización de las fechas llegadas del *backend* de la aplicación, debido a que el tipo de datos no se convertía correctamente a el utilizado en el *frontend*.

No se convertía correctamente debido a lo siguiente, todo ello mostrado en la Tabla 7:

- El dato llegado en la petición era de tipo *Date* y nuestro *frontend* maneja fechas con el tipo de dato *DateTime*.
- Esto implica una conversión de *Date* a *DateTime* de la siguiente manera: `DateTime.parse(rutina!.fecha).toLocal()`

<i>DateTime</i>	<i>Date</i>
2024-04-21 00:00:00.000Z	2024-04-21T00:00:00.000+00:00

Tabla 7. Tabla Ejemplo Conversiones

## Una aplicación para la planificación y gestión de rutinas de gimnasio. Desarrollo del Frontend

Tras realizar la conversión y como se puede observar en la columna *DateTime*, el nuevo valor de fecha adquiere un carácter “Z” al final de los milisegundos de la misma, esto hacía que cuando se comparaba la fecha de la rutina con la del día en la que se quería asignar esa rutina en el calendario, nunca la asignaba, ya que no se consideraban iguales debido al carácter “Z”.

Tras varios desarrollos fallidos intentando solucionar el problema se llegó a la siguiente conclusión:

- El campo hora, minutos y segundos no nos interesaban por lo que podíamos crear un *DateTime* con solo el siguiente formato *YYYY-MM-DD*.
- Había que normalizar la fecha para crearla con el formato descrito en el punto anterior.
- Y para una comparación estricta y que no hubiera posibilidad de fallo en la asignación de la rutina no debíamos comparar la fecha entera, si no comparar el año, mes y día de manera independiente.

Con este nuevo análisis se realizó la nueva implementación y resultó exitosa.

Por otra parte, y relacionado con este desarrollo, el segundo de los desafíos fue el **diseño del calendario**, ilustrada en la Figura 27.

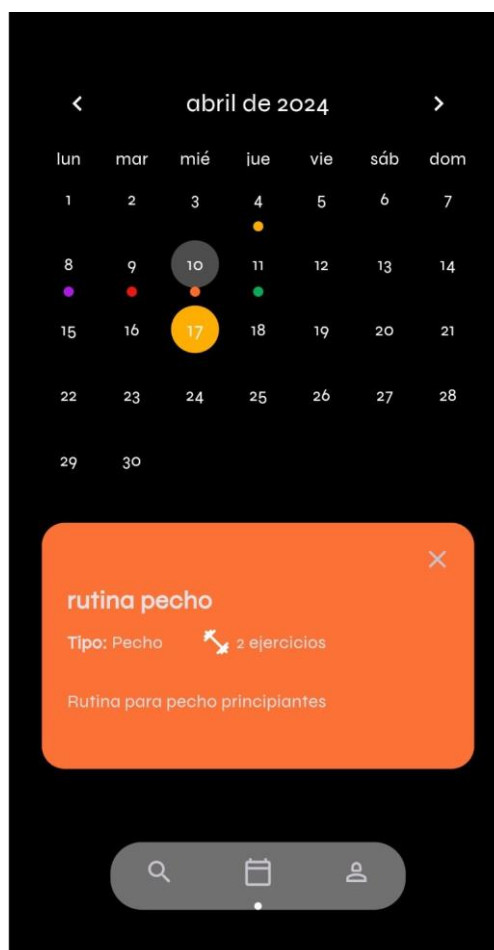


Figura 27. Pantalla Calendario

La implementación del diseño mostrado en el mockup nos sugiere la implementación de los siguientes requisitos a nivel de diseño *frontend*:

- Los *markers* (situados debajo de los días en el calendario si existe rutina ese día), deben ser redondos y del color definido al crear la rutina (Figura 28).
- Debe haber dos tipos de circunferencias, una que indique el día actual y otra que indique el día seleccionado, deben ser redondas y con los colores indicados en el mockup.
- El color de los símbolos, letras y números del calendario deben ser blancos.
- Deben aparecer dos flechas a cada lado del mes para poder navegar entre los meses.

Todos estos requisitos se resumen en la implementación bastante extensa y difícil de entender, se debe tener en cuenta de que *Dart* nos ofrece un widget llamado *TableCalendar* que, nos ofrece un calendario predeterminado, pero que se tuvo que rediseñar porque no cumplía, en absoluto, ninguno de los requisitos de diseño especificados anteriormente.

```
calendarBuilders: CalendarBuilders(  
  markerBuilder: (context, date, events) {  
    // Normaliza la fecha para compararla con las fechas de las rutinas  
    DateTime normalizedDate =  
      DateTime(date.year, date.month, date.day);  
    // Verifica si hay una rutina asociada a la fecha normalizada  
    if (rutinasPorFecha.containsKey(normalizedDate)) {  
      String? color = rutinasPorFecha[normalizedDate]!  
        .color  
        ?.colorHex;  
      Color markerColor;  
      color != null  
        ? markerColor = Color(int.parse(color))  
        : markerColor = Colors.white;  
      // Renderiza el marcador con el color correspondiente  
      return Container(  
        width: 8,  
        height: 8,  
        decoration: BoxDecoration(  
          shape: BoxShape.circle,  
          color: markerColor,  
        ), // BoxDecoration  
      ); // Container  
    } else {  
      // No hay rutina asociada a esta fecha, no se muestra ningún marcador  
      return null;  
    }  
  }  
), // CalendarBuilders
```

Figura 28. Primer requisito: Requisito de los markers

## 4.5 Pruebas

En el ámbito del desarrollo de *software*, se realizan una amplia gama de pruebas, cada una dirigida a aspectos específicos del producto a evaluar. En esta sección, nos centraremos en dos tipos particulares de pruebas *frontend* que se han implementado durante el desarrollo del *software*: *widget testing* y pruebas de aceptación.

### 4.5.1 *Widget Testing*

El *widget testing* es crucial en el desarrollo de interfaces de usuario (UI). Estas pruebas se enfocan en validar el comportamiento y la apariencia de los componentes de la interfaz de usuario, como botones, formularios y menús desplegables. Al simular interacciones del usuario con estos elementos, se garantiza que la UI funcione correctamente en diferentes situaciones y dispositivos, lo que contribuye a una experiencia de usuario consistente y satisfactoria.

Dentro del entorno de *Flutter*, se pueden llevar a cabo diferentes tipos de pruebas de widgets, dependiendo del tipo de prueba que se quiere realizar. Estos tipos incluyen encontrar widgets, simular desplazamiento y simular interacciones del usuario. En este apartado nos centraremos en el *testing* de encontrar *widgets* o también llamado *Find widgets*.

Este subtipo de prueba consiste, como bien su palabra indica, en localizar widgets en un entorno de test.

Se refiere a la capacidad de localizar y seleccionar widgets específicos dentro de la jerarquía de la interfaz de usuario durante una prueba en el entorno de *Flutter*. Para lograr esto, se utilizan las clases *Finder* proporcionadas por el paquete *flutter\_test*, las cuales ofrecen herramientas convenientes para identificar widgets de manera eficiente.

Al realizar pruebas de *widgets*, es común necesitar encontrar *widgets* específicos para interactuar con ellos o para verificar su estado y comportamiento. La capacidad de encontrar *widgets* es esencial para simular interacciones de usuario y verificar que la interfaz de usuario responda correctamente.

El proceso de encontrar *widgets* puede implicar diferentes métodos, como buscar *widgets* por su tipo, identificarlos mediante una clave específica asignada a ellos, o incluso encontrar instancias particulares de un widget.

Durante el desarrollo del proyecto se ha realizado estos tipos de test a la mayoría de los widgets de la aplicación, entre ellos podemos destacar test en los widgets del Feed de rutinas, en las renderizaciones de las diferentes listas que hay en la aplicación, en el calendario de la misma, etc.

Debido a la gran cantidad de test que se han realizado y la gran dificultad que tiene la mayoría de ilustrarlos en este apartado concreto comentaremos un caso concreto y sencillo de entender de *widget testing* que se ha llevado a cabo en esta aplicación ha sido crear un entorno de prueba para el *widget CardRutina*, ilustrada en la Figura 29.



Figura 29. CardRutina

```
testWidgets('Test de CardRutina', (WidgetTester tester) async {
  // Define los datos de prueba
  final nombre = 'Test Nombre';
  final descripcion = 'Test Descripcion';
  final tipo = 'Tipo: Pecho';
  final numEjercicios = 6;
  final color = '1';
  final imageUrl = '';

  // Construye el widget CardRutina con los datos de prueba
  await tester.pumpWidget(
    MaterialApp(
      home: Scaffold(
        body: CardRutina(
          nombre: nombre,
          descripcion: descripcion,
          tipo: tipo,
          numEjercicios: numEjercicios,
          color: color,
          imageUrl: imageUrl,
        ), // CardRutina
      ), // Scaffold
    ), // MaterialApp
  );

  // Verifica que los elementos esperados se encuentren en el widget
  expect(find.text(nombre), findsOneWidget);
  expect(find.text(descripcion), findsOneWidget);
  expect(find.text('Tipo: ' + tipo), findsOneWidget);
  expect(find.text('$numEjercicios ejercicios'), findsOneWidget);
  expect(find.byIcon(Icons.close), findsOneWidget);
});
```

Figura 30. Test CardRutina

Mediante este código, mostrado en la Figura 30, se ha creado un entorno de prueba para *CardRutina*, lo construye con datos de prueba específicos y luego verifica que los elementos esperados estén presentes en el widget resultante, esto ayuda a garantizar que el widget se renderice correctamente y muestre los datos de manera adecuada en la interfaz de usuario.



#### 4.5.2 Pruebas de Aceptación

Las pruebas de aceptación son un tipo de pruebas que se realizan para validar que el sistema cumple con los criterios de aceptación definidos por el cliente, el usuario final o las partes interesadas. Estas pruebas se realizan desde la perspectiva del usuario y se centran en verificar si el software satisface los requisitos funcionales y no funcionales, así como en confirmar que el sistema se comporta de acuerdo con las expectativas del cliente.

Para el caso que nos atañe en este proyecto se han definido pruebas de aceptación para cada una de las tareas desarrolladas en el mismo, es decir, cada requisito de la aplicación implementado siempre lleva un plan de pruebas de aceptación asociado.

Calendario de rutinas			
Jesús Fierrez	Se mostrará un calendario, en el que en el día correspondiente, si el usuario tiene una rutina asignada, debe aparecer en ese día		
22-03-24			
PASOS	PRUEBAS REALIZADAS	RESULTADO	COMENTARIOS
1	Se debe poder visualizar un calendario mensual	OK	
2	Se podrá navegar entre los días y los meses	OK	
3	Si y solo si el usuario tiene rutina asociada a ese día, tras pulsar un día se debe mostrar la rutina asociada	OK	
4	Si y solo si el usuario no tiene una rutina asociada a ese día, tras pulsar en el día se deben mostrar un boton "Crear Rutina" y "Añadir Rutina"	OK	
5	Para que el usuario sepa que tiene una rutina asociada a ese día, debe aparecer un punto de cualquier color debajo del día correspondiente en el que tiene una rutina	FAIL	No aparece o no se ve el punto

Figura 31. Plan de Pruebas de Aceptación Calendario Rutinas

Se ha definido una estructura del plan de pruebas trivial para todos los requisitos, como se puede observar en la Figura 31.

Para que un requisito pueda darse como terminado el resultado de todos los pasos del plan de pruebas debe ser "OK", en caso contrario la tarea/s correspondiente/s a este requisito no se darán por terminadas y volverán al estado de desarrollo.

Tras trasladarlas a este estado en el *workflow* se analizará el problema y se estimará el tiempo necesario para darle solución, si esta nueva estimación entra dentro del alcance del Sprint en curso se llevará a cabo el proceso de desarrollo para resolver el problema para tras su solución volver a realizar el plan de pruebas. En caso contrario, este error/bug/fleco será añadido como tarea en el próximo Sprint de desarrollo.

En el caso que se muestra en la imagen, el error se pudo solucionar en el Sprint en curso, pero muchos han sido los casos en los cuales, tras la estimación, la solución no se ha podido llevar a cabo en el Sprint en curso.

Como conclusión final a este apartado, podemos decir que el *widget testing* nos ha servido para detectar precozmente errores en la renderización de diferentes widgets de la aplicación, así como a mejorar la disposición de los distintos elementos en diferentes widgets.

Por otro lado, las Pruebas de Aceptación garantiza un ciclo de desarrollo iterativo y adaptable, donde los problemas se abordan de manera oportuna y organizada. Aunque en algunos casos las soluciones no pueden implementarse dentro del Sprint en curso, este método nos permite una gestión eficaz de errores y mejora continua del sistema, asegurándonos que la aplicación final cumpla con los estándares de calidad y las expectativas del usuario.

## 5. Cronología del TFG

---

La gestión de un proyecto software es un pilar fundamental para que su desarrollo sea exitoso, para ello hay que hacer un estudio de la cronología del mismo utilizando herramientas o software de gestión para la causa.

En primer lugar y tras la definir las características de nuestro producto, que en Jira están ilustradas como Épicas y definir los requisitos tanto funcionales y no funcionales correspondientes a cada característica hay que introducirlas todas al backlog de nuestro proyecto.

Este backlog se ira mostrando en la explicación de cada Sprint, ya que existe cierta complicación mostrándolo como un todo y que todo se vea de manera clara.

Como ya mencionamos en el apartado de la metodología cada requisito está definido bajo una épica que corresponde a una característica del producto y sigue un *workflow* determinado. Como se puede observar en la Figura 32, las tareas del backlog seguirán la siguiente escala de prioridad:

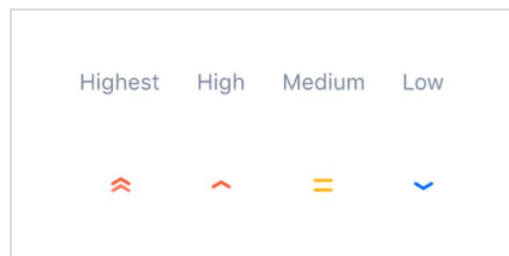


Figura 32. Escala de Prioridad

- **Highest:** prioridad de las tareas que son necesarias para cumplir las funcionalidades básicas de la aplicación, de los competidores (Calendario de rutinas, crear rutinas, ejercicios, etc.) y también todas aquellas tareas relacionadas con la evaluación de la idea de negocio (Estudio de mercado y competidores, proyección de ingresos y gastos, etc.).
- **High:** enfocada a todas aquellas tareas relacionadas con la mejora de la aplicación y su cumplimentación, es decir, tareas relacionadas con las tareas de prioridad *Highest* como, por ejemplo, borrar rutinas, editar rutinas, borrar series, borrar ejercicios y otras relacionadas con las estadísticas del usuario, ya que se piensa que una aplicación como es *Social Lift*, que gestiona gran cantidad de datos, debería de tener ciertos baremos estadísticos.
- **Medium:** son aquellas tareas que entran dentro del alcance del proyecto pero que determinamos que no tiene tanta prioridad como las mencionadas anteriormente, estas son la creación de un feed de usuarios, publicación de rutinas, registro y login de usuario, etc.
- **Low:** prioridad destinada a aquellas tareas que no entran dentro del alcance del proyecto ya sea porque no es estrictamente necesario o porque su estimación de tiempo es muy alta. Podemos destacar tareas relacionadas con coaching virtual, IA para la recomendación de ejercicios, entre otras.

Para el desarrollo del proyecto se ha seguido una línea temporal concreta, ilustrada en la Figura 33, especificando así tiempos concretos para llegar al objetivo del mismo:

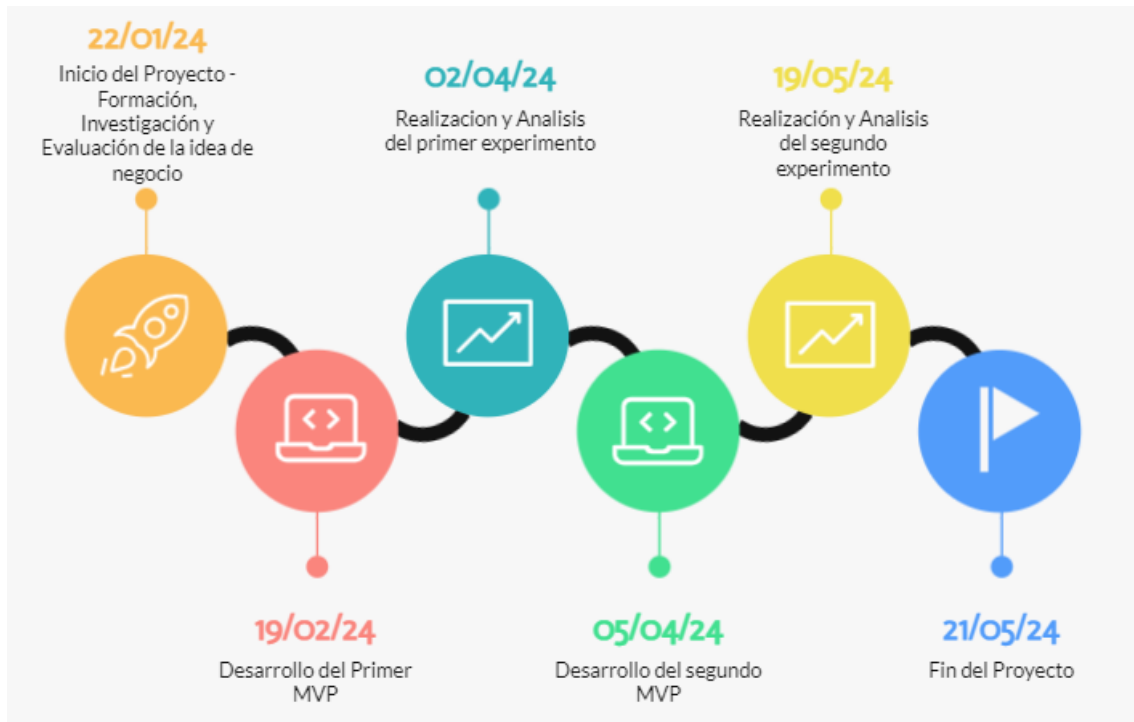


Figura 33. Línea temporal del proyecto

En relación, al Sprint 0 se han llevado actividades relacionadas con la evaluación de la idea de negocio y formación en la tecnología, las cuales son primordiales realizar previamente al comienzo del desarrollo del producto.

Durante este periodo, en que se ha invertido un mes de tiempo, se ha realizado una formación autodidacta de *Flutter*, mediante documentación explícita de la página oficial de *Flutter*<sup>18</sup>, foros, tutoriales y *Chat-gpt*<sup>19</sup>.

También se ha llevado a cabo una investigación y evaluación de la idea de negocio utilizando diferentes técnicas que ya hemos comentado en el Capítulo 2.

<sup>18</sup> <https://docs.flutter.dev/>

<sup>19</sup> <https://chatgpt.com/>

## 5.1 Desarrollo del primer MVP

Como se ilustra en la imagen anterior, el primer MVP estará desarrollado tras un primer tramo de desarrollo que abarca dos *Sprints* de 3 semanas cada uno de ellos, el Sprint 1 y el Sprint 2, a la finalización de cada uno de ellos se realiza una reunión de retrospectiva para revisar cómo ha ido el Sprint y mejorar de cara a los próximos.

Primeramente, se lleva a cabo del desarrollo del primer Sprint (Figura 34), antes del desarrollo de las tareas se ha llevado una previa especificación de requisitos.

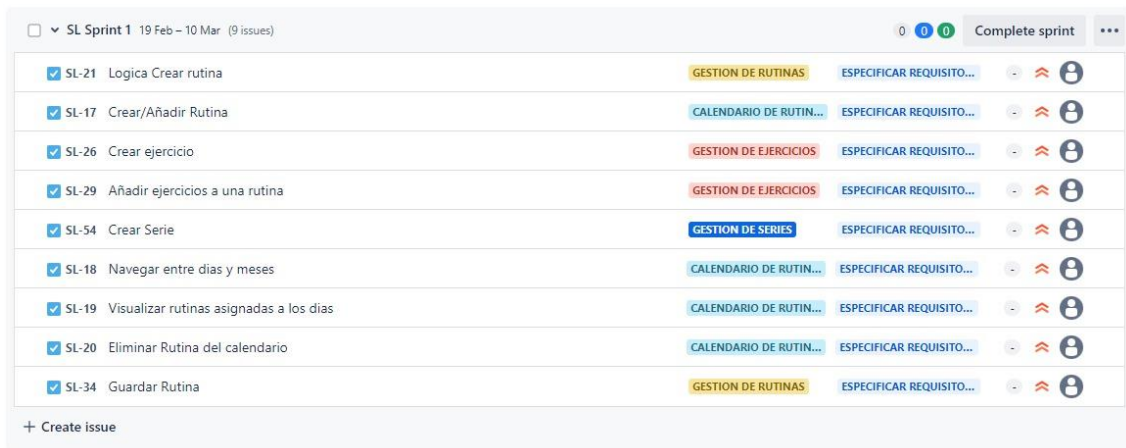


Figura 34. Estado Tareas del Sprint 1 al comienzo

Tras la especificación de requisitos, todas las tareas pasan por el estado de desarrollo, posteriormente pasan a pruebas y si las pruebas son válidas pasa al estado de finalizadas (Listo) como se observa en la Figura 35.

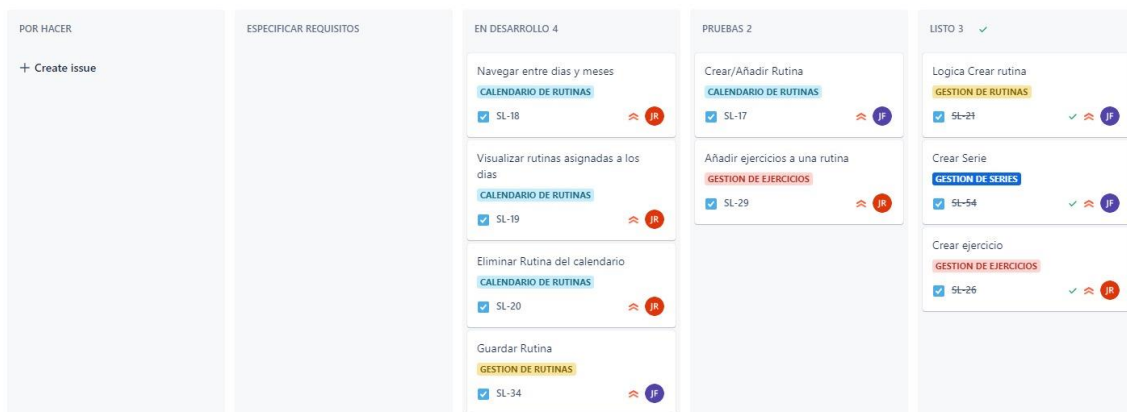


Figura 35. Tablero Kanban Sprint 1

El último día de cada Sprint se realizaron reuniones de retrospectiva, para confrontar opiniones y proponer mejoras sobre el transcurso del Sprint 1 y llevarlas a cabo en futuros *Sprints*.

Tras la finalización del *Sprint 1*, dio comienzo el *Sprint 2*, los requisitos seleccionados para este Sprint fueron los siguientes (Figura 36):

Issue ID	Title	Category	Action
SOC-8	Borrar Serie	GESTION DE SERIES	ESPECIFICAR REQUISITO...
SOC-9	Editar Serie	GESTION DE SERIES	ESPECIFICAR REQUISITO...
SOC-10	Crear estadísticas por ejercicio realizado	ESTADISTICAS EJERCICIO...	ESPECIFICAR REQUISITO...
SOC-11	Editar peso, altura e IMC	ESTADISTICAS USUARIO	ESPECIFICAR REQUISITO...
SOC-12	Editar medidas corporales	ESTADISTICAS USUARIO	ESPECIFICAR REQUISITO...
SOC-13	Visualización de rutinas creadas	GESTION DE RUTINAS	ESPECIFICAR REQUISITO...
SOC-14	Visualización estadísticas por ejercicio	ESTADISTICAS EJERCICIO...	ESPECIFICAR REQUISITO...
SOC-15	Visualización estadísticas corporales	ESTADISTICAS USUARIO	ESPECIFICAR REQUISITO...
SOC-16	Buscar ejercicios	GESTION DE EJERCICIOS	ESPECIFICAR REQUISITO...
SOC-17	Borrar ejercicio	GESTION DE EJERCICIOS	ESPECIFICAR REQUISITO...
SOC-18	Borrar rutina	GESTION DE RUTINAS	ESPECIFICAR REQUISITO...

Figura 36. Tareas Sprint 2

Tras la reunión de retrospectiva del Sprint 2 se detectaron una serie de bugs que debían ser solucionados en *Sprints* posteriores y que se verán reflejados como tareas en futuros *Sprints*. También, se confrontan opiniones sobre las buenas prácticas metodológicas y se comentaron aspectos a mejorar como:

- Asignarse como responsable en las tareas que realizas
- Realizar planes de pruebas de aceptación más exhaustivos y no tan genéricos como se venía haciendo en estos dos *Sprints*.

Este Sprint 2 fue previo a la realización del primer experimento, es decir, con la finalización de este Sprint se dio por creado el primer MVP de nuestra aplicación, el cual sería utilizado para el primer experimento.

En el primer tramo de desarrollo, se abarcaron funcionalidades que cubren las necesidades básicas de una aplicación típica de gestión de rutinas de gimnasio, mejorándola siempre con un diseño *frontend* limpio, bonito y llamativo, es por eso que introdujeron en el Sprint 1 y 2 las tareas más prioritarias y las que daban forma a un MVP básico para la gestión de rutinas de entrenamiento.

### 5.1.1 Primer Experimento

Después de completar el primer MVP de *Social Lift*, el siguiente paso fue llevar a cabo un experimento que implicaba la realización de una encuesta por parte de un grupo de *early adopters* compuesto por 10 personas. Es relevante señalar que todos estos *early adopters* son usuarios regulares de gimnasios, como se muestra en la Figura 37. Además, dentro de este grupo, hay 3 estudiantes de Ingeniería Informática y una estudiante de Marketing y Publicidad.

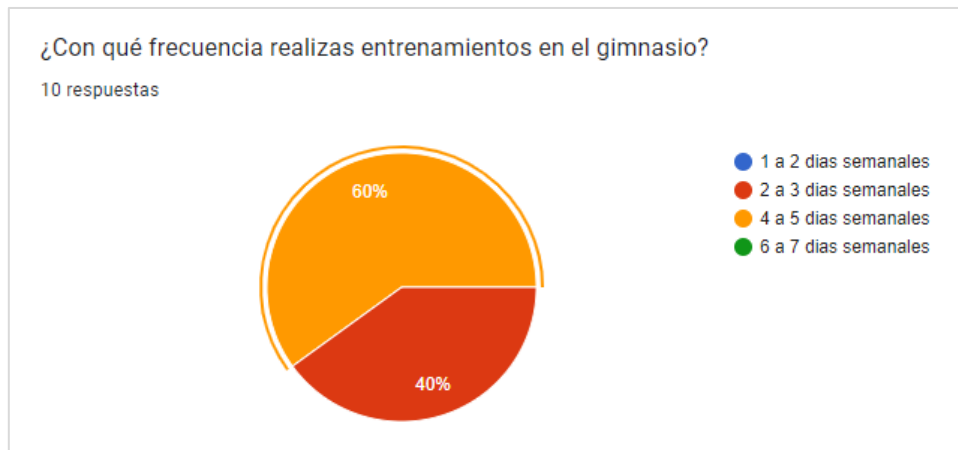


Figura 37. Gráfico ¿Con qué frecuencia realizas entrenamientos en el gimnasio?

La encuesta estaba formada por 7 secciones de preguntas de todo tipo (respuesta abierta, escala lineal, varias opciones, etc.), una de las secciones relacionada con datos correspondientes a experiencia con sus rutinas y en gimnasios y 6 relacionadas cada una con las IGUs (Interfaces Gráficas de Usuario) más importantes de la primera versión de la aplicación.

El objetivo del experimento era realizar una encuesta de satisfacción a los *early adopters* con el fin de realizar una validación de las diferentes IGU (Interfaces gráficas de usuario) y a su vez definir nuevos requisitos para implementaciones futuras de la aplicación.

La estructura de la encuesta consta de 8 secciones, como ya se ha comentado anteriormente, cada una de las secciones está compuesta por preguntas:

- **Sección 1: Introducción.** Las preguntas que los encuestados tenían que responder son las siguientes:
  - ¿Con qué frecuencia realizas entrenamientos en el gimnasio?
  - ¿Dónde apuntas tu progreso diario en los diferentes ejercicios de tus rutinas?
  - ¿Cuál es tu fuente de información para añadir nuevos ejercicios en tus rutinas?
  - ¿Conoces alguna aplicación útil para gestionar tus entrenamientos de gimnasio? Si tu respuesta es Sí, indique cuál.

- **Secciones 2 a 7: Validaciones de las IGUs y elicitación de nuevos requisitos.** Estas 6 secciones siguen una estructura de preguntas idéntica:
  - ¿Qué impresión tienes de las pantallas de la aplicación? Tipo de respuesta: Escala numérica del 1 al 5, siendo 1 muy mala impresión y 5 muy buena.
  - ¿Consideras que la información presentada en las pantallas es clara y concisa? Tipo de respuesta: Escala numérica del 1 al 5, siendo 1 poco clara y concisa y 5 muy clara y concisa.
  - ¿Cómo calificas, del 1 al 10, tu nivel de satisfacción general con el estilo de las pantallas de la aplicación? Tipo de respuesta: Escala numérica del 1 al 10, siendo 1 poco satisfecho y 10 muy satisfecho
  - ¿Qué funcionalidades te gustaría ver añadidas a esta pantalla en futuras versiones de la aplicación? Tipo de respuesta: Respuesta abierta.

Como se puede observar en estas secciones, las 3 primeras preguntas están relacionadas con la validación de las IGUs y la cuarta y última con la elicitación de nuevos requisitos.

Tras la realización de la encuesta por parte de los *early adopters* y el análisis de los resultados obtenidos, sacamos las siguientes conclusiones de cada una de las secciones y preguntas:

Con respecto a la **Sección 1: Introducción.**

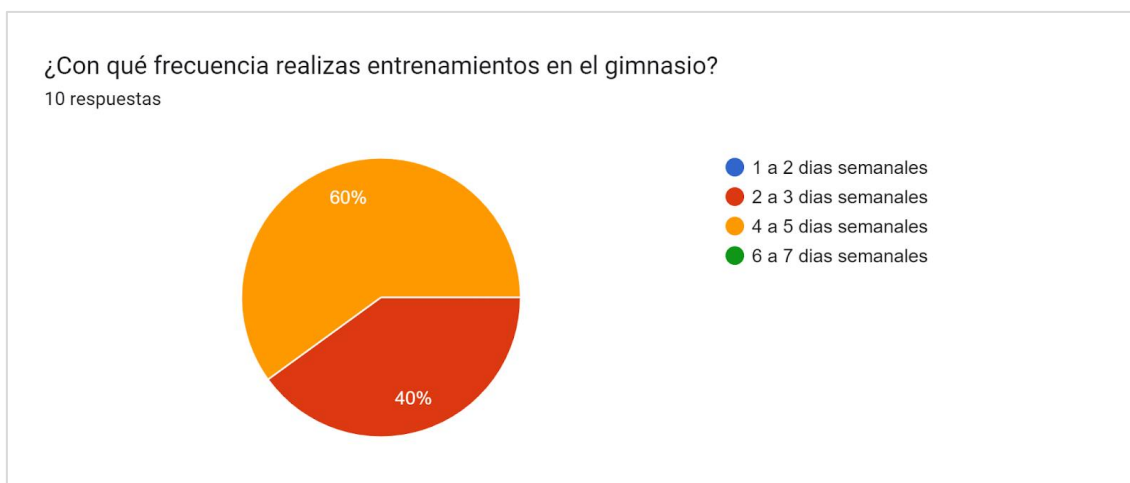


Figura 38. Gráfico ¿Con qué frecuencia realizas entrenamientos en el gimnasio?

La finalidad de esta pregunta era confirmar que nuestros *early adopters* son usuarios regulares de gimnasios. Al analizar los datos de la Figura 38, se observa que el 60% de los encuestados van al gimnasio entre 4 y 5 días a la semana, mientras que el 40% lo hacen entre 2 y 3 días.





Figura 39. Gráfico ¿Dónde apuntas tu progreso diario en los diferentes ejercicios de tus rutinas?

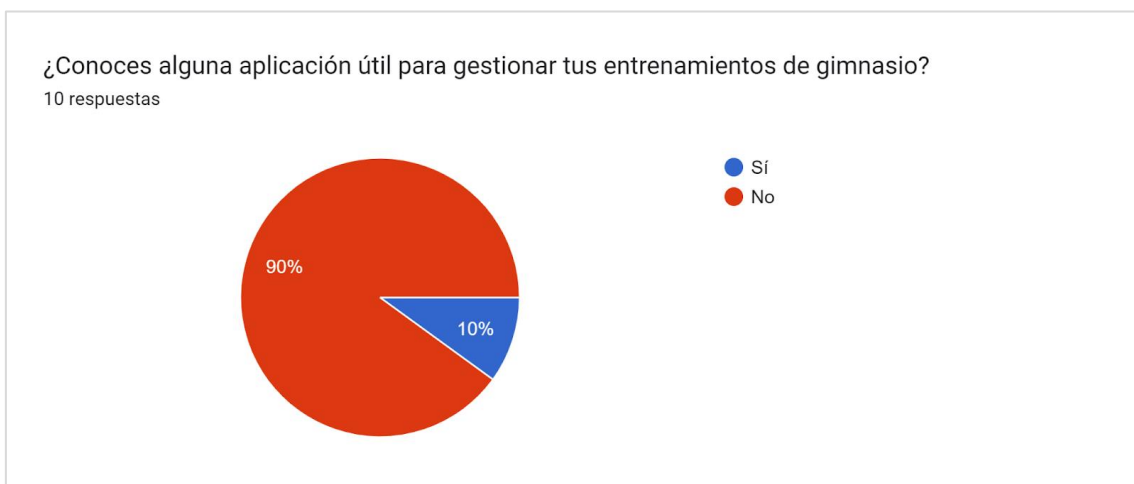


Figura 40. Gráfico ¿Conoces alguna aplicación útil para gestionar tus entrenamientos de gimnasio?

Tras analizar los resultados de estas preguntas, Figura 39 y 40, podemos afirmar que el 60% de los encuestados no apuntan su progreso, que un 20% lo apunta en las “notas del móvil” o similares y que solo un 20% utiliza aplicaciones de gestión de entrenamientos y que, además, solo 1 de los 10 encuestados conoce alguna aplicación útil para gestionar entrenamientos de gimnasio.

Como continuación a esta última pregunta ilustrada, existe otra pregunta en relación a esta y es que, en el caso de conocer alguna aplicación para la gestión de entrenamientos, les pedimos a los encuestados que nos dijeran cuál era, en este caso la única persona que conocía una aplicación contestó que la aplicación que conocía era “Hevy”, la cual es una de nuestros principales competidores como ya dijimos en el estudio de los competidores.

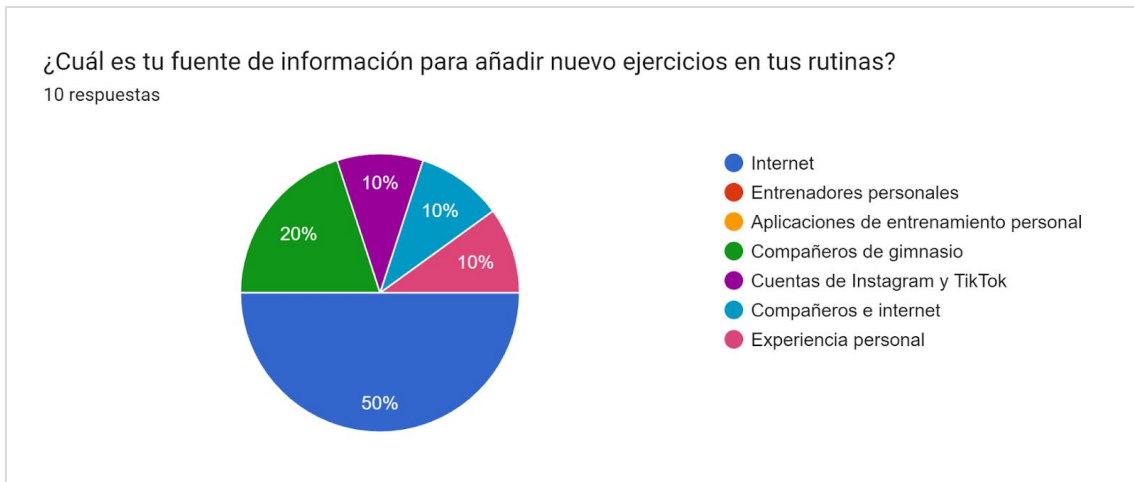


Figura 41. Gráfico ¿Cuál es tu fuente de información para añadir nuevos ejercicios en tus rutinas?

Como conclusión a esto resultados (Figura 41), podemos corroborar que un nuevo requisito de *Social Lift* es proporcionar ejercicios a los usuarios o que ellos mismos los puedan crear y compartir con otra gente.

Con respecto a la **Sección 2 a 7: Evaluación y Elicitación de requisitos de las IGUs.**

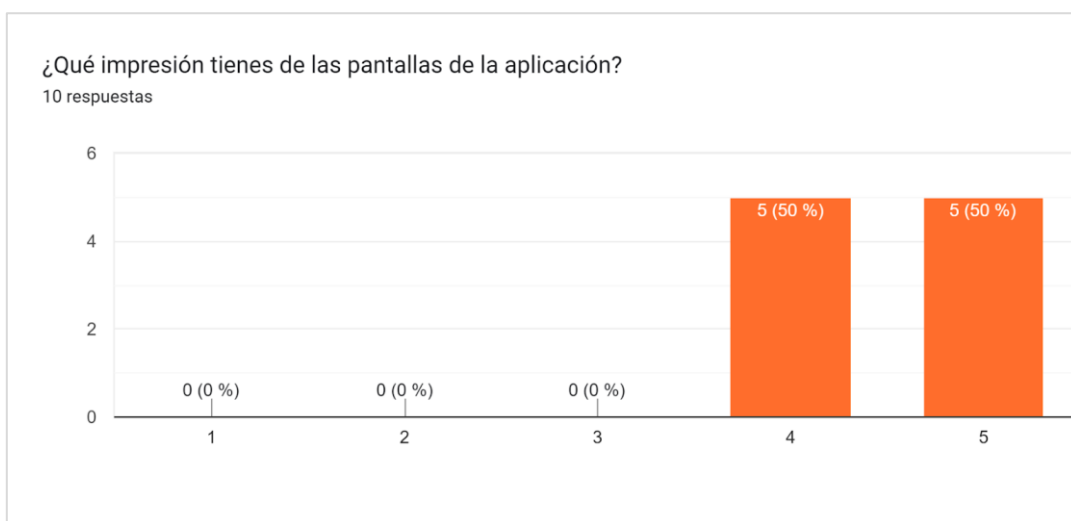


Figura 42. Gráfico ¿Qué impresión tienes de las pantallas de la aplicación?

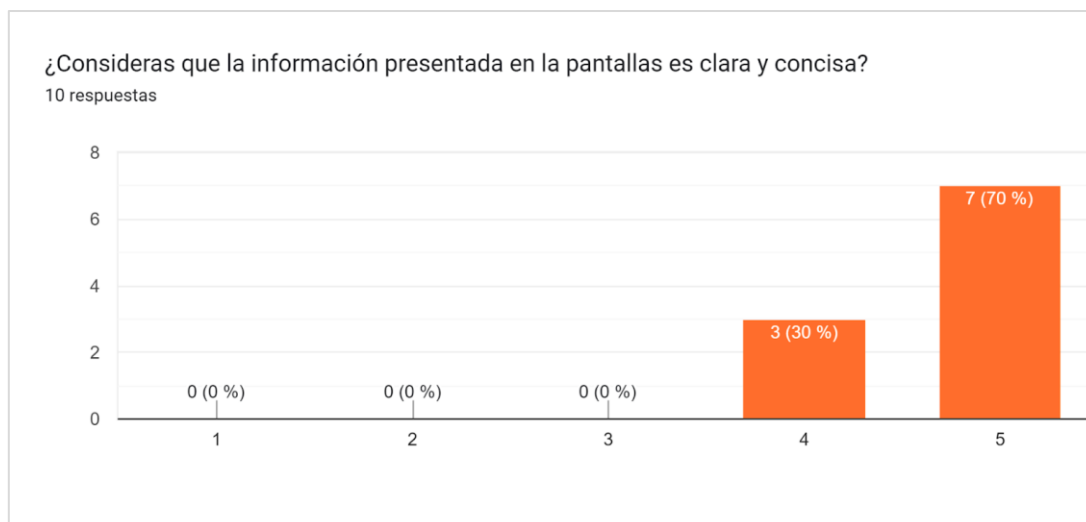


Figura 43. Gráfico ¿Consideras que la información presentada en las pantallas es clara y concisa?

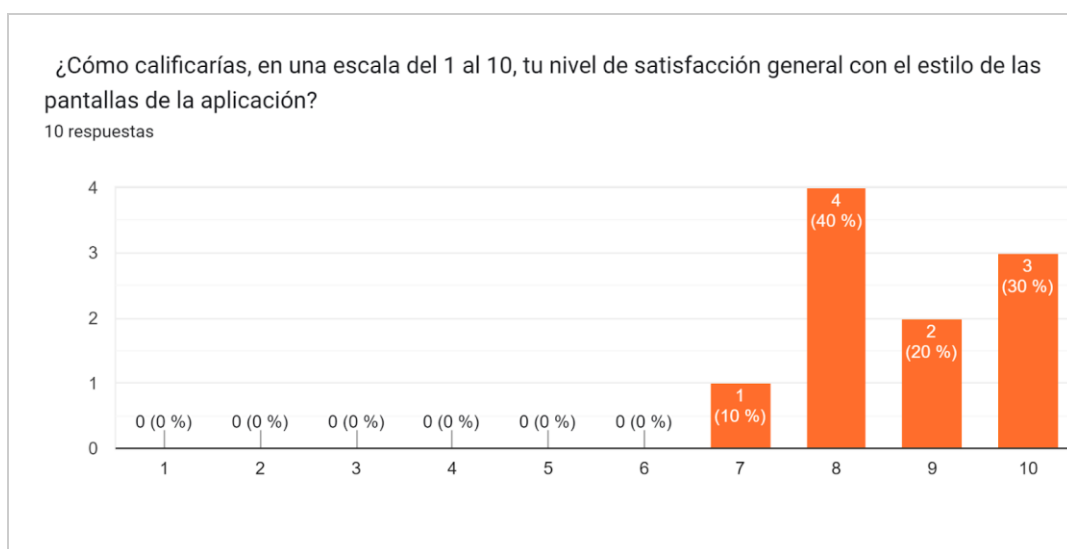


Figura 44. Gráfico ¿Cómo calificarías tu nivel de satisfacción general con el estilo de las pantallas de la aplicación?

Tras analizar los resultados de estas preguntas (Figura 42, 43 y 44) podemos decir que el diseño, la claridad y el entendimiento de esta pantalla son óptimos, debido a que el 100% de los encuestados califican el diseño de la aplicación con una nota mayor o igual a 7.

En relación a la última pregunta: ¿Qué funcionalidades te gustaría ver añadidas a esta pantalla en futuras versiones de la aplicación?

Tras analizar las respuestas, cabe destacar que algunas de ellas que son recurrentes:

- Que aparezca la duración de la rutina de entrenamiento.
- Que aparezcan videos en los ejercicios para que los demás usuarios los puedan ver.

- Poder publicar rutinas y que diferentes usuarios de la aplicación las puedan usar (Feed).
- Verificar a los usuarios.
- Poder añadir el gimnasio en el que haces tu rutina de entrenamiento y el tiempo
- Desplegable con el tipo de rutina

Como conclusión a esta pregunta se han elicitado estos nuevos requisitos para desarrollos futuros del producto.

Finalmente, tras el análisis de todos los resultados obtenidos en la encuesta, podemos decir que ha sido todo un éxito, tanto la validación de las IGUs como la elicitación de nuevos requisitos ha sido exitosa.

## 5.2 Desarrollo del segundo MVP

Durante el segundo tramo de desarrollo se llevó a cabo el segundo MVP, cuyo objetivo era implementar los nuevos requisitos que los *early adopters* nos definieron en el primer experimento y además todas aquellas tareas que estaban en nuestro backlog y entraban dentro del alcance de nuestro proyecto.

El objetivo principal, tras la finalización del desarrollo del segundo MVP era crear un MVP en el que los requisitos comunicados en el primer experimento estuvieran implementados, así como requisitos que hicieran de este MVP, y de la aplicación en general, una aplicación con una buena usabilidad para llevar a cabo el segundo experimento, así como una aplicación que se distinga de la competencia.

Este segundo tramo de desarrollo se ha llevado a cabo bajo dos *Sprints* (Sprint 3 y 4).

Con respecto al Sprint 3 se implementaron los siguientes requisitos (Figura 45):

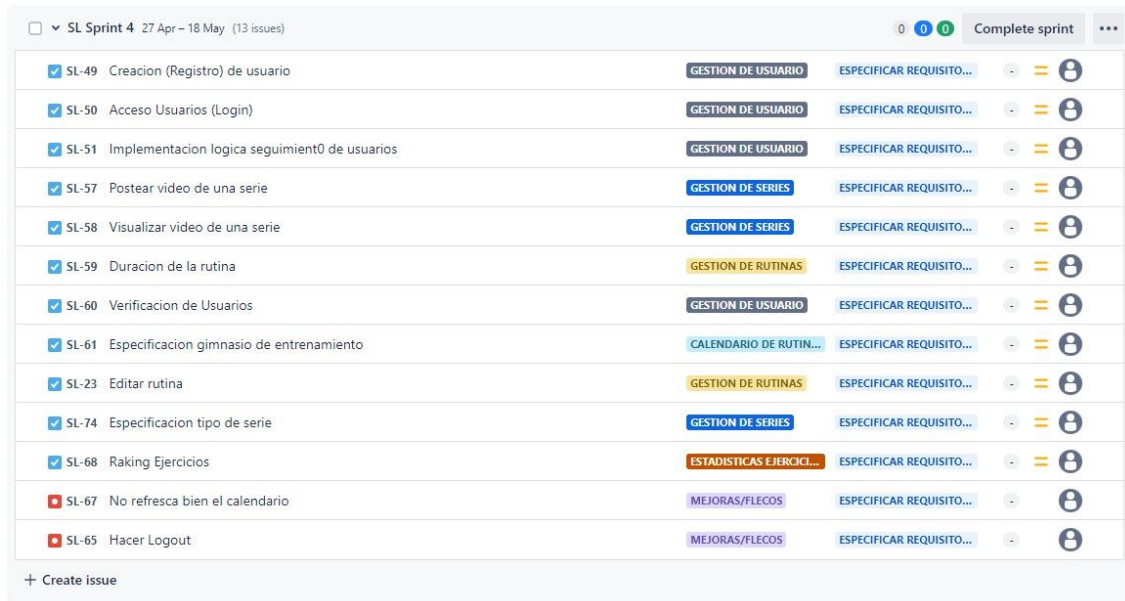
ID	Issue Title	Category	Status	Assignee
SL-31	Visualización de perfiles de usuario	PERFIL DE USUARIO	ESPECIFICAR REQUISITO...	JP
SL-35	Visualización rutinas guardadas	PERFIL DE USUARIO	ESPECIFICAR REQUISITO...	JP
SL-36	Visualización rutinas publicadas perfiles externos	PERFIL DE USUARIO	ESPECIFICAR REQUISITO...	JP
SL-25	Buscar rutina	GESTION DE RUTINAS	ESPECIFICAR REQUISITO...	JP
SL-28	Editar Ejercicio	GESTION DE EJERCICIOS	ESPECIFICAR REQUISITO...	JP
SL-24	Publicar rutina	GESTION DE RUTINAS	ESPECIFICAR REQUISITO...	JP
SL-48	Postear imagen de perfil	PERFIL DE USUARIO	ESPECIFICAR REQUISITO...	JP
SL-38	Seguimiento usuarios	PERFIL DE USUARIO	ESPECIFICAR REQUISITO...	JP
SL-42	Visualizar rutinas publicadas por usuario seguidos	FEED DE RUTINAS	ESPECIFICAR REQUISITO...	JP
SL-43	Guardar rutinas de otros usuarios	FEED DE RUTINAS	ESPECIFICAR REQUISITO...	JP
SL-44	Buscar rutinas	FEED DE RUTINAS	ESPECIFICAR REQUISITO...	JP
SL-45	Buscar Usuarios	FEED DE RUTINAS	ESPECIFICAR REQUISITO...	JP
SL-66	Cambiar iconos de los ejercicios	MEJORAS/FLECOS	ESPECIFICAR REQUISITO...	JP
SL-64	Añadir Desplegable en tipo rutina	MEJORAS/FLECOS	ESPECIFICAR REQUISITO...	JP

Figura 45. Sprint 3

Como se puede observar, las dos últimas tareas son bugs reportados tras las reuniones de retrospectiva de los *Sprints* anteriores.

## Una aplicación para la planificación y gestión de rutinas de gimnasio. Desarrollo del Frontend

Con respecto al Sprint 4, los requisitos implementados fueron los siguientes (Figura 46):



ID	Issue Title	Category	Action	Assignee
SL-49	Creacion (Registro) de usuario	GESTION DE USUARIO	ESPECIFICAR REQUISITO...	
SL-50	Acceso Usuarios (Login)	GESTION DE USUARIO	ESPECIFICAR REQUISITO...	
SL-51	Implementacion logica seguimiento de usuarios	GESTION DE USUARIO	ESPECIFICAR REQUISITO...	
SL-57	Postear video de una serie	GESTION DE SERIES	ESPECIFICAR REQUISITO...	
SL-58	Visualizar video de una serie	GESTION DE SERIES	ESPECIFICAR REQUISITO...	
SL-59	Duracion de la rutina	GESTION DE RUTINAS	ESPECIFICAR REQUISITO...	
SL-60	Verificacion de Usuarios	GESTION DE USUARIO	ESPECIFICAR REQUISITO...	
SL-61	Especificacion gimnasio de entrenamiento	CALENDARIO DE RUTIN...	ESPECIFICAR REQUISITO...	
SL-23	Editar rutina	GESTION DE RUTINAS	ESPECIFICAR REQUISITO...	
SL-74	Especificacion tipo de serie	GESTION DE SERIES	ESPECIFICAR REQUISITO...	
SL-68	Raking Ejercicios	ESTADISTICAS EJERCICL...	ESPECIFICAR REQUISITO...	
SL-67	No refresca bien el calendario	MEJORAS/FLECOS	ESPECIFICAR REQUISITO...	
SL-65	Hacer Logout	MEJORAS/FLECOS	ESPECIFICAR REQUISITO...	

Figura 46. Sprint 4

En este *Sprint* se realizaron tareas aportadas por los *early adopters* en el primer experimento, así como distintos bugs.

El cuarto *Sprint* marca la culminación del segundo MVP y, por ende, del marco de desarrollo del proyecto *Social Lift*, que es parte integral de este Trabajo de Fin de Grado. Tras la conclusión de este Sprint, se llevará a cabo un segundo experimento. Su propósito será validar los requisitos delineados por los *early adopters* en el primer experimento, así como verificar la funcionalidad del *frontend* de la aplicación.

Es esencial destacar que el término del desarrollo del MVP no indica el fin del proyecto, sino más bien un hito en su continuo progreso. El ciclo de vida de desarrollo de un producto de software es intrínsecamente fluido, sujeto a evolución constante y mejora continua.

### 5.2.1 Segundo Experimento

Tras terminar el segundo tramo de desarrollo, el cual dio lugar al segundo MVP, se procedió a la realización del segundo experimento, basado en este MVP.

Como ya se explicó, el primer experimento constaba con preguntas destinadas a la elicitación de nuevos requisitos, los cuales se implementan en este segundo MVP, en relación a esto la estructura del segundo experimento es la siguiente:

El experimento está dividido en dos secciones bien diferenciadas:

- **Sección 1: Validación de los nuevos requisitos.** En esta sección se llevó a cabo la validación de los requisitos recopilados en el primer experimento. La validación se realizó mediante una pregunta con las opciones de respuesta: Sí, No, Otro. De esta manera, se buscó confirmar la aceptación o rechazo de cada requisito por parte de los interesados. Los requisitos validados fueron los siguientes:
  - Aportación del tiempo en la rutina de entrenamiento.
  - Videos en las series de las rutinas
  - Feed de rutinas
  - Verificación de usuarios
  - Añadir gimnasio donde realizas la rutina de entrenamiento
- **Sección 2: DEMO.** En esta sección, se proporcionó a los encuestados acceso a la aplicación para que pudieran usarla y proporcionar su opinión. Durante esta fase, se recopilaron comentarios y respuestas sobre diversos aspectos de la aplicación. Las preguntas se dividieron en tres categorías: funcionalidades del *backend*, funcionalidades del *frontend* y preguntas generales sobre la idea de negocio. En nuestro caso, nos centraremos en las dos últimas categorías:
  - ¿Qué tan intuitiva encuentras la interfaz de usuario de la aplicación? Tipo de respuesta: Escala lineal del 1 al 5, siendo 1 poco intuitiva y 5 muy intuitiva
  - En una escala del 1 al 5, ¿cómo calificarías la facilidad de navegación dentro de la aplicación? Tipo de respuesta: Escala lineal del 1 al 5, siendo 1 algo complicada y 5 bastante fácil
  - ¿Te resulta fácil encontrar las funciones que necesitas dentro de la aplicación? Tipo de respuesta: Escala lineal del 1 al 5, siendo 1 algo complicada y 5 bastante fácil
  - ¿Cuál es la característica/funcionalides más útil que encuentras en la aplicación en términos de usabilidad? Tipo de respuesta: respuesta abierta
  - Que aspectos, tras el uso de la aplicacion, destacarías (tanto buenos como malos) y que crees que deberíamos tener en cuenta para futuras versiones de la aplicación. Tipo de respuesta: Respuesta abierta

## Una aplicación para la planificación y gestión de rutinas de gimnasio. Desarrollo del Frontend

- ¿Cómo evaluarías la variedad y utilidad de las funcionalidades ofrecidas por la aplicación en general? Tipo de respuesta: Escala numérica del 1 al 5, siendo 1 mala y 5 excelente.
- ¿Qué tan satisfecho/a estás con la diversidad de características y opciones que ofrece la aplicación para satisfacer tus necesidades? Tipo de respuesta: Escala numérica del 1 al 5, siendo 1 muy insatisfecho/a y 5 muy satisfecho/a.

Este experimento ha sido realizado por 20 personas, entre las cuales destacamos 10 de ellas, que fueron las encargadas de realizar el primer experimento, una de las intenciones por la que estas 10 personas debían realizar este experimento es para realizar la validación de los requisitos que ellos mismos definieron durante el primer experimento.

De entre las personas restantes podemos destacar la participación de 5 usuarios habituales de gimnasio y 5 usuario principiantes que se están adentrando en el mundo del *fitness*.

Tras la realización del experimento se procedió a realizar el siguiente análisis de los resultados obtenidos:

Con respecto a la **Sección 1**, podemos decir que los resultados obtenidos son bastante satisfactorios (Figura 47), casi todos los requisitos han tenido una validación satisfactoria.



Figura 47. Validación del requisito Feed de rutinas

Pero cabe mencionar el siguiente resultado (Figura 48) en la validación del requisito: *Verificación de los usuarios*.



Figura 48. Validación requisito *Verificación de los usuarios*

Un usuario destaca que *no es necesario, pero que aporta valor*, lo que quiere decir que no todos los usuarios ven indispensable este requisito.

Con respecto a la **Sección 2**, sacamos como conclusión que, en reglas generales, y haciendo un balance estadístico de todas las preguntas relacionadas con la experiencia, del usuario podemos afirmar que el 100 % de los usuarios están satisfechos con la experiencia de usuario tras la DEMO que han realizado, como muestran los siguientes resultados (Figura 49) de una de las preguntas relacionadas con ello:

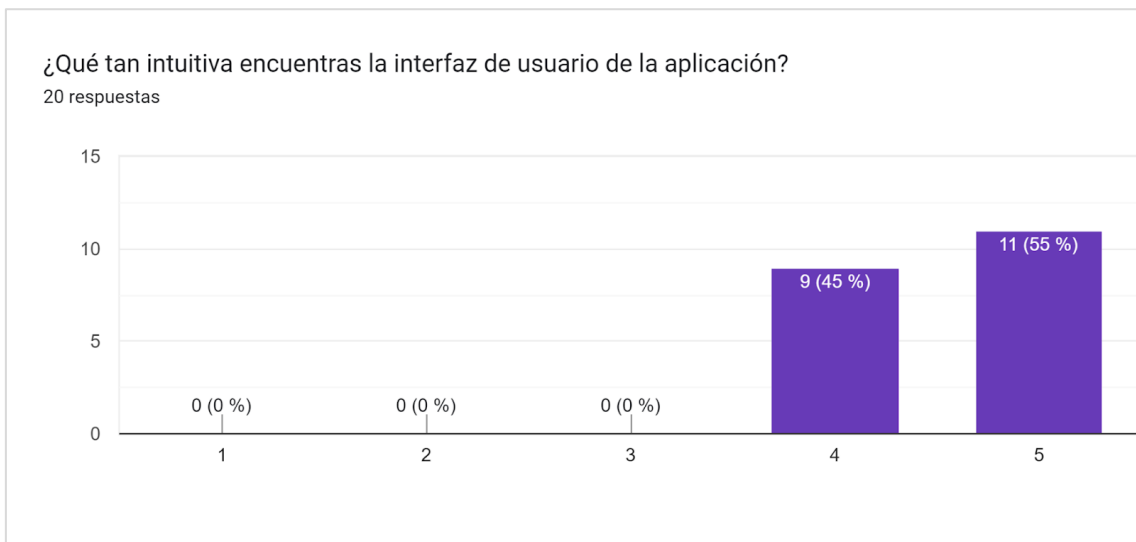


Figura 49. ¿Qué tan intuitiva encuentras la interfaz de usuario de la aplicación?



También debemos darles importancia a los siguientes resultados de la siguiente pregunta (Figura 50):

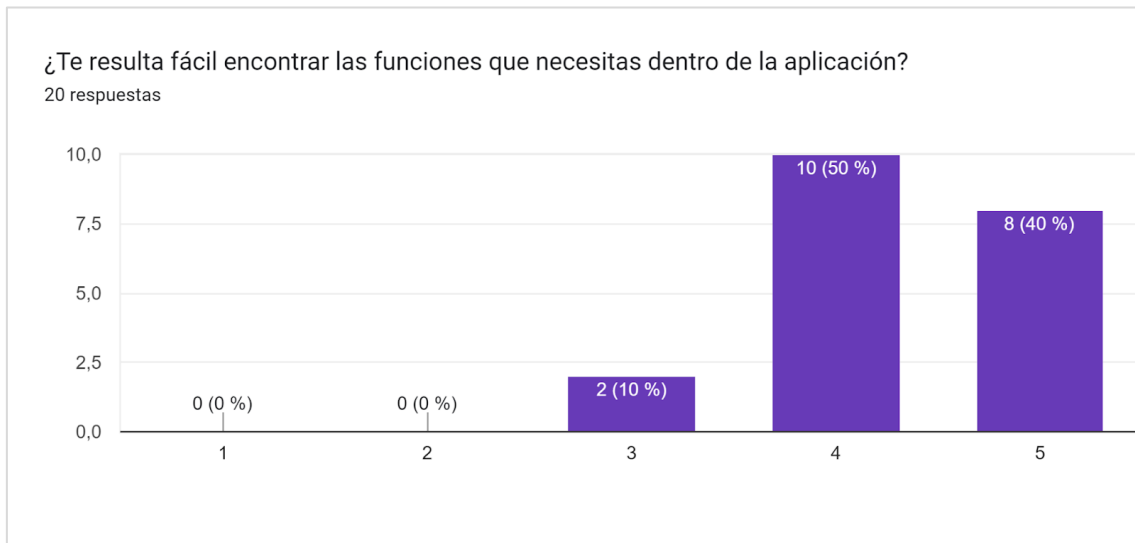


Figura 50. ¿Te resulta fácil encontrar las funciones que necesitas dentro de la aplicación?

Dado que el 10% de los encuestados no reportan buena facilidad de uso, hay dos encuestados a los que no les resulta fácil encontrar ciertas funcionalidades de la aplicación. Este era un riesgo que se corría, ya que, para evitar una interfaz de usuario recargada, se optó por colocar ciertas funcionalidades en sitios que puede que no fueran los más adecuados.

Además, el uso de ciertos iconos para las funcionalidades no ilustraba bien la funcionalidad en sí misma.

Esto, junto con lo mencionado anteriormente, corrobora que existe una parte de los encuestados que no encuentran *Social Lift* demasiado fácil de usar debido a la dificultad para encontrar ciertas funcionalidades.

La idea de este experimento también era que los usuarios nos ilustran sobre los aspectos que destacan de la aplicación o cuales son los que más o menos les han gustado, eso lo conseguimos mediante las dos siguientes preguntas:

- ¿Cuál es la característica/funcionalidad más útil que encuentras en la aplicación en términos de usabilidad?
- Que aspectos, tras el uso de la aplicación, destacarías (tanto buenos como malos) y que crees que deberíamos tener en cuenta para futuras versiones de la aplicación.

Debemos destacar las siguientes respuestas, las cuales han sido las más repetidas:

- Con respecto a la primera pregunta:
  - “El calendario y los videos de las series”
  - “El calendario, sin duda y, además el feed me parece muy útil para conectar con otros usuarios”
  - “Poder guardar rutinas de otras personas”
- Con respecto a la segunda pregunta:
  - “Me parece un acierto total el Feed de las rutinas que publican los usuarios que sigues, ya que provoca un "pique" con ellos y te hace motivarte en el gimnasio”
  - “Me gustaría que de cara a futuras versiones se pudiera realizar un ranking filtrando por gimnasio o ejercicio”

Como conclusión del experimento, se destaca que la validación exitosa de los requisitos confirma una implementación efectiva y fiel a la metodología ágil.

Además, el *feedback* recopilado a través de las respuestas relacionadas con las preguntas mencionadas anteriormente demuestra que nuestro producto ofrece características distintivas que lo diferencian de la competencia, como el *feed*, los videos de las series, etc. Asimismo, el buen *feedback* recibido en la evaluación de la usabilidad respalda el logro de uno de los principales objetivos de la aplicación: *ser intuitiva y fácil de usar*.

### 5.2.2 Dedicación

Para comentar la dedicación aportada a este proyecto debemos tener en cuenta varias actividades que se encuadran dentro del mismo, el tiempo invertido se ilustrara en horas hombre y en la Tabla 8 y 9:

Actividad	Marco de Trabajo	Fecha Inicio	Fecha Fin	Tiempo Invertido
<b>Formación e investigación</b>	-	22/01/2024	04/02/2024	28 h
<b>Evaluación de la idea de negocio *</b>	Sprint 0	04/02/2024	17/02/2024	26 h
<b>Desarrollo Frontend del primer MVP</b>	Sprint 1	19/02/2024	09/03/2024	49 h
	Sprint 2	11/03/2024	01/04/2024	49 h
<b>Realización y análisis de Experimentos *</b>	Experimento 1	02/04/2024	05/04/2024	6 h
<b>Desarrollo Frontend del segundo MVP</b>	Sprint 3	05/04/2024	26/04/2024	49 h
	Sprint 4	27/04/2024	18/05/2024	49 h
<b>Realización y análisis de Experimentos *</b>	Experimento 2	19/05/2024	21/05/2024	6h
<b>Redacción de la memoria</b>	-	-	-	95 h
<b>TOTAL</b>				<b>357 h</b>

Tabla 8. Tabla dedicación horas (Desglose) del proyecto

Actividad	Tiempo Invertido
<b>Formación e investigación</b>	28 h
<b>Evaluación de la idea de negocio</b>	26 h
<b>Desarrollo del producto y experimentos</b>	208 h
<b>Redacción de la memoria</b>	95
<b>TOTAL</b>	<b>357 h</b>

Tabla 9. Tabla dedicación horas (Genérico) del proyecto

\* Actividades realizadas conjuntamente junto a mi compañero Juan Pablo Bueno.

# 6. Conclusiones y trabajo futuro

---

## 6.1 Conclusiones

*Social Lift* surgió como una respuesta directa a un problema experimentado en un gimnasio, dando origen a un proyecto emprendedor que se reflejó en dos Trabajos de Fin de Grado (TFG): uno enfocado en el desarrollo del *frontend* de la aplicación, realizado por mí, y otro en el *backend*, llevado a cabo por mi compañero Pablo Bueno. Este proceso de desarrollo ha sido una experiencia enriquecedora, marcada por un constante aprendizaje.

En términos de objetivos, el TFG ha logrado con éxito cumplir cada uno de los objetivos planteados inicialmente. Hemos evaluado la viabilidad financiera del proyecto, asegurando su sostenibilidad a largo plazo. Hemos creado una aplicación con una usabilidad sencilla y una interfaz llamativa, diferenciándola de la competencia y mejorando la experiencia del usuario. Además, hemos implementado funcionalidades específicas que permiten monitorear el progreso de los usuarios en sus rutinas de ejercicios.

Nos hemos enfocado en construir una red social dentro de la aplicación, permitiendo a los usuarios compartir sus rutinas y logros en un feed, fomentando así una comunidad fitness estrecha y solidaria. La gestión del proyecto se ha realizado de manera clara y ágil, utilizando aplicaciones de gestión de proyectos para mantenernos organizados y eficientes. Finalmente, hemos validado el producto en un entorno real, llevando a cabo dos experimentos con *early adopters*, cuyos comentarios han sido invaluable para mejorar la aplicación y aplicar técnicas de ingeniería del software, como la ingeniería de requisitos y la validación de la interfaz gráfica de usuario (IGU).

*Social Lift* ha cumplido con éxito todos los objetivos establecidos, resultando en una aplicación funcional y atractiva que no solo facilita y motiva a la comunidad fitness global, sino que también ofrece una experiencia de usuario superior y una gestión eficiente del progreso en los entrenamientos.

El estado actual de la aplicación refleja un avance significativo desde su concepción inicial, pero aún está en una fase de desarrollo que requiere refinamientos y mejoras adicionales. Hasta ahora, hemos desarrollado dos MVPs los cuales nos han permitido validar las funcionalidades básicas y obtener valiosos comentarios de los usuarios tempranos. Estos MVPs han sido fundamentales para demostrar la viabilidad del concepto y han proporcionado una base sólida sobre la cual construir. Sin embargo, reconocemos que aún estamos en una etapa de desarrollo. Estamos dedicados a continuar iterando sobre la base actual, incorporando *feedback* de los usuarios y perfeccionando la aplicación para que pueda ofrecer el máximo valor a la comunidad fitness global.

Personalmente, este proyecto ha sido una base sólida para futuros emprendimientos, brindándome la oportunidad de gestionar un proyecto desde sus inicios, ampliando mis conocimientos en desarrollo de nuevos lenguajes y *frameworks*, y fortaleciendo mis *soft skills*, como la comunicación efectiva y el trabajo en equipo.

Para la realización de este proyecto, me he basado en los conocimientos adquiridos a lo largo de mi carrera universitaria, especialmente en asignaturas como:

- Proyecto de Ingeniería de Software (PIN).
- Proceso Software (PSW).
- Desarrollo de Software (DDS).
- Análisis y Especificación de Requisitos (AER).

En resumen, el desarrollo de *Social Lift* ha sido un desafío exigente pero gratificante, que me ha ayudado a mejorar mi resiliencia personal y mi capacidad resolutiva, tanto en el ámbito profesional como personal.

## 6.2 Trabajo Futuro

En cuanto al futuro del proyecto, se espera un continuo desarrollo y consolidación en el mercado. Se planea seguir mejorando la aplicación, basándose en el *feedback* de los usuarios para introducir cambios que mejoren su experiencia y añadir nuevas funciones relevantes.

Además, se busca aumentar la base de usuarios mediante estrategias de promoción y marketing. Esto se complementará con el desarrollo de características innovadoras que aporten un valor diferencial y distingan a la aplicación en un mercado competitivo.

Asimismo, se pretende optimizar el rendimiento de la aplicación para garantizar una experiencia de usuario fluida. Se considera explorar colaboraciones estratégicas con entidades relacionadas con el sector del *fitness* y el bienestar para ampliar la oferta de servicios de *Social Lift*.

En resumen, el futuro del proyecto se centra en seguir evolucionando para satisfacer las necesidades cambiantes de los usuarios y mantener su competitividad en el sector del desarrollo de aplicaciones para el bienestar y el *fitness*.

# 7. Referencias

---

- [1] Informe global de la IHRSA 2022. Disponible en: <https://es.ihrsa.org/publications/the-2022-ihrsa-global-report/> . Consultada el 01/2024.
- [2] Disponible en: <https://ifbbSpain.com/> . Consultada el 01/2024
- [3] Acerca de Strava. Disponible en: <https://es.wikipedia.org/wiki/Strava> . Consultada el 02/2024.
- [4] Sobre nosotros Heavy: <https://www.hevyapp.com/about-us/> . Consultada el 02/2024.
- [5] Análisis DAFO. Disponible en: [https://es.wikipedia.org/wiki/An%C3%A1lisis\\_FODA](https://es.wikipedia.org/wiki/An%C3%A1lisis_FODA) . Consultada el 02/2024.
- [6] Acerca de Flutter. Disponible en: [https://es.wikipedia.org/wiki/Flutter\\_\(software\)](https://es.wikipedia.org/wiki/Flutter_(software)) . Consultada el 03/2024.
- [7] Lean Startup: [https://es.wikipedia.org/wiki/Lean\\_startup](https://es.wikipedia.org/wiki/Lean_startup) . Consultada el 03/2024.
- [8] MVP: [https://es.wikipedia.org/wiki/Producto\\_viable\\_m%C3%ADnimo](https://es.wikipedia.org/wiki/Producto_viable_m%C3%ADnimo) . Consultada el 03/2024.
- [9] Scrum: <https://www.atlassian.com/es/agile/scrum> . Consultada el 03/2024

# ANEXOS

## ANEXO A: GUIA DE USO

### 1. Feed de Rutinas

Desde esta ventana de la aplicación, en la parte de **Publicaciones**, el usuario podrá visualizar y/o guardar las rutinas publicadas por los usuarios que sigue.

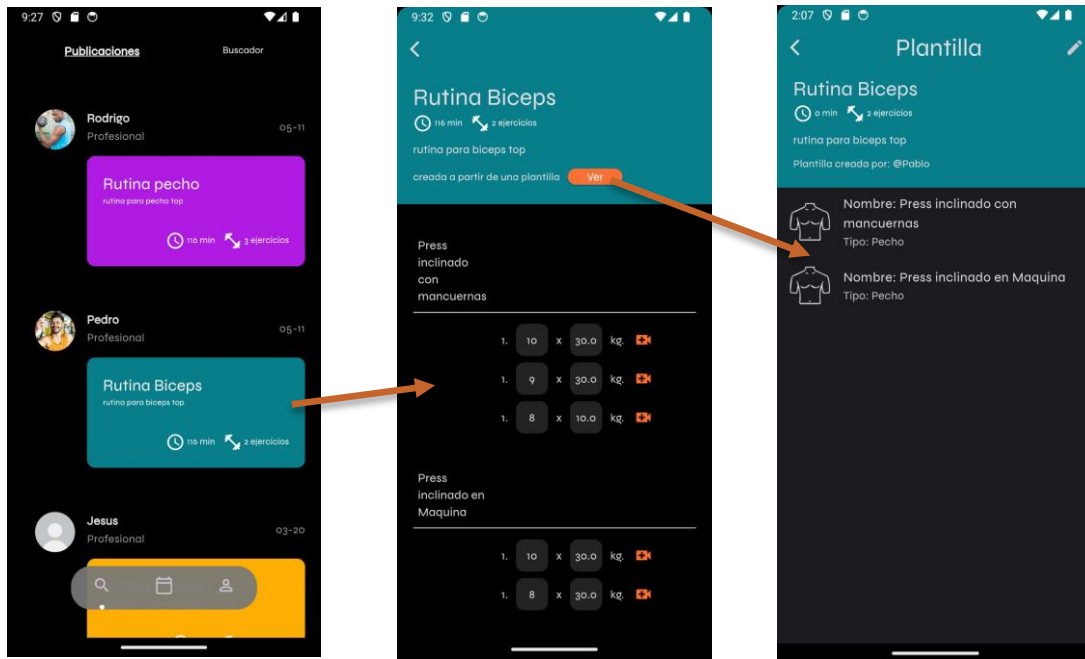


Figura 51. Navegar por las rutinas

Cabe destacar que el usuario al añadir una serie a un ejercicio de la rutina, podrá subir un video realizando la serie (Figura 51).

Dentro del Feed en el apartado de **Buscador** (Figura 52) podrás buscar tanto **usuario** como **plantillas de rutinas** que quieras realizar en tus entrenamientos.

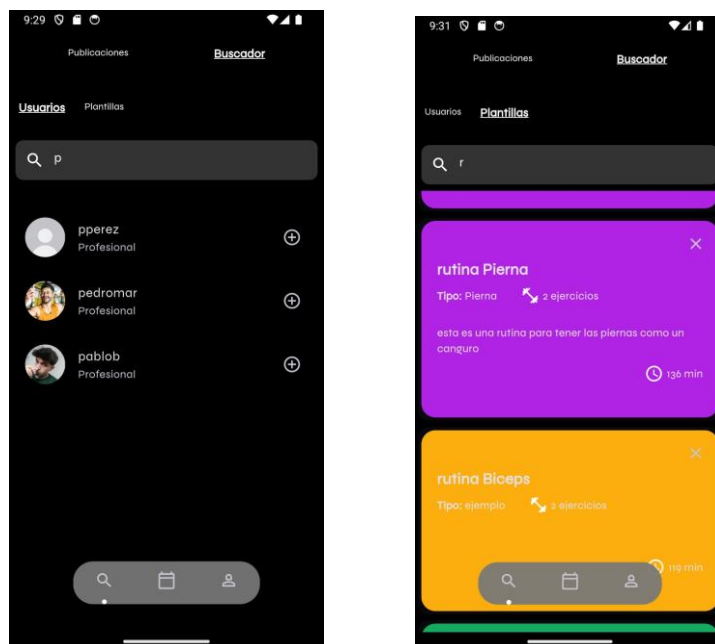


Figura 52. Buscador

Desde estas ventanas, se podrá tanto navegar hacia el perfil del usuario como navegar hacia la rutina que quieras, visualizar su información y usarla o guardarla para futuros entrenamientos.



## 2. Calendario de Rutinas

En esta sección se muestra un calendario mensual de las rutinas que ha realizado o realizará el usuario en un mes, ilustrado en la Figura 53, estas se muestran mediante un punto de color en la parte inferior del y al pulsar el día, se podrán visualizar.

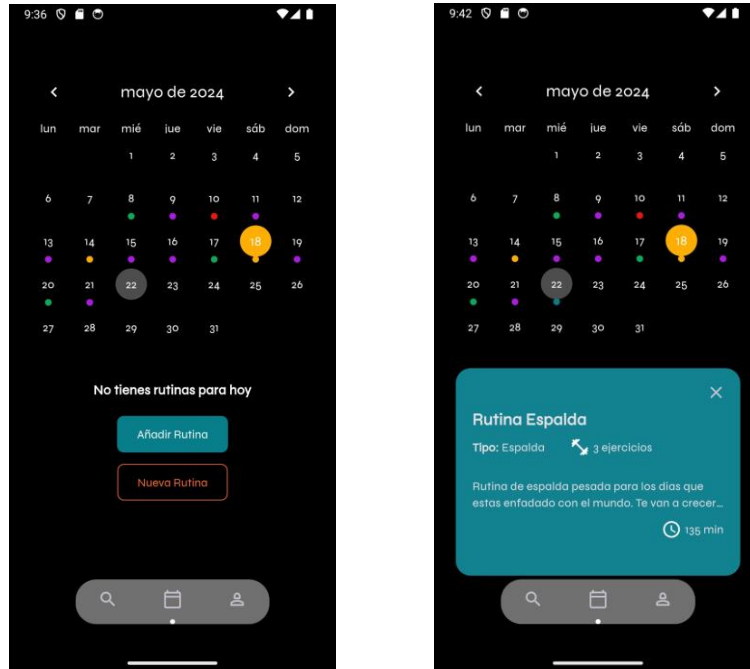


Figura 53. Pantalla calendario

En el caso de la imagen de la derecha se observa que te ofrece información acerca de la rutina que has realizado ese día. Como se puede observar en la imagen de la izquierda, puedes **Añadir** una rutina, donde te aparecerán tus rutinas guardadas (Figura 54):



Figura 54. Rutinas

O también podrás **Crear** tu propia rutina personalizada (Figura 55):

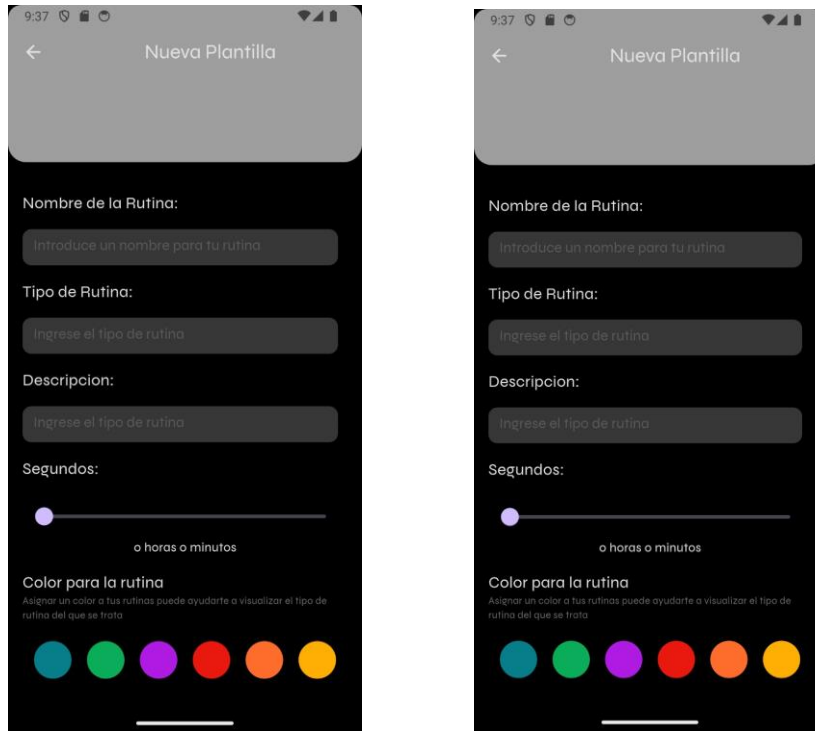


Figura 55. Nueva rutina

En el apartado de **Ejercicios**, le podrás añadir los ejercicios que tú quieras (Figura 56):

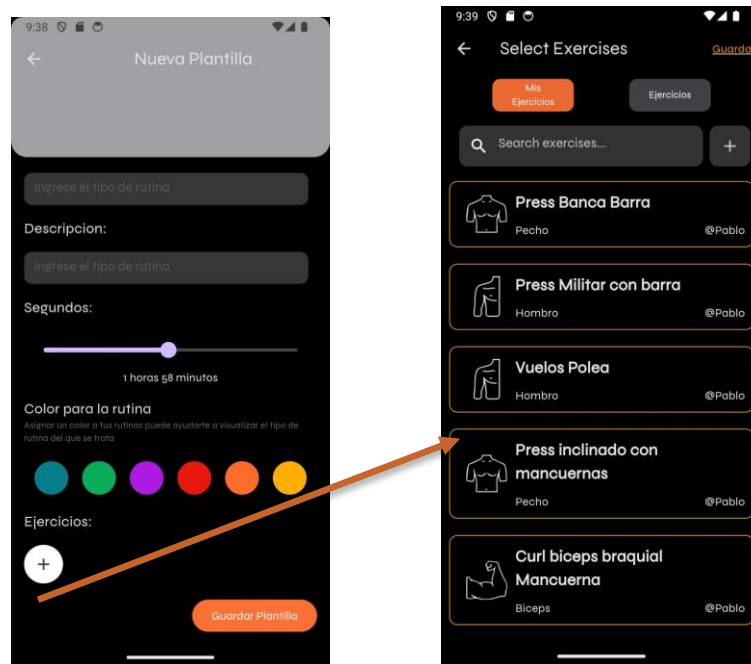


Figura 56. Nueva rutina y añadir ejercicios

Al crear la rutina tras añadir los ejercicios, quedará de la siguiente manera (Figura 57):



Figura 57. Rutina Detalles

Pulsando el botón de + podrás añadir tus series.

### 3. Perfil de Usuario

Esta ventana (Figura 58) muestra información común del usuario. Dentro de esta hay tres subsecciones **Datos morfológicos**, **Datos ejercicios** y **Plantillas** la cual mostraremos más adelante.

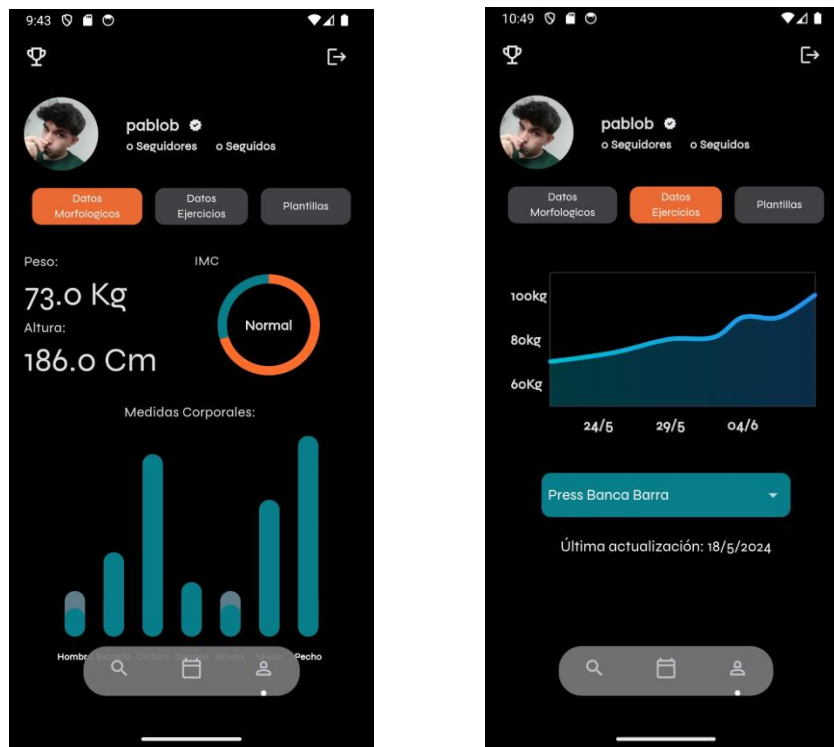


Figura 58. Perfil de Usuario- Datos Morfológicos/Datos Ejercicios

Dentro de Datos Ejercicios se puede observar una gráfica con tu progreso de levantamientos por cada ejercicio que has realizado y durante el tiempo.

## Una aplicación para la planificación y gestión de rutinas de gimnasio. Desarrollo del Frontend

Dentro del apartado de **Plantillas** (Figura 59) el usuario podrá visualizar tanto sus rutinas guardadas, las cuales ha guardado de otros usuarios o sus propias rutinas creadas.

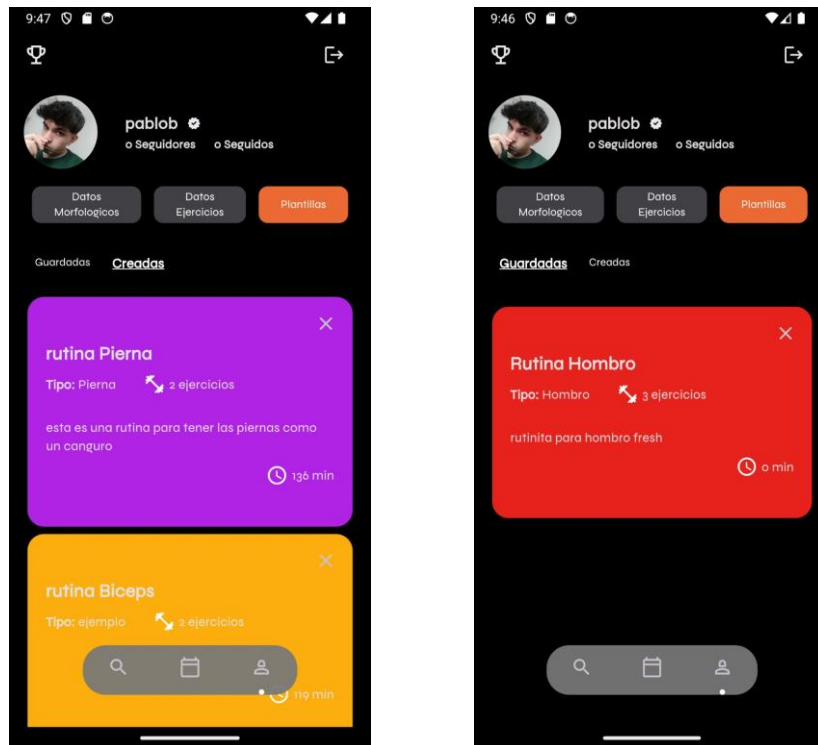


Figura 59. Perfil Usuario - Plantillas

#### 4. Ranking

Esta sección de la aplicación, muestra un **ranking** (Figura 60) estadístico del número de entrenamientos realizados por los usuarios de la aplicación.

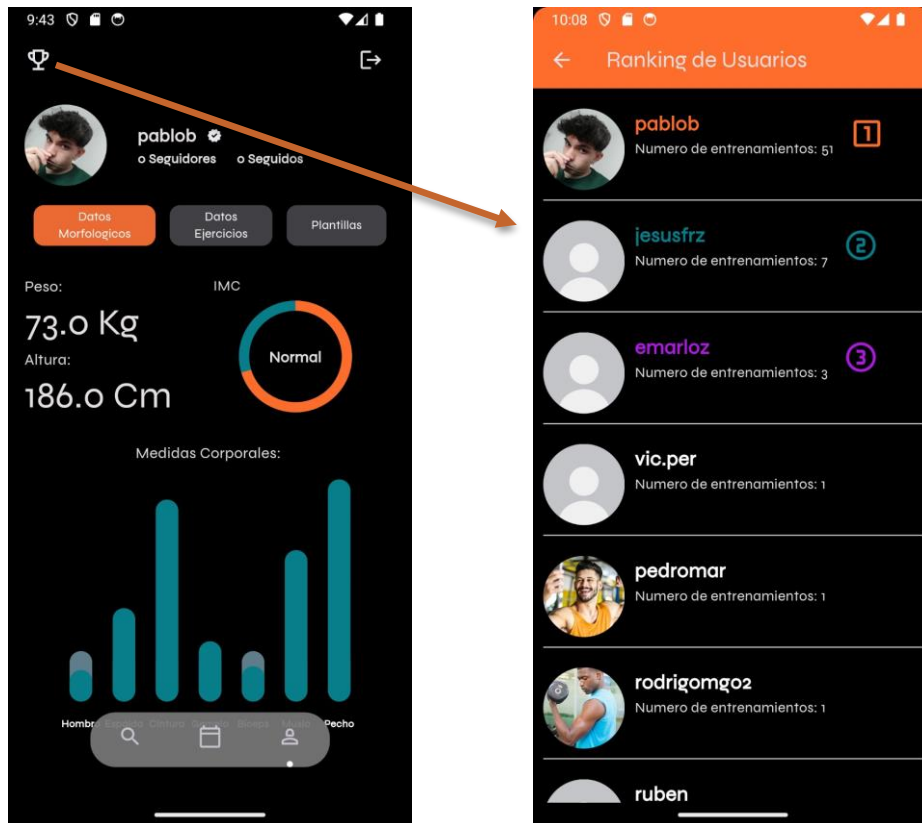


Figura 60. Ranking

## ANEXO B: OBJETIVOS DE DESARROLLO SOSTENIBLE

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

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

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

El Trabajo de Fin de Grado (TFG) en el desarrollo de la aplicación *Social Lift* presenta una oportunidad única para analizar y reflexionar sobre su contribución a los Objetivos de Desarrollo Sostenible (ODS) propuestos por la Organización de las Naciones Unidas (ONU). *Social Lift* es una innovadora aplicación móvil multiplataforma diseñada para facilitar y mejorar la gestión de rutinas de ejercicio y promover la colaboración dentro de la comunidad fitness. Esta reflexión examina en detalle cómo *Social Lift* se relaciona con varios ODS, identificando áreas de alto impacto, contribuciones indirectas y aquellos objetivos con los que la relación es menos evidente.

### **ODS 3: Salud y Bienestar**

La relación más directa y significativa de *Social Lift* es con el ODS 3, que se centra en garantizar una vida sana y promover el bienestar para todos en todas las edades. *Social Lift* promueve la salud y el bienestar de varias maneras:

- **Facilitación de la actividad física:** Proporciona una plataforma donde los usuarios pueden registrar sus entrenamientos, acceder a una amplia variedad de ejercicios y rutinas, y seguir su progreso. Esto fomenta hábitos saludables y una vida activa, elementos cruciales para el bienestar físico.
- **Educación y conocimiento:** A través del *feed* integrado, los usuarios pueden compartir y aprender de las experiencias y rutinas de otros, lo que incrementa el conocimiento sobre prácticas de ejercicio eficaces y seguras.
- **Feedback y mejora continua:** La posibilidad de subir videos y recibir retroalimentación de usuarios verificados ayuda a mejorar la técnica y prevenir lesiones, promoviendo una práctica de ejercicio más segura y efectiva.

La promoción de la salud física y mental es fundamental para el bienestar general, y *Social Lift* proporciona herramientas y recursos que facilitan esta promoción de manera accesible y personalizada.

### **ODS 4: Educación de calidad**

Aunque no es una plataforma educativa en el sentido tradicional, *Social Lift* contribuye al ODS 4 al proporcionar un espacio para la educación informal y el intercambio de conocimientos en el ámbito del fitness.

Este tipo de educación no formal es esencial para la construcción de conocimientos prácticos y aplicables que mejoren la calidad de vida de los individuos.

### **ODS 5: Igualdad de género**

*Social Lift* promueve la igualdad de género al ser una plataforma inclusiva donde todos los usuarios, independientemente de su género, tienen acceso equitativo a recursos de fitness y bienestar.

Fomentar un entorno inclusivo y equitativo contribuye a la reducción de las desigualdades de género en el acceso a información y oportunidades de bienestar.

### **ODS 9: Industria, innovación e infraestructuras**

La innovación tecnológica de *Social Lift* en el sector de la salud y el fitness se alinea con el ODS 9, que promueve la construcción de infraestructuras resilientes, la industrialización inclusiva y sostenible, y la innovación.

Estas contribuciones son fundamentales para el desarrollo sostenible de infraestructuras que apoyen el bienestar y la salud pública.



### **ODS 10: Reducción de las desigualdades**

*Social Lift* puede ayudar a reducir desigualdades al proporcionar acceso equitativo a recursos de salud y bienestar, independientemente de la ubicación geográfica o el nivel socioeconómico.

Este enfoque contribuye a la creación de una sociedad más equitativa y justa, donde todos tienen la oportunidad de mejorar su salud y bienestar.

### **ODS 17: Alianzas para lograr objetivos**

Finalmente, *Social Lift* puede facilitar alianzas y colaboraciones dentro de la comunidad fitness, promoviendo un esfuerzo colectivo para mejorar la salud y el bienestar.

Estas alianzas son cruciales para maximizar el impacto de la aplicación y contribuir al logro de objetivos de desarrollo sostenible.

En resumen, *Social Lift* tiene una relación directa y significativa con varios ODS, especialmente con el ODS 3 (Salud y bienestar), el ODS 4 (Educación de calidad), el ODS 5 (Igualdad de género), el ODS 9 (Industria, innovación e infraestructuras), el ODS 10 (Reducción de las desigualdades) y el ODS 17 (Alianzas para lograr objetivos). La aplicación no solo promueve la salud y el bienestar, sino que también facilita la educación, la igualdad, la innovación y la reducción de desigualdades a través de su plataforma inclusiva y colaborativa. Aunque algunas ODS tienen una relación menos evidente con *Social Lift*, es importante reconocer el potencial de la tecnología para contribuir al desarrollo sostenible en múltiples dimensiones. Esta reflexión subraya la importancia de integrar consideraciones de sostenibilidad en el diseño y desarrollo de aplicaciones tecnológicas para maximizar su impacto positivo en la sociedad.