UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

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

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Bueno Fuentes, Juan Pablo

Tutor/a: Letelier Torres, Patricio Orlando

CURSO ACADÉMICO: 2023/2024

Para mis padres por el apoyo constante en todo mi proceso del grado universitario.

Agradecimientos

Quiero agradecer a todas las personas que hicieron posible la realización de este TFG. En primer lugar, a mi compañero Jesús Fiérrez Ruipérez, por su dedicación en el desarrollo del *Frontend*, su contribución fue fundamental para iniciar el proyecto de emprendimiento.

Agradezco también a tres compañeras del grado de Bellas Artes, Andrea Palomo Gordón, Joanna López Cremades y Noemí Ortega Rodríguez, que contribuyeron con su talento para realizar el diseño de la aplicación.

Finalmente, a mi tutor Patricio Letelier, por su orientación y apoyo a lo largo de este proceso.

Resumen

Cada día son más los usuarios que se inscriben a gimnasios buscando un estilo de vida sano y un buen estado físico. Si se busca conseguir el mejor rendimiento, es necesario tener un registro del progreso en los entrenamientos y rutinas, por lo que, tener una herramienta que faciliten esta gestión resulta primordial. El mercado de las aplicaciones móviles está experimentando un crecimiento exponencial, ofreciendo nuevas soluciones de gestión en áreas como la salud y el *fitness*, Con este concepto en mente, este TFG describe la puesta en marcha de un proyecto de emprendimiento sobre el desarrollo del *Backend* de una aplicación móvil que gestiona ejercicios y rutinas de entrenamientos de gimnasio.

El proyecto tiene como nombre *Social Lift* y su objetivo es desarrollar una aplicación para dispositivos móviles nativa, que proporcione a los usuarios soporte para la gestión de rutinas relacionadas con el entrenamiento físico, ayude a los usuarios a tener seguimiento de su progreso y genere una comunidad sólida para compartir información entre los usuarios. Para ello se utilizará la metodología *Lean Startup,* incluyendo el desarrollo de al menos dos MVP y la puesta en marcha de experimentos con *early adopters*.

Este proyecto de emprendimiento se enmarca en Start.inf, el espacio de emprendimiento de la ETSINF. Participaron dos personas del grado de ingeniería informática con la ayuda creativa de alumnas de la facultad de BBAA.

Palabras clave: Desarrollo de software, metodologías agiles, Lean Startup, Scrum, emprendimiento.

Abstract

Every day more users are enrolling in gyms seeking a healthy lifestyle and good physical condition. To achieve the best performance, it is necessary to keep track of progress in workouts and routines, so having a tool that facilitates this management is essential. The mobile application market is experiencing exponential growth, offering new management solutions in areas such as health and fitness. With this concept in mind, this Final Degree Project describes the launch of an entrepreneurial project focused on developing the Backend of a mobile application that manages gym exercises and workout routines.

The project is named *Social Lift*, and its objective is to develop a native mobile application that provides users with support for managing workout routines, helps users track their progress, and builds a strong community for sharing information among users. To achieve this, the Lean Startup methodology will be used, including the development of at least two MVP and experiments with early adopters.

This entrepreneurial project is part of Start.inf, the entrepreneurship space of ETSINF. Two computer engineering students participated with the creative help of students from the Faculty of Fine Arts.

Keywords: Software development, agile methodologies, Lean Startup, Scrum, entrepreneurship.

Tabla de contenido

índice de figuras

Índice de tablas

1.Introducción

1.1. Motivación

Cada día, más personas se apasionan por el mundo del estado físico (*fitness*), contribuyendo al aumento significativo de individuos que frecuentan gimnasios en la actualidad. Según datos de la IHRSA[1] (International Health Racquet & Sportsclub Association) en 2021, más de 66 millones de personas en Estados Unidos eran miembros activos de instalaciones de *fitness* [1], cifra que continúa en aumento. Según datos de Statista[2], en 2022, aproximadamente 5.4 millones de personas eran miembros de gimnasios y centros de fitness en España [2]. Este crecimiento no solo refleja una tendencia hacia un estilo de vida más saludable, sino que también una mayor conciencia sobre la importancia del bienestar físico en la sociedad.

Asimismo, El sector de centros de entrenamiento ha experimentado una transformación. Según el informe económico del Fitness 2024 [3], las principales cadenas registraron un aumento del 39,4% en la facturación. Este fenómeno puede atribuirse a varios posibles factores, incluyendo la influencia de las redes sociales, así como una mayor accesibilidad a la información sobre salud y bienestar por medio de internet.

Con el progreso tecnológico, el mundo del *fitness* podría verse afectado. En la actualidad, muchos usuarios buscan herramientas que les permitan llevar un seguimiento detallado del progreso y mejora de rendimiento de los ejercicios y sus rutinas de gimnasio. Las aplicaciones móviles son las encargadas de ofrecer soluciones innovadoras que facilitan el monitoreo de entrenamientos. Sin embargo, aún existe una carencia de soluciones integrales que combinen la gestión de entrenamientos con la posibilidad de intercambio de conocimientos entre usuarios.

Esta necesidad ha impulsado la idea de desarrollar un software que facilite el control de los entrenamientos y fomente una comunidad de *fitness* sólida y saludable. Buscando integrar funcionalidades que respondan a las demandas actuales del mercado. Por estos motivos nace el proyecto *Social Lift*, una aplicación móvil multiplataforma con herramientas funcionales e innovadoras para gestionar rutinas de entrenamiento, hacer seguimiento de los ejercicios y promover un estilo de vida saludable en la comunidad *fitness*.

[1] https://www.healthandfitness.org/

[2] https://es.statista.com/

1.2. Objetivos

El objetivo del proyecto es desarrollar una aplicación de móvil nativa denominada *Social Lift* que proporcione a los usuarios soporte para la gestión de rutinas relacionadas con el mundo del *fitness*, ayude a los usuarios a monitorear su progreso y genere una comunidad sólida para compartir información entre los usuarios. Este TFG se centra en el *Backend* de la aplicación, en el desarrollo de una API REST para ofrecer el servicio de gestión de datos a la aplicación de *Social Lift*.

Para conseguir este objetivo, se han establecido las siguientes tareas:

- Desarrollar una API REST funcional para la aplicación móvil *Social Lift*.
- Diseñar la API para manejar un alto número de solicitudes concurrentes.
- Desarrollar un producto disponible en todo momento para que los usuarios potenciales (*early adopters)* puedan utilizar la aplicación desde cualquier lugar si tienen acceso a internet.
- Evaluar y validar el producto en un entorno real y conseguir un impacto positivo en la comunidad *fitness*.
- Conseguir una buena gestión de rutinas, ejercicios, series y usuarios en la aplicación sin errores en los datos.

Por otro lado, hay objetivos personales que no son necesarios para el desarrollo del proyecto, pero que contribuyen al crecimiento profesional:

- Aumentar mi experiencia tanto en la gestión y planificación de proyectos como en el desarrollo de un producto *Backend* con las tecnologías relevantes de la actualidad.
- Adquirir conocimiento en nuevas áreas de programación y nuevas tecnologías
- Mejorar la práctica de metodologías útiles para la puesta en marcha de un proyecto.

1.2. Estructura de la memoria

La memoria se ha estructurado con una organización de cinco capítulos:

- **Primer capítulo:** Como ya se ha visto, introduce toda la información relativa a la motivación, objetivos del proyecto de desarrollo, describe la estructura de la memoria y finalmente habla sobre la colaboración obtenida para la puesta en marcha de este proyecto.
- **Segundo capítulo:** Describe como está consolidada la idea de negocio, mostrando los distintos clientes de la aplicación, distintas herramientas de planificación estratégicas y un análisis de mercado y financiero del proyecto.
- T**ercer capítulo:** Relata las distintas tecnologías utilizadas para la puesta en marcha del *Backend* de *Social Lift* y el porqué de su elección.
- **Cuarto capítulo:** Muestra todo lo relacionado con aspectos técnicos del proyecto y su desarrollo, describiendo la metodología usada, los requisitos definidos y todo lo relativo al desarrollo software.
- **Quinto capítulo**: Describe la cronología del proyecto mostrando una línea temporal y los eventos más relevantes.
- **Sexto capítulo:** Este último capítulo contiene la conclusión del TFG, al igual que el trabajo futuro que le espera al proyecto.

1.4. Colaboraciones

Este proyecto lo han llevado a cabo dos estudiantes de la Universidad Politécnica de Valencia del grado de Ingeniería Informática. Jesús Fiérrez Ruipérez, encargado de realizar el desarrollo del *Frontend* de la aplicación en su respectivo TFG, y Pablo Bueno Fuentes, el autor de esta memoria, encargado de realizar el desarrollo del *Backend* de la aplicación que se relata en este documento.

Ambos participaron en la generación de la idea de negocio y su evaluación, la elicitación de los requisitos necesarios para el proyecto, el desarrollo del Backlog, el desarrollo de los experimentos junto con la realización de los ciclos de Lean Startup y la toma de decisiones importantes del proyecto.

Cada uno se centró en su respetivo desarrollo, enfrentándose solos a los desafíos que se le presentaban.

La colaboración entre los alumnos se llevó a cabo mediante las herramientas Teams[3], Google Drive[4] y Jira[5], en las cuales se tomaban las decisiones importantes que afectaban a ambos, para así tener un control sobre el proyecto y dejar un objetivo claro. Una vez con la planificación realizada y los problemas resueltos, cada uno empezaba a trabajar por separado.

Cierta documentación del TFG es obtenida a partir de herramientas compartidas, debido a que el objetivo del proyecto es el mismo.

En la finalización de cada Sprint se realizaron reuniones de retrospectiva con el fin de mejorar la metodología usada y buscar posibles mejoras a poner en práctica para los siguientes Sprints.

Por último, también se ha obtenido la colaboración de 3 alumnas de la Universidad Politécnica de Valencia del grado de Bellas artes, Andrea Palomo Gordón, Joanna López Cremades y Noemí Ortega Rodríguez, las cuales aportaron un diseño centrándose en la interfaz de usuarios, para ofrecer un producto con mejor apariencia, y en la experiencia de usuario para mejorar la sensación general del usuario al interactuar con el producto y así de esta manera desarrollar un producto más profesional.

[3] https://www.microsoft.com/es-es/microsoft-teams/group-chat-software
[4] https://www.google.com/intl/es_es/drive/
[5] https://www.atlassian.com/es/software/jira

2. Evaluación de idea de negocio

Se ha realizado una evaluación de la idea de negocio del proyecto *Social Lift* para determinar la viabilidad y potencial que podría tener la aplicación en el mercado actual, buscando definir una base sólida en la toma de decisiones y funcionalidades principales de la aplicación.

2.1. Clientes

Los clientes potenciales son personas o entidades que tendrían cierto interés en nuestro producto sin aún convertirse en un cliente activo.

Es importante analizar los clientes potenciales que utilizan *Social Lift*, por lo que a continuación se describen los perfiles de usuarios a los que nuestro producto va dirigido, sus necesidades y expectativas.

- **Usuarios experimentados en el mundo del *fitness* y culturistas:** incluye a usuarios con experiencia previa en el *fitness* que busquen herramientas para optimizar su entrenamiento, tener seguimiento de su progreso y compartir experiencias.
- **Usuarios principiantes en el *fitness*:** Incluye a personas nuevas en el mundo del *fitness* en busca de orientación y apoyo, de forma que se les proporciona el conocimiento aportado por usuarios expertos.
- **Usuarios deportistas:** Incluye a atletas y deportistas que busquen mejorar su rendimiento físico mediante el seguimiento de su progreso, la participación en competiciones y la motivación.
- **Usuarios Entrenadores personales y profesionales:** incluye a entrenadores personales independientes y profesionales en el campo del *fitness* que buscan una herramienta en la cual distribuir su conocimiento, al igual que supervisar la rutina de entrenamiento de sus clientes.
- **Posibles gimnasios y centros deportivos:** incluye a propietarios y gerentes de centros deportivos interesados en incorporar *Social Lift* como parte de sus servicios ofrecidos, gestionando rutinas de entrenamiento únicas, y la posibilidad de interactuar con sus miembros y mejorar su comunidad.

Para nuestro proyecto diferenciamos dos tipos de roles de usuarios:

- **Usuario básico**: Usuario con acceso gratuito a las funcionalidades fundamentales de la aplicación, pudiendo disfrutar de una experiencia sólida y funcional que satisfaga las necesidades esenciales de la gestión de rutinas y el seguimiento de su progreso.
- **Usuarios *Premium***: Usuarios con acceso a características avanzadas exclusivas de la aplicación mediante una suscripción de pago. Tendrán acceso a las funcionalidades de los usuarios básicos y recibirán beneficios extras como *coaching premium online* y ejercicios *premium*.

2.2. Estudio de mercado

Una parte fundamental en el estudio de la idea de negocio es investigar a los competidores y, para entender el entorno competitivo en el que se encuentra *Social Lift*, es necesario realizar un análisis de mercado. Se ha llevado a cabo un estudio sistemático de los productos y servicios que ofrecen sus principales rivales. Esto ayuda a identificar las fortalezas y debilidades de la competencia, así como las posibles oportunidades en el mercado, resaltando características esenciales que la aplicación debería tener en este sector o aspectos únicos distintivos que la diferenciarían y harían de *Social Lift* una aplicación más atractiva para los usuarios.

Se ha realizado un estudio sobre aplicaciones que ofrecen características similares a las de *Social Lift*, detallando las funcionalidades y servicios para luego representarlas en una tabla comparativa.

EFFORT APP

Effort App[6] es una aplicación móvil que da soporte en el seguimiento de los entrenamientos físicos. Dispone de herramientas para registrar el peso y las medidas corporales del usuario, así como funcionalidades para la creación de rutinas personalizadas y para la visualización de estadísticas detalladas de progreso.

Los usuarios pueden registrar sus entrenamientos, agregando ejercicios y organizándolos por semana. Además, tienen la opción de asociar estas rutinas a días específicos en el calendario para una planificación más estructurada.

La aplicación permite tener seguimiento regular del peso y los ejercicios realizados. Estos datos se utilizan para generar gráficos estadísticos personalizados, lo que permite a los usuarios monitorear su rendimiento.

En la figura 1 se puede observar que la aplicación se organiza por bloques de entrenamiento, en cada bloque se crean varios días de entrenamiento para luego asociar ejercicios a esos días. La interfaz de usuario de la aplicación no es del todo intuitiva, con ciertos problemas de funcionamiento.

[6] https://efortcoach.com/

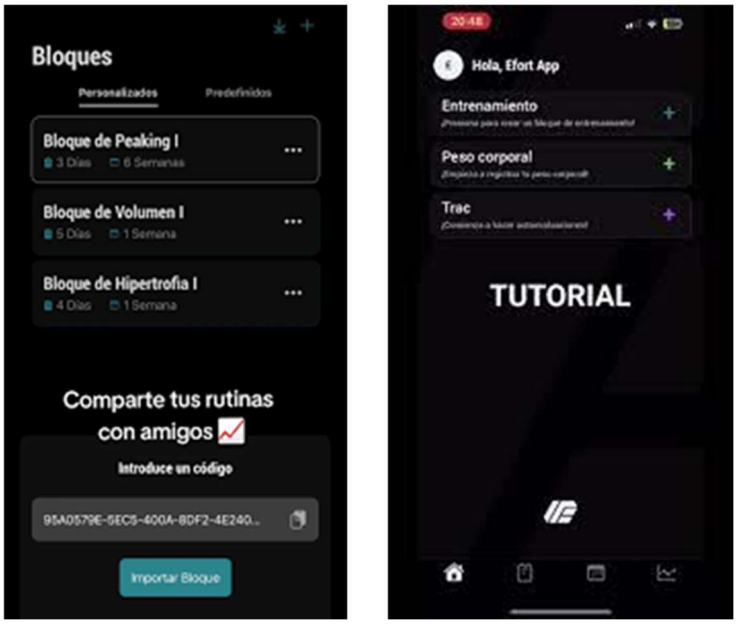

Figura 1. Interfaz de Effort App

STRAVA

Strava[7] es una aplicación móvil para deportistas y atletas. Les permite registrar y tener seguimiento de actividades deportivas al aire libre como correr o ir en bicicleta. Ofrece funcionalidades para compartir las actividades a otros usuarios y participar en desafíos o competiciones.

Los usuarios de Strava pueden registrar sus actividades utilizando dispositivos con GPS como relojes deportivos. Permite compartir experiencias con la comunidad de Strava y recibir retroalimentación sobre su entrenamiento. De igual manera, Strava dispone de 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.

Está mayoritariamente centrada en actividades que tengan un recorrido geográfico. En la figura 2 se muestra la interfaz de Strava donde se puede ver un muro interactivo de actividades que realizan los usuarios, el recorrido realizado representado en un mapa, y diferentes medidas relevantes en las actividades como el tiempo o la distancia recorrida.

[7] https://www.strava.com/

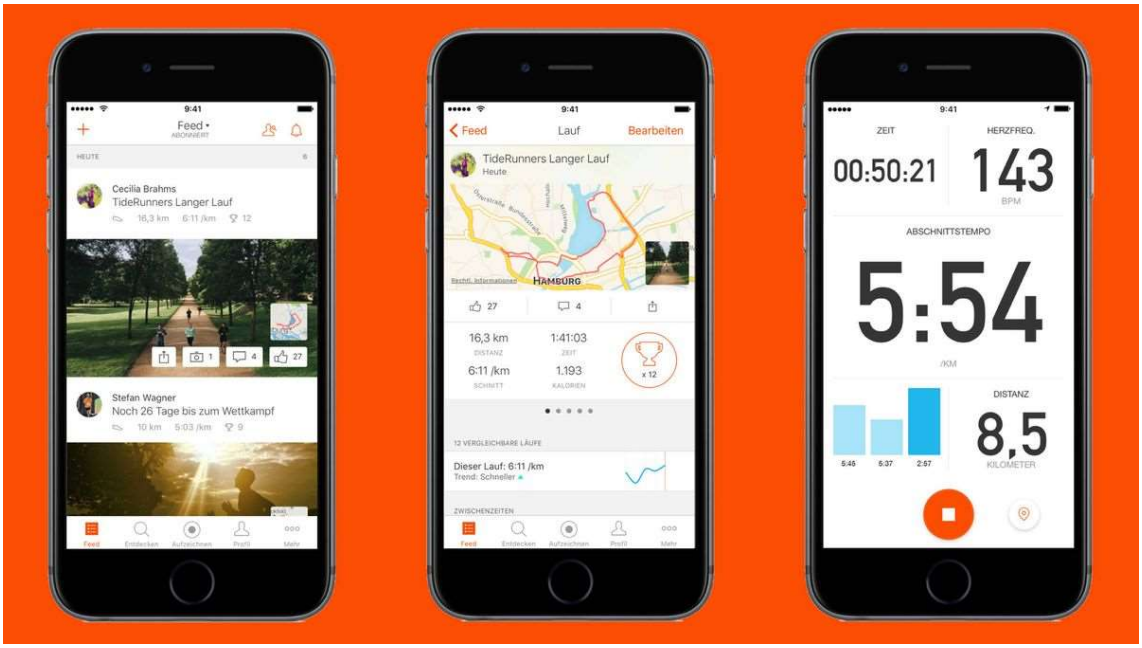

Figura 2. Interfaz de Strava.

Hevy

Hevy[8] es una aplicación móvil centrada en el levantamiento de pesas. Es posible registrar rutinas de entrenamiento y hacer seguimiento de estos. Ofrece una biblioteca de ejercicios y técnicas de entrenamiento propias de la aplicación.

En la aplicación, puedes registrar rutinas, agregar ejercicios predeterminados y 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, la figura 3 muestra una interfaz de estadísticas de un ejercicio teniendo en cuenta el mayor peso que ha levantado en los diferentes días.

Se puede 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 y con Strava, ocultar las opciones de red social y cambiar información sobre el perfil de usuario.

[8] https://hevy.com/

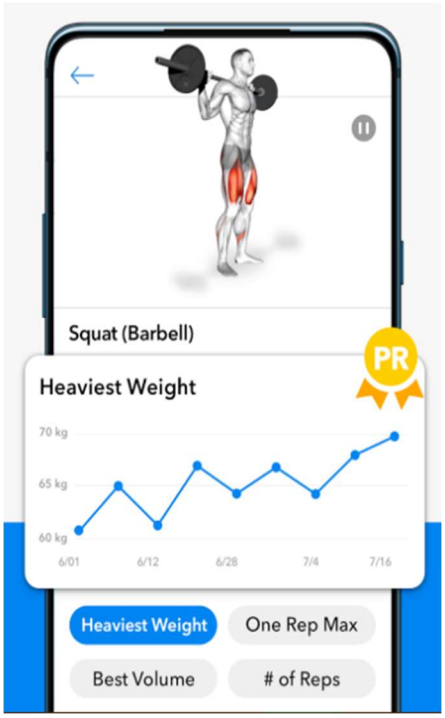

Figura 3. Interfaz de Hevy.

Gymshark training and fitness

Gymshark[9] es una Aplicación móvil que ofrece una gran variedad de programas de entrenamiento diseñados por entrenadores y deportistas profesionales. Dispone de herramientas para tener seguimiento en tu progreso y ofrece rutinas variadas según el tipo de entrenamiento que se desea realizar, con diferentes planes de entrenamiento, dando la opción de personalizarlos a tu gusto y mostrando diferentes estadísticas sobre tu progreso. En la figura 4 se muestra las distintas interfaces principales de la aplicación.

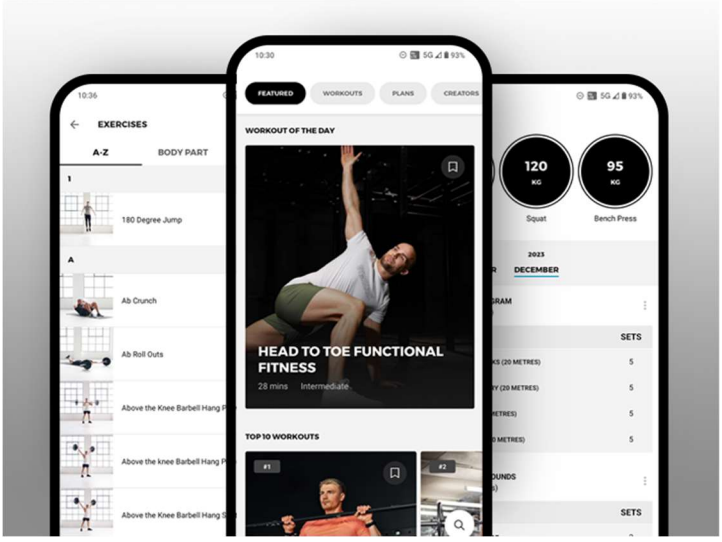

Figura 4. Interfaz de Gymshark.

[9] https://central.gymshark.com/article/the-gymshark-training-app-for-ios-and-android

2.3. Tabla comparativa

Tras el análisis, se ha desarrollado una tabla comparativa en la que se muestran 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. Dicha tabla se representa en la tabla 1.

	Effort App	Strava	Hevy	Gymshark Training and fitness	Social Lift
Registro de actividades deportivas	Si	Si	Si	Si	Si
Calendario interactivo	Si	No	No	No	Si
Estadísticas de progreso de ejercicios	Si	Si	Si	Si	Si
Estadísticas personales	No	No	Si	No	Si
Proporcionar rutinas	No	Si	Si	Si	Si
Seguimiento de usuarios	No	Si	Si	No	Si
Feed de rutinas	No	No	Si	No	Si
Registro de ejercicios	Si	Si	Si	Si	Si
Registro de rutinas/entrenamiento	Si	Si	Si	Si	Si
Proporción de Rutinas predeterminadas	No	Si	Si	Si	No
Foro Usuarios	No	Si	No	Si	Si
Multiplataforma	Si	Si	Si	Si	Si
Anuncios de actividades	No	No	Si	No	No
Vinculación entre Gimnasios/Lugares	No	Si	No	No	Si
Rankings	No	Si	No	No	Si
publicación audiovisual	No	Si	Si	No	Si
Integración con inteligencia artificial	No	No	No	No	A MEDIAS (en un futuro)
Personalización de entrenamiento	Si	Si	Si	SI	SI

Coaching online	**No**	**No**	**No**	**No**	**Si**
Competiciones *online*	**No**	**Si**	**No**	**No**	**SI**
Suscripciones Premium	**No**	**Si**	**Si**	**No**	**Si**
Suscripciones Gratuitas	**Si**	**Si**	**Si**	**Si**	**Si**

Tabla 1. Comparativa de aplicaciones

Social Lift busca cubrir todas las funcionalidades básicas que ofrecen los distintos competidores y además ofrecer herramientas innovadoras como el coaching online o competiciones y rankings para motivar a los usuarios. Por otro lado, se propone ofrecer a planes futuros integración con inteligencia artificial, puede ser en la recomendación de rutinas o en un *chatbot* que ofrecerá consejos y advertencias en los ejercicios que se realicen.

A diferencia de Strava o Hevy, que ofrecen rutinas predeterminadas, *Social Lift* busca crear una comunidad sólida en la que cada usuario aporte su experiencia, por lo que las rutinas son proporcionadas directamente por los usuarios.

Por otro lado, *Social Lift* utiliza un foro de usuarios y un muro social interactivo (*feed)* para que la aplicación funcione como una red social, ampliando la interacción de usuarios. Asimismo, para fomentar la comunidad, las rutinas se pueden asociar a un gimnasio y así descubrir nuevos usuarios a partir de los centros a los que asista.

Por último, una de las funcionalidades más fuertes de *Social Lift* es el Calendario interactivo, el cual, permite a los usuarios planificar sus rutinas a lo largo de los días de la semana de manera clara y ordenada. Las rutinas se crean y se añaden directamente desde esa funcionalidad.

2.4. Proyección ingresos y gastos

Uno de los aspectos más importantes a tener en cuenta es el modelo de negocio y la proyección económica, por lo que se ha realizado una estimación simple de ingresos y gastos en los primeros 3 años de vida del producto, de forma que nos ayuda a estudiar la viabilidad del proyecto.

Las fuentes principales de ingreso son las siguientes:

- Suscripciones de Coaching *premium*: el usuario paga por una suscripción que proporciona servicio de coaching en línea (*online*). El precio es de 7 € por trimestre.
- Publicidad de marcas de la comunidad *fitness*: Distintas marcas relacionadas con la comunidad *fitness* pueden publicitar sus productos dentro de nuestra aplicación pagando una cuota trimestral recurrente de 1500 €.

El análisis que se muestra a continuación se ha hecho trimestralmente, pero en las figuras solo se muestra un resumen cogiendo el 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 €
Total Ingresos	27,395 €	76,175 €	359,117 €

Tabla 2. Ingresos Trimestrales

En la tabla 2 se muestra una estimación de los usuarios suscritos al servicio de *coaching* y al número de marcas que publicitarían sus productos en nuestra aplicación. Para la estimación de usuarios se ha tenido en cuenta el dato mencionado en la motivación proporcionado por Statista[10] en el que aproximadamente 5,4 millones de personas son miembros de gimnasio en España. Suponiendo que al final del primer año un 4% (216.000 personas) utilicen *Social Lift* y el 2% de esos usuarios (4.320 personas) se suscriban a la membresía *premium* nos dan una aproximación para realizar la estimación.

Total Ingresos	27,395 €	76,175 €	359,117 €
Gastos Anuales			
Infraestructura Cloud	500 €	1,000 €	1,000 €
Ordenadores	2,000 €	0 €	0 €
CEO - Director Ejecutivo - Desarrollador Senior	6,000 €	8,000 €	10,000 €
CTO - Director de Tecnologia - Desarrollador Senior	3,600 €	6,000 €	6,000 €
Desarrollador Junior	0 €	0 €	4,000 €
Desarrollador de mantenimiento	4,500 €	4,500 €	5,000 €
Desarrollador de mantenimiento	4,500 €	4,500 €	5,000 €
Director de Marketing	5,000 €	5,000 €	6,000 €
Personal administrativo	6,000 €	6,000 €	6,500 €
Campaña Marketing	2,000 €	5,000 €	7,000 €
Internet, electricidad y alquiler	1,000 €	2,000 €	2,500 €
Gestoria	400 €	600 €	800 €
Total Gastos	35,500 €	42,600 €	53,800 €
Resultado T4	-8,105 €	33,575 €	305,317 €
Resultado Trimestral Acumulado	-39,677 €	42,191 €	746,919 €

Tabla 3. Gastos trimestrales

También se hizo una estimación de los gastos mostrados en la tabla 3, que presentaría el proyecto. Los cuales caben destacar el aumento en el servicio en la nube *(cloud)* debido a que buscamos un producto escalable. La contratación de un CEO y un CTO encargados para dirigir el proyecto, desarrolladoras de mantenimiento de la aplicación, un desarrollador junior, una campaña y director de marketing con el fin de captar

[10] https://es.statista.com/

más clientes y promover más ingresos, personal administrativo y, por último, temas relacionados con la gestoría, internet y electricidad.

El primer año se tiene que hacer una inversión para llevar a cabo el proyecto, pero teniendo en cuenta la gran cantidad de usuarios potenciales a mitad del 2 año ya se llega a Punto de Equilibrio como se muestra en la figura 5, de igual manera, se aprecia que el crecimiento del modelo de negocio es exponencial.

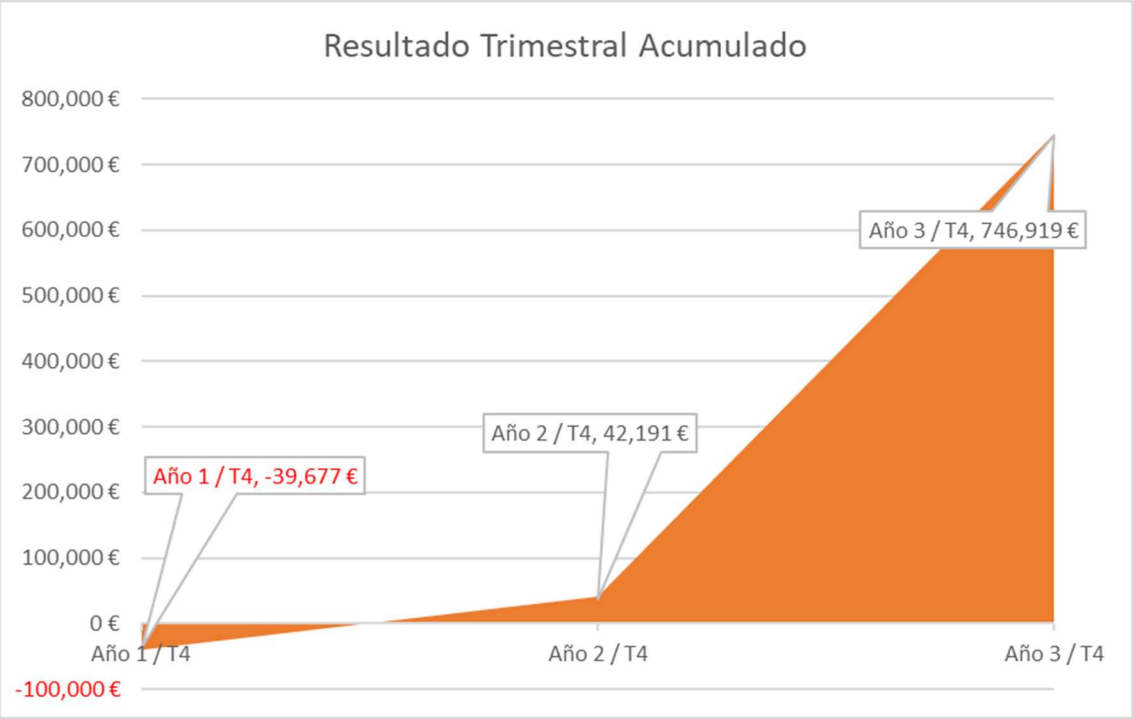

Figura 5. Punto de equilibrio económico

2.5. Análisis DAFO

El Análisis DAFO [4] es una herramienta de estudio utilizada para evaluar la situación de una empresa o proyecto, analizando sus características internas (Debilidades y Fortalezas) y su entorno externo (Amenazas y Oportunidades) en una matriz cuadrada. Es una herramienta fundamental para comprender la situación real en que se encuentra un proyecto y planificar estrategias futuras.

El objetivo principal de la creación de la matriz DAFO es representar de manera más clara cuáles son los puntos a favor y en contra de la idea de negocio, lo que permite orientar mejor las estrategias.

DEBILIDADES	AMENAZAS
• Aplicación desconocida, esto implica un gran coste humano en entrometerse de lleno en el nicho de mercado. • Poca experiencia en gestión empresarial.	• Competencia de aplicaciones similares. • Aplicaciones similares asentadas en el mundo del fitness.
FORTALEZAS	**OPORTUNIDADES**
• No implica una gran inversión económica. • Buen conocimiento del público objetivo. • Enfoque único.	• Nicho de mercado en pleno auge. • Interés de la aplicación a gente joven. • Enfoque de la aplicación como una red social que pueda crear una amplia comunidad en el mundo del *fitness*.

Tabla 4. Análisis DAFO

Como se puede observar en la tabla 4, las principales debilidades de nuestra idea de negocio provienen de la falta de experiencia en la gestión empresarial y al elevado coste inicial necesario para ganar visibilidad en el mercado y poder competir con la competencia. Sin embargo, hay ciertas fortalezas que nos da ventaja frente a otros productos, Como el conocimiento en el sector del *fitness* y un enfoque único para desarrollar una aplicación ideal.

Durante la evaluación de la idea de negocio también hemos encontrado amenazas debido a la competencia con productos software similares que, aunque no ofrecen todas las herramientas de *Social Lift*, ya se encuentran asentados en mercado. Sin embargo, existen diferentes oportunidades, como que el nicho del mercado 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*.

2.6. Lean Canvas

El Lean Canvas es una herramienta de gestión estratégica utilizada principalmente en el ámbito del emprendimiento e innovación [5], nos sirve como herramienta visual para analizar el modelo de negocio de manera rápida y clara para el cliente. El Lean Canvas está mostrado en la tabla 5.

<table>
<tr><td colspan="3">7 costos
Los costes principalmente son de 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 gestoría, servicios e infraestructuras cloud y material personal como ordenadores</td><td colspan="3">6 ingresos
Modelo de suscripciones premium a los clientes para acceder a funcionalidades como coach de personal experto.
Publicidad de marcas relacionadas con el mundo del fitness dentro de la aplicación</td></tr>
<tr><td>2 problema
Los usuarios que realizan actividades para ejercitarse carecen de una forma de registro de rutinas eficiente y especializada que les ayude a hacer seguimiento.</td><td>4 solución
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.</td><td>3 proposición de valor
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 Coaches que ayuden a los usuarios a mejorar y progresar.</td><td>9 ventaja competitiva
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.</td><td>1 clientes
Nuestros early adopters son clientes de gimnasios con cierta experiencia en el mundo del fitness.
Por otro lado, distintos gimnasios podrían estar interesados en saber y planificar las rutinas de sus clientes, por lo que podrían utilizar nuestro servicio.
A deportistas profesionales o personas influyentes en redes sociales nuestro producto les facilitaría en aumentar su comunidad.</td></tr>
</table>

Tabla 5. Lean Canvas

Cabe destacar la proposición de valor de entrenador en línea (*coach online)*, ya que es una solución innovadora que tienen muy pocos competidores.

2.7. Conclusiones de la evaluación

Tras el análisis de la idea de negocio, *Social Lift* muestra una serie de aspectos clave para garantizar su viabilidad y potencial en el mercado del *fitness*, la cual se presenta como una idea innovadora diseñada para satisfacer las necesidades de la comunidad.

En el mercado se revela que tiene un gran valor, combinando ciertas características de otras plataformas con servicios distintivos como el *coaching online*. Esta combinación lo posiciona favorablemente para atraer a una amplia gama de usuarios.

En la parte financiera, aunque se presentan desafíos iniciales con el costo de entrada al mercado y la inversión necesaria en *Marketing*, la proyección financiera muestra un gran potencial de crecimiento una vez que se establezca una base de usuarios.

Existe una gran oportunidad en el mercado y *Social Lift* busca integrase en el sector de *fitness* para ofrecer soluciones innovadoras y eficaces en la gestión de entrenamientos.

En conjunto, todos estos aspectos hacen que *Social Lift* tenga potencial en convertirse en una herramienta funcional para el mundo *fitness*.

3. Tecnología Utilizadas

En este apartado se ilustran las distintas tecnologías principales usadas para poner en marcha la API REST que sustenta la aplicación *Social Lift.*

3.1. Java Spring Boot y Maven

Spring Boot[11] es un marco de trabajo (*framework*) de Código abierto para la construcción de API RESTful [6]. Ofrece funciones de inyección de dependencias que permite definir dependencias en objetos para luego ponerlos en el contenedor Spring[12], lo cual permite crear componentes sin conexión directas ideales para aplicaciones de red distribuidas.

Ofrece también todas las herramientas y funciones que necesitan para crear aplicaciones Java EE multiplataforma sin conexión directa para que se ejecuten en cualquier entorno.

Se ha escogido esta tecnología para desarrollar la API porque ofrece las siguientes ventajas:

- Java Spring Boot lleva consigo prestaciones de configuración automáticas incorporadas, lo cual permite acelerar el inicio de desarrollo de la aplicación, debido a que no es necesario configurar ciertas dependencias manualmente y reduce errores.
- Spring Boot puede crear aplicaciones que no dependen de un sistema externo, permitiendo que se ejecutan por sí solas.
- También ofrece una gran cantidad de herramientas y bibliotecas que nos ayuda a construir aplicaciones robustas y escalables de manera eficiente.

Como herramienta de gestión de proyectos y dependencias se utiliza Maven[13], permitiendo centrarse en la lógica de negocio y las funcionalidades de la aplicación sin preocuparse demasiado por la configuración y gestión de dependencias inicialmente.

3.2. IntelliJ IDEA

IntelliJ IDEA[14], desarrollado por JetBrains[15], es un entorno de desarrollo integrado (IDE) con funcionalidades avanzadas, que proporciona soporte para gran cantidad de tecnologías y marcos de trabajo, al igual que posee herramientas de refactorización. Dispone de un entorno de desarrollo productivo de Java y, gracias a sus herramientas de depuración y finalización de código inteligente, permite trabajar de manera más productiva en la creación del *Backend* necesario para la aplicación.

[11] https://spring.io/projects/spring-boot
[12] https://spring.io/
[13] https://maven.apache.org/
[14] https://www.jetbrains.com/es-es/idea/
[15] https://www.jetbrains.com/es-es/

Se escogió este IDE frente a muchos otros debido a sus avanzadas características, además de la facilidad de crear una aplicación usando Spring Boot mediante el Spring Intializr integrado en el IDE

3.3. Supabase

Supabase[16] es una plataforma de base de datos y autenticación de Código abierto. Simplifica todo el proceso de creación y gestión de aplicaciones web, mediante una infraestructura segura para almacenar datos y autentificar usuarios.

Dispone de una API RESTful para interactuar con bases de datos PostgreSQL, lo que simplifica el desarrollo. En el *Backend* de *Social Lift* Supabase es utilizado para gestionar y almacenar los datos necesarios en la aplicación.

Se escogió esta tecnología frente a otros motores de base de datos, ya que ofrece herramientas accesibles *online*, gratuitas y escalables que permiten crear una infraestructura de base de datos según las necesidades de la aplicación, con un rendimiento óptimo. Además, ofrece autenticación incorporada, lo que facilita la implementación de seguridad en la aplicación.

3.4. Render y Docker

Render[17] es una plataforma que ofrece una infraestructura de alojamiento moderna y escalable para aplicaciones web y API. Simplifica el proceso de implementación y administración de aplicaciones al proporcionar servicios como escalado automático, certificados SSL gratuitos y despliegue continuo.

Este servicio de alojamiento es ideal para desplegar un servidor *online* para el desarrollo de la API REST; su enfoque automatizado de despliegue simplifica considerablemente el proceso de implementación y mantenimiento del *Backend* de *Social Lift*.

Esta tecnología ayuda a establecer un servidor que esté en constante marcha para así asegurar que la API REST ofrece un servicio constante y que la aplicación podrá ser utilizada siempre y cuando se disponga de conexión a internet.

Para desplegar la API es necesario que esté contenida en un Docker[18], y Render facilita su empaquetamiento. La aplicación y sus dependencias son encapsuladas en un contenedor ligero y portátil, lo que garantiza su entorno de ejecución consistente y reproducible, por lo que la aplicación se podrá ejecutar de la misma manera en cualquier entorno y así prevenimos posibles errores de despliegue.

[16] https://supabase.com/
[17] https://render.com/
[18] https://www.docker.com/

3.5. GitHub

GitHub[19] es una plataforma de desarrollo colaborativo de software que utiliza el control de versiones Git. Es ampliamente utilizado por desarrolladores de software para alojar proyectos, colaborar en código, revisar cambios y gestionar versiones de software de manera eficiente.

En el desarrollo, es utilizado para alojar y gestionar el código fuente del *Backend* de la aplicación, el cual se utilizará para realizar un seguimiento de cambios y gestionar versiones de manera eficiente utilizando control de versiones Git.

Se ha escogido esta tecnología debido a su facilidad de uso, además de que ofrece servicios gratuitos y disponibles para cualquier desarrollador.

3.6. Jira

Jira[20] es una herramienta de gestión de proyectos desarrollada por Atlassian[21], se ha escogido porque ofrece ciertas funcionalidades útiles para llevar a cabo la metodología, como son:

- Permite crear, asignar y priorizar tareas y cada una puede ser detallada.
- Permite gestionar un Backlog de desarrollo que incluye unidades de trabajo creadas y nos permite ordenarlas y priorizarlas.
- Es compatible con Scrum y Kanban.
- Fomenta la colaboración mostrando los responsables de las tareas y cualquier actualización relevante.

3.7. JUnit y Mockito

Junit[22] es un *framework* de pruebas unitarias para el lenguaje Java, Mockito[23] es un *framework* de simulación para las pruebas, se utiliza para crear objetos simulados para probar componentes de forma aislada. Al usar Junit con Mockito se puede asegurar que los componentes actúan de forma correcta con sus dependencias

Se escogió estas tecnologías, ya que es la mejor forma para realizar pruebas para el lenguaje de programación usado en el *Backend.*

[19] https://github.com/
[20] https://www.atlassian.com/es/software/jira
[21] https://www.atlassian.com/
[22] https://junit.org/junit5/
[23] https://site.mockito.org/

3.8. Swagger

Para facilitar el desarrollo, se ha hecho uso de Swagger[24], una herramienta utilizada para diseñar, documentar y consumir servicios RESTful. Permite generar documentación interactiva automática de la API, lo que nos facilita comprender su funcionamiento. También ayuda a probar los diferentes Endpoints para validar el correcto funcionamiento del *Backend*, de manera que se pueden realizar pruebas en las peticiones de manera aislada sin necesidad del *Frontend*.

3.9. Loader.io

Loader.io[25] es una herramienta diseñada para realizar pruebas de carga y estrés en aplicaciones web. Permite evaluar como las aplicaciones manejan grandes volúmenes de tráfico y determinar su rendimiento.

Se ha utilizado esta tecnología para comprobar el rendimiento de los Endpoints de *Social Lift*.

[24] https://swagger.io/
[25] https://loader.io/

4. Desarrollo de la solución.

4.1. Metodología

Las metodologías son herramientas que guían el desarrollo del producto y del proyecto, promoviendo la colaboración y la entrega de soluciones validadas por el cliente para así desplegar un producto con alta calidad.

4.1.1. LEAN STARTUP

Para el desarrollo del proyecto *Social Lift* se ha adoptado la metodología Lean Startup [7]. La cual se centra en la creación rápida de un producto mínimo viable (MVP) y en la iteración continúa basada en retroalimentación de usuarios potenciales (*earley adopters*). Esto ha permitido un enfoque ágil y flexible para el desarrollo de la solución, dando prioridad a las características claves y minimizando el tiempo en funcionalidades menos críticas.

En este proceso iterativo, se desarrollan MVP [8] con las funcionalidades básicas y esenciales que creemos que la aplicación debería tener. Cada MVP llevaba consigo un experimento controlado con usuarios que frecuentan gimnasios con el objetivo de validar los requisitos y las funcionalidades de la aplicación, al igual que recopilar informaciones directas de los usuarios sobre su experiencia de uso que nos será útil para desarrollar el MVP posterior.

4.1.2. Metodología ágil SCRUM

Por otro lado, también se ha utilizado la metodología ágil SCRUM [9] para gestionar el desarrollo de la solución de manera eficiente y adaptable.

Dentro del marco SCRUM, se emplean roles definidos como Scrum Master, para garantizar el correcto seguimiento de prácticas SCRUM, y Product Owner, para definir los requisitos del producto y organizar el Backlog. Mi compañero de *Frontend* y yo estamos a cargo de estos roles.

También se emplean ciclos de desarrollo cortos, conocidos como Sprints, que permiten producir un entregable y mantener un enfoque iterativo. Se realizan reuniones de retrospectiva al final de los Sprints para revisar el trabajo realizado e identificar posibles mejoras para el siguiente Sprint.

4.1.3. Backlog, Sprints y Workflow

El proyecto dispone de un Backlog priorizado con todas las tareas pendientes a realizar que deben completarse para llevar a cabo el proyecto, el cual se modifica constantemente, agregando nuevas tareas de corrección de errores o mejoras a medida que avanzaba el proyecto.

Durante el desarrollo del TFG se realizaron 5 Sprints de desarrollo de 3 semanas de duración aproximadamente. Durante cada Sprint se buscó completar un conjunto de unidades de trabajo (UT) seleccionadas según su prioridad en el Backlog, en la figura 6 y 7, se muestra el Sprint 1 y el Sprint 2 respectivamente, cada una con sus UT asociadas

Figura 6. Sprint 1

Figura 7. Sprint 2

Cada una de las unidades de trabajo sigue un flujo de trabajo (*Workflow*), el cual representa el conjunto de pasos que sigue la UT para que se dé por resuelta. El *Workflow* utilizado es mostrado en la figura 8:

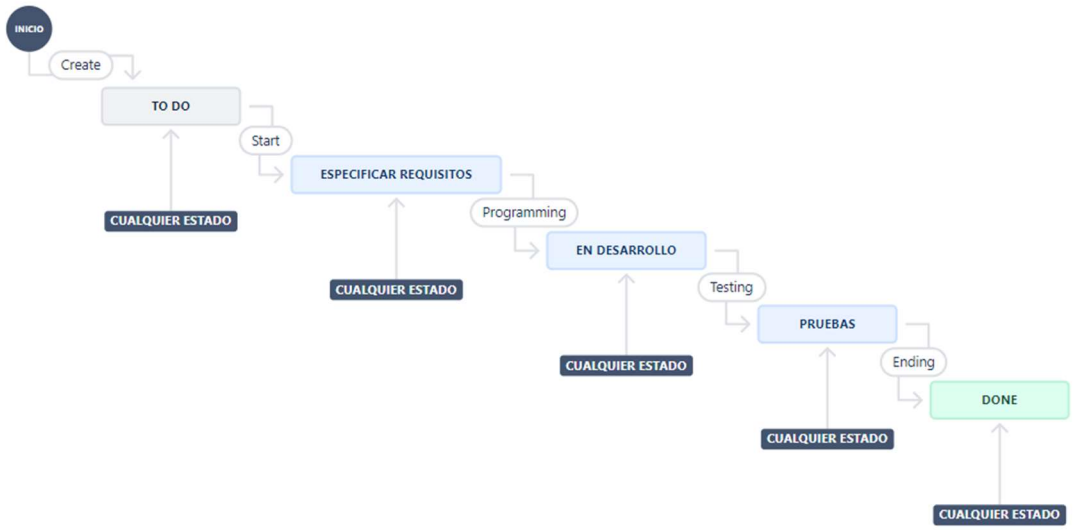

Figura 8. Workflow.

Cada estado representa lo siguiente:

- To do: Estado inicial de creación de una unidad de trabajo.
- Especificar Requisitos: Estado en el que se definen los requisitos que debe cumplir dicha unidad de trabajo, con una breve descripción, establecer las pruebas de aceptación, documentar la programación esencial y, si es necesario, poner un *mock up*.
- En Desarrollo: Estado en el que se lleva a cabo el desarrollo software de la unidad de trabajo.
- Pruebas: Estado en el que se validan las pruebas definidas anteriormente. Si todas pasan correctamente, puede pasar a *Done*.
- Done: Estado en el que se encuentran las unidades de trabajo resueltas.

4.1.4. Herramienta para gestión de proyectos

Para este proceso nos hemos apoyado en Jira como herramienta de planificación y seguimiento de tareas para monitorear el progreso del proyecto y la gestión del Backlog de desarrollo.

Jira permite la creación de épicas, que consisten en una gran característica compuesta por unidades de trabajo más pequeñas, lo que resulta interesante para la organización del proyecto.

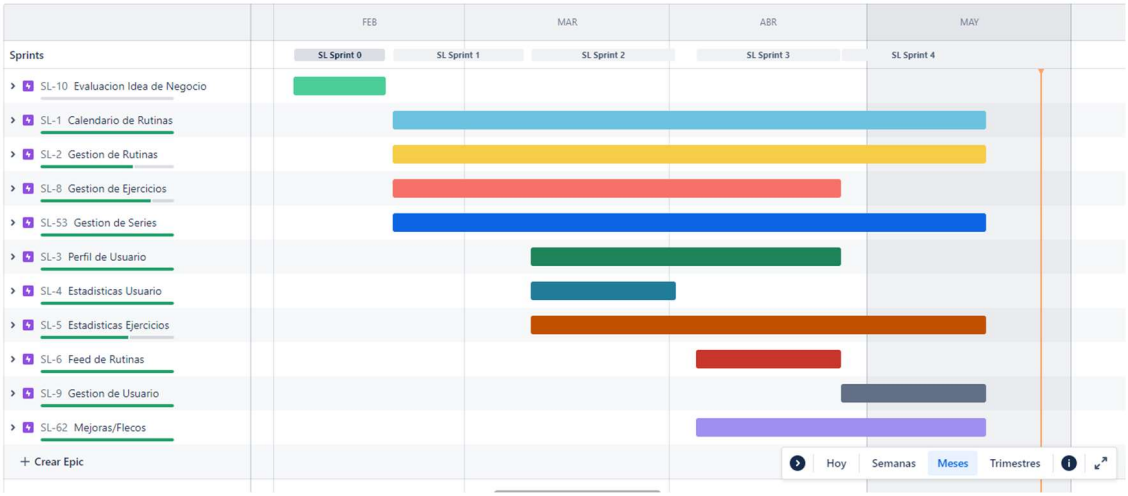

Figura 9. Planificación de las épicas.

En la figura 9 se muestra las diferentes épicas que se han creado para el desarrollo de la aplicación junto con su cronología de trabajo en que se han ido realizando.

Gracias a todas estas herramientas, el uso de Jira en el proyecto ha resultado fundamental a lo largo de la realización del TFG.

4.2. Requisitos

Antes de mostrar los requisitos se hace una descripción general de la aplicación.

4.2.1. Descripción de Social Lift.

Figura 10. Logo de Social Lift.

Social Lift, con logo mostrado en la figura 10, es una aplicación novedosa desarrollada para ayudar a la comunidad del *fitness* en la gestión de sus rutinas y el seguimiento de su progreso en los entrenamientos de manera fácil, sencilla y eficaz. Aporta una gran cantidad de ejercicios y rutinas personalizadas. La aplicación incluye un foro integrado que permite a los usuarios interactuar, compartir sus rutinas de entrenamientos y publicarlas, guardar rutinas de otros usuarios para su posterior uso, fomentando la colaboración y el intercambio de conocimiento dentro del ámbito del *fitness*.

La aplicación tiene una sección en el perfil del usuario donde se puede ver distintas estadísticas basadas en el progreso de cada persona, para conseguir objetivos y metas propuestas, permitiendo hacer seguimiento de su progreso.

Dispone de un calendario interactivo en la que se asocian rutinas en cada uno de los días del calendario. A cada rutina se le asocia un color para que sea mucho más fácil la búsqueda de rutinas creadas en días anteriores.

Proporciona la posibilidad de subir un video por cada ejercicio realizado para su posterior visualización. Cada ejercicio está asociado a una rutina, la cual se puede publicar para que otros usuarios vean tus entrenamientos y tus videos. Los usuarios pueden ver dichas publicaciones en un muro social (*feed)* de rutinas y también pueden guardar la rutina para tenerla a mano en el momento en que deseen realizarla.

Asimismo, aporta diferentes competiciones entre los usuarios para motivar a las personas entre ellos. Mostrando un *ranking* y consiguiendo logros de la aplicación según eventos realizados.

Si un gimnasio o una marca está inscrita en la aplicación, permite a los usuarios recibir nuevas funcionalidades o rutinas específicas.

Social Lift promueve la organización de rutinas de entrenamiento con el fin de simplificar y mejorar el progreso en el gimnasio y crear una comunidad de *fitness* sólida.

4.2.2. Casos de Uso

Después de realizar la investigación en el estudio de mercado, comparando las funcionalidades ofrecidas por las aplicaciones competidoras, se han identificado ciertos requisitos necesarios que *Social Lift* tiene que cumplir. Para estructurar de manera clara las funcionalidades, se ha elaborado un diagrama de casos de uso que sirve como guía estratégica, permitiendo priorizar el desarrollo de acuerdo con la importancia y relevancia para los usuarios representado en la figura 11.

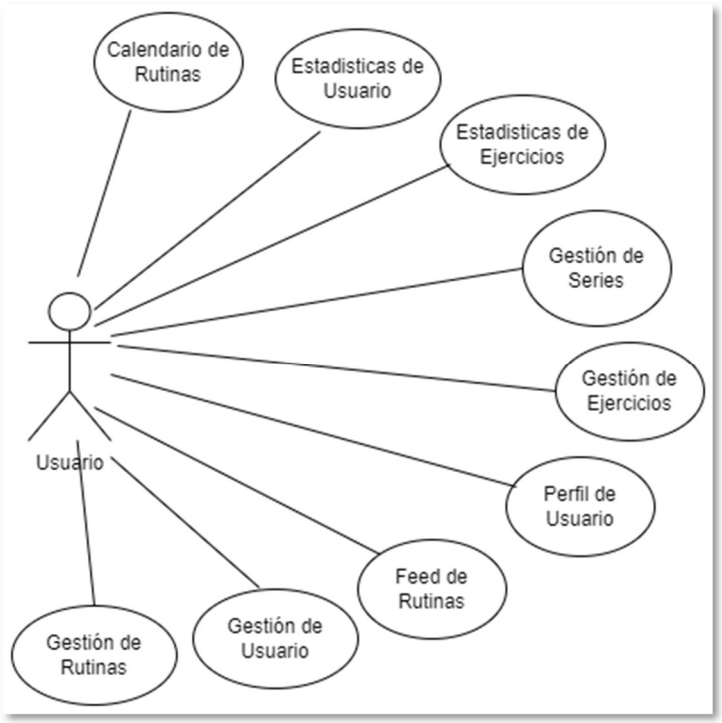

Figura 11. Épicas de la aplicación.

Cada uno de estos Casos de Uso representan las funcionalidades globales que tiene que ofrecer el sistema y cada uno de estos se desarrollará a través de unidades de trabajo necesarias para cumplir dicha funcionalidad.

Para cada funcionalidad es necesario definir requisitos tanto funcionales como no funcionales para definir como debe comportarse el sistema.

A continuación, se describen los distintos Casos de Uso. Para los requisitos no funcionales se priorizará la disponibilidad, seguridad y rendimiento.

4.2.2.1. Calendario de rutinas

Consiste en:

- Crear/añadir rutina: El sistema podrá crear nuevas rutinas personalizadas o añadir rutinas creadas basadas en plantillas.
- Navegación entre días y meses: El sistema podrá mostrar la información de las rutinas, dependiendo del día y mes en que se encuentre el usuario.
- Eliminar Rutina: El sistema podrá eliminar una rutina asociada a un día directamente desde el calendario siempre que lo desee el usuario.

4.2.2.2. Gestión de rutinas

Consiste en:

- Crear rutina: El sistema podrá crear nuevas rutinas personalizadas o plantillas de rutinas, definiendo detalles como el nombre, descripción, ejercicios, tipo de grupo muscular que se ejercita, entre otros:
- Borrar rutina: El sistema podrá eliminar la rutina existente de la base de datos de la aplicación en el momento en que el usuario desee.
- Editar rutina: El sistema podrá editar cualquier información de las rutinas existentes.
- Publicar rutinas: El sistema podrá publicar sus rutinas creadas para que estén disponibles para otros usuarios de la plataforma.
- Buscar rutina: El sistema podrá buscar rutinas en la plataforma utilizando diferentes criterios como palabras clave o tipo de grupo muscular ejercitado.
- Guardado rutinas: El sistema podrá guardar rutinas en una lista para acceder a ellas fácilmente en el momento que lo desee el usuario.

4.2.2.3. Gestión de ejercicios

Consiste en:

- Crear ejercicio: El sistema podrá crear nuevos ejercicios personalizados definiendo atributos como el nombre, descripción y tipo de grupo muscular ejercitado.
- Borrar ejercicio: El sistema podrá eliminar los ejercicios creados en el momento en que desee el usuario.
- Editar ejercicio: El sistema podrá modificar los atributos de los ejercicios creados.
- Añadir ejercicios a una rutina: El sistema podrá agregar ejercicios a una rutina específica, seleccionando los ejercicios de una lista, ya sea los ejercicios creados por dicho usuario o la colección de ejercicios creados por la comunidad de usuarios de la aplicación.
- Buscar ejercicios: El sistema podrá buscar ejercicios utilizando diferentes criterios como palabras clave o tipo de ejercicio.

4.2.2.4. Gestión de series

Consiste en:

- Crear serie: El sistema podrá crear nuevas series asociadas a un ejercicio y definir atributos como el peso, el tipo de peso, y el número de repeticiones.
- Editar serie: El sistema podrá modificar cualquiera de los atributos de las series creadas.
- Borrar serie: El sistema podrá eliminar una serie creada en un ejercicio en el momento en que desee el usuario.
- Postear un video de una serie: Los usuarios podrán agregar un video a una serie creada.

Como requisitos no funcionales relevantes en la gestión de series tenemos:

- Tiempo de subida de los videos: El guardado y la carga de los videos deben completarse en menos de 10 segundos para archivos de hasta 100 MB, permitiendo a los usuarios cargar contenido sin demoras significativas.
- Restricción: no puede ser posible agregar más de un video a una serie.

4.2.2.5. Perfil del usuario

Consiste en:

- Seguimiento de usuarios: El sistema permitirá a un usuario seguir a otros usuarios para recibir actualizaciones sobre sus rutinas publicadas.
- Ver rutinas publicadas: El sistema dará como respuesta las rutinas publicadas del usuario que solicite la información.
- Ver rutinas guardadas: El sistema dará como respuesta las guardadas del usuario que solicite la información.
- Ver rutinas creadas: El sistema dará como respuesta las rutinas creadas del usuario que solicite la información.
- Ver estadísticas de los ejercicios: El sistema dará como respuesta las estadísticas de un ejercicio seleccionado del usuario que solicite la información.
- Ver estadísticas corporales: El sistema dará como respuesta las estadísticas corporales del usuario que solicite la información.
- Postear imagen: el sistema permitirá guardar una imagen a la base de datos y asociarla a un usuario.

Como requisitos no funcionales relevantes para el perfil del usuario:

- Tiempo de subida de los videos: El guardado y la carga de los videos deben completarse en menos de 10 segundos para archivos de hasta 100 MB, permitiendo a los usuarios cargar contenido sin demoras significativas.
- Restricción: Solo se podrán ver rutinas publicadas por usuarios que sigues.

4.2.2.6. Estadísticas del usuario

Consiste en:

- Editar medidas corporales: El sistema podrá actualizar las medidas de los grupos musculares básicos de un usuario.
- Editar peso: El sistema podrá podrán actualizar el peso de un usuario.
- Editar IMC: El sistema podrá actualizar el IMC de un usuario.

- Editar altura: El sistema podrá actualizar la altura de un usuario.

4.2.2.7. Estadísticas de los ejercicios

Consiste en:

- Crear estadísticas por ejercicio: El sistema dará como respuesta las estadísticas relevantes de un ejercicio, el cual es seleccionado por el usuario.
- Ver ranking de ejercicios: El sistema enviará la información necesaria para permitir crear un *ranking* de los usuarios de la aplicación comparando sus estadísticas.

4.2.2.8. Feed de rutinas

Consiste en:

- Ver rutinas publicadas: El sistema dará como respuesta una lista de rutinas publicadas por los usuarios seguidores del usuario *logueado*.
- Guardar rutinas: El sistema podrá guardar rutinas de ejercicios.
- Buscar usuarios: El sistema podrá buscar a otros usuarios de la plataforma por su nombre de usuario.
- Buscar rutinas: El sistema podrá buscar rutinas de ejercicios utilizando palabras clave.

Como requisitos no funcionales relevantes en el Feed de rutinas tenemos:

- Tiempo Real: El servicio se debe actualizar en tiempo real para que cuando un usuario que sigas publique una rutina, se muestre un aviso de recarga en la pantalla.
- Restricción: No se deben mostrar en el feed rutinas con fechas muy antiguas.

4.2.2.9. Gestión del usuario

Consiste en:

- Registrarse: El sistema podrá registrar usuarios en la plataforma proporcionando la información básica necesaria.
- Login: El sistema permitirá el iniciar sesión en la plataforma a los usuarios verificando sus credenciales de inicio de sesión.
- Seguimiento de usuarios: El sistema dispondrá de la lógica de seguimiento de usuarios.

4.2.2.10. Requisitos no funcionales del sistema

Como requisitos no funcionales relevantes en la gestión de series tenemos:

- Disponibilidad: El sistema debe estar siempre disponible para todos los usuarios.
- Rendimiento: El sistema debe cargar la información en menos de 6 segundos, incluso con grandes cantidades de datos (1000 usuarios)
- Seguridad: los datos relacionados con los ejercicios deben estar protegidos adecuadamente con seguridad como autenticación de usuarios.

4.3. Diseño.

En esta sección están los elementos fundamentales que conforman la estructura y el funcionamiento de la aplicación.

4.3.1. La Arquitectura

Este concepto define la organización general de los componentes y su interacción. En la arquitectura del *Backend* desarrollada con Spring Boot sigue un enfoque modular y escalable diseñado para facilitar la interacción de los componentes del sistema. Este se divide en varias capas [10].

En la capa de presentación empleamos controladores REST que manejan las solicitudes HTTP entrantes, provenientes del Frontend, y dan como respuesta archivos de texto en formato JSON, los cuales representan la información lógica de la aplicación. Estos controladores son los responsables de dirigir el flujo de datos entre los clientes y el sistema.

En la capa de lógica de negocio se implementan los servicios que realizan operaciones específicas requeridas por los clientes. Estos servicios procesan la lógica de negocio, validan los datos y coordinan la interacción con la capa de acceso de datos para satisfacer las solicitudes de los clientes de manera efectiva.

La capa de acceso a datos y persistencia sirve como método de conexión con la base de datos en Supabase, utiliza tecnologías como Spring Data JPA para interactuar con la base de datos. Los repositorios de Spring data proporcionan métodos para realizar las operaciones CRUD en la base de datos, aumentando la productividad del código.

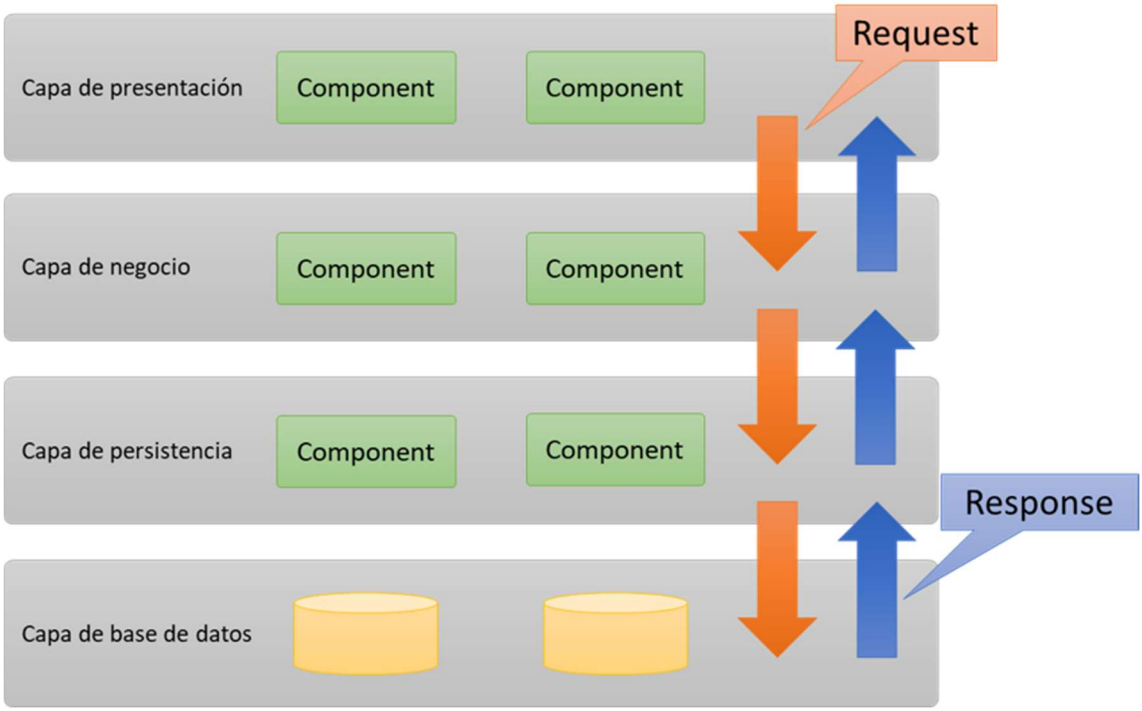

Figura 12. Arquitectura del sistema.

En la figura 12 se observa la representación y relaciones que tiene cada una de las capas. Cada componente representa una entidad del diagrama de clases, por lo que cada entidad dispone de un controlador, de un servicio, de un repositorio y un modelo en la base de datos necesarios para el flujo de la información.

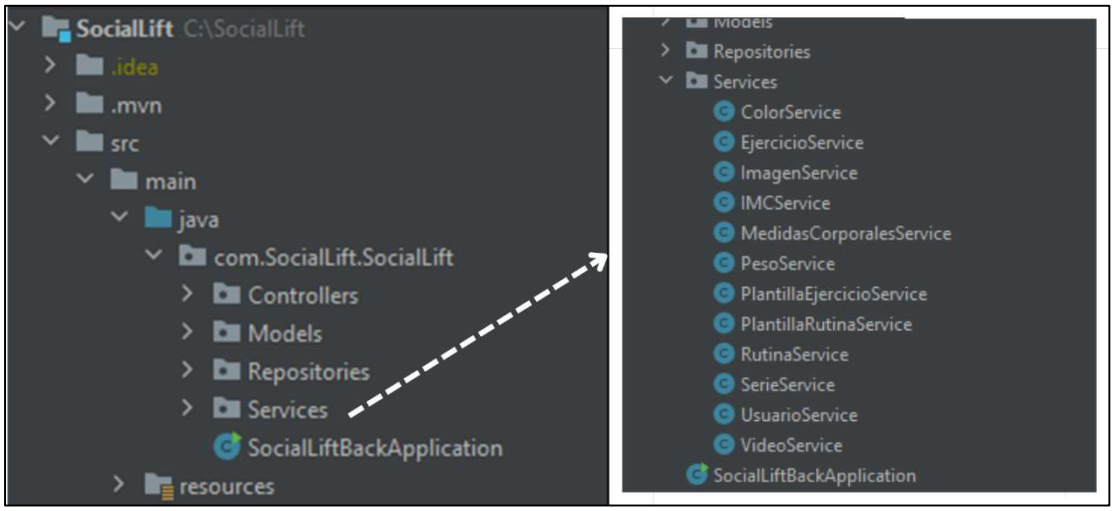

Figura 13. Carpetas de código de Social Lift.

En la figura 13 se muestra como está distribuido el código del *Backend* de *Social Lift*. Estas carpetas representan las distintas capas mencionadas anteriormente. Cada una está compuesta por archivos de Java, los cuales representan los componentes (Component) de la figura 12.

- En la carpeta “CONTROLLERS” están los controladores de la capa de presentación.
- En la carpeta “MODELS” están los modelos de datos de la capa de base de datos. Cada uno de estos modelos representan objetos, los cuales, son guardados y recuperados de la base de datos y definen la estructura de los datos que la aplicación manejará.
- En la carpeta “REPOSITORIES” están los repositorios de la capa de persistencia.
- En la carpeta “SERVICES” están los servicios de la capa de negocio.

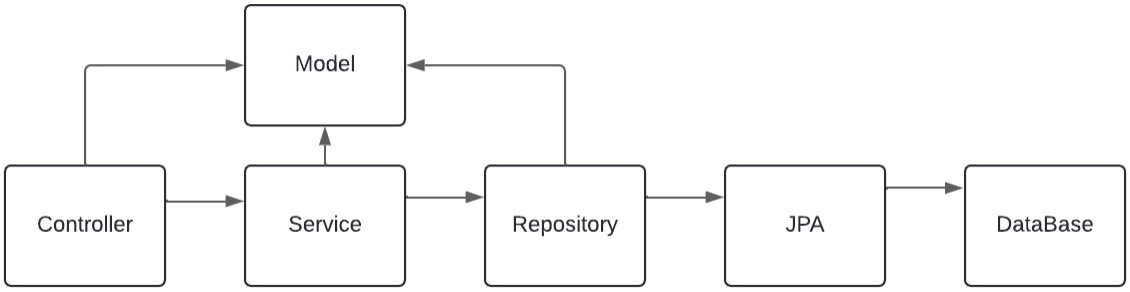

Figura 14. Diagrama de componentes

En la figura 14 se muestra un diagrama simplificado de las dependencias de los componentes de la aplicación. Este diagrama es genérico, de forma que por cada entidad de la aplicación hay una relación similar. Por ejemplo, para la entidad “Usuario” existen los componentes “UsuarioController”, “UuarioService”, “UsuarioRepository” y “Usuario” que representa el modelo.

4.3.2. El Diagrama de clases

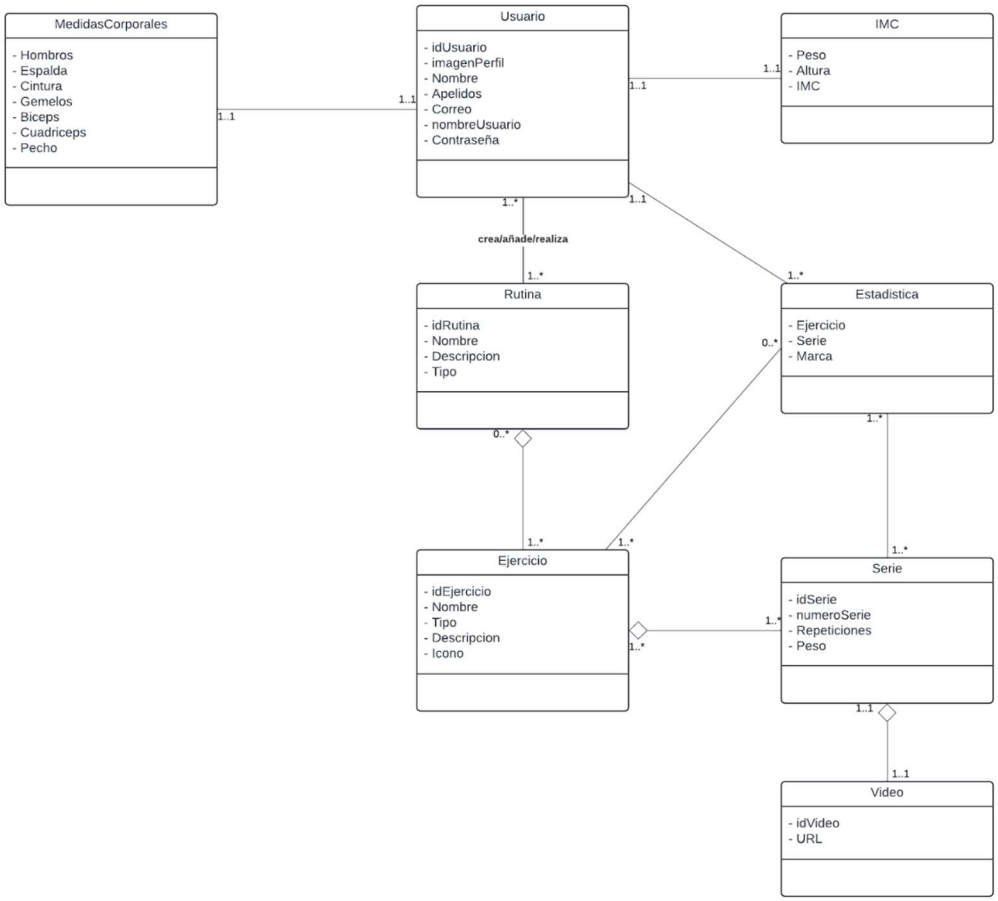

Figura 15. Diagrama de Clases inicial

El Diagrama de Clases resulta una herramienta visual útil para representar una estructura del sistema, mostrando las clases del sistema software y sus atributos. La figura 15 representa el diagrama de clases que se utilizó para empezar el desarrollo de la aplicación.

Las clases representan las entidades del sistema. La entidad Usuario representa las diferentes personas que utilizan la aplicación, en la que guarda información relevante del usuario. Cada usuario tiene medidas corporales, que simbolizan las medidas de los grupos principales musculares, y tiene un registro para calcular el IMC. Cada usuario tiene varias rutinas en la que guarda la información relevante a dicha rutina, como nombre, descripción o tipo de músculo que se trabaja en la rutina. Cada rutina tiene varios ejercicios, que simbolizan los distintos tipos de ejercicios que puedes realizar en los entrenamientos. Cada ejercicio está compuesto por Series que simbolizan la realización de un ejercicio en concreto y se guarda la cantidad de peso levantado, las repeticiones que se han realizado y además se puede asociar un video a la serie. Las estadísticas de un usuario, por su parte, se generan, a partir de los ejercicios y las series realizadas.

Su implementación ha presentado varios desafíos lógicos, por lo que ha ido evolucionando a medida que avanzaba el proyecto, llegando a una estructura mostrada en

la figura 16. En él representa el sistema de asociaciones de identidades complejo en que se basa *Social Lift*.

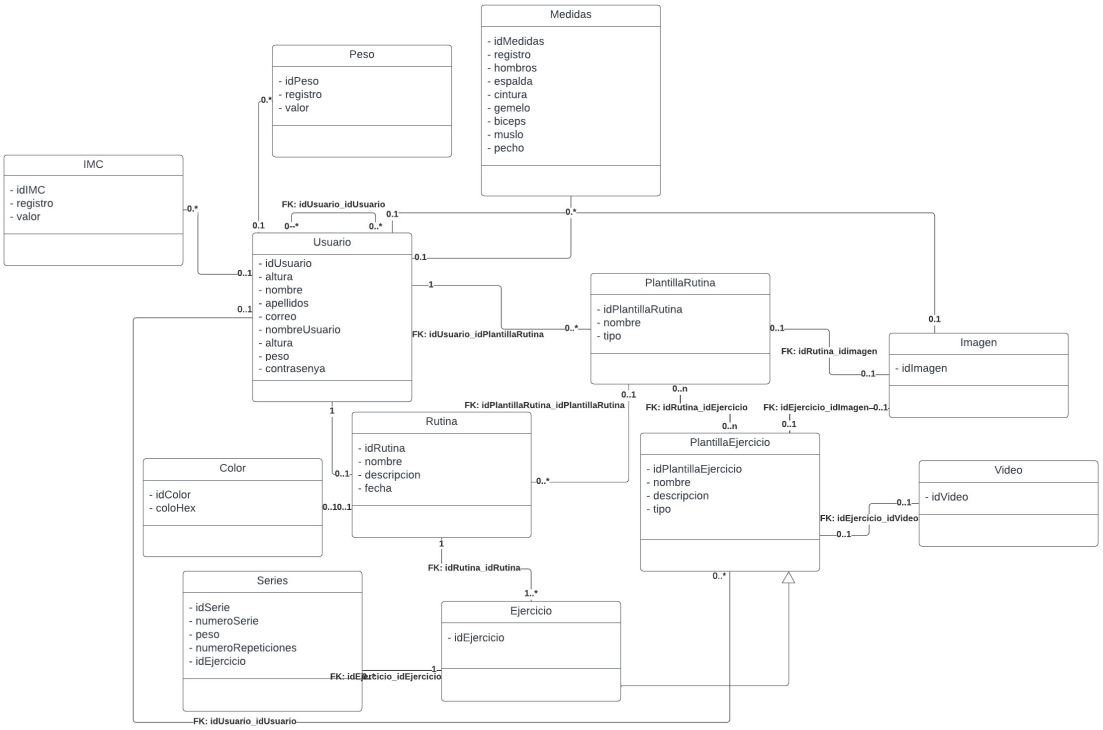

Figura 16. Diagrama de clases actual

Se han realizado varios cambios significativos como nuevas relaciones en las clases, por ejemplo, una relación recursiva en Usuario para simbolizar la lógica de seguimiento de usuarios. Se han creado nuevas clases como “Imagen” que gestiona todas las imágenes de la aplicación de manera aislada y las asocia en las clases que lo necesitan y otra clase “Peso” para tener seguimiento del cambio de peso a medida que pasa el tiempo. También se crearon nuevas entidades Plantillas que sirven para no tener que crear rutinas o ejercicios de 0 siempre que empieces un nuevo entrenamiento y una nueva entidad color que sirve para asociar un color a una rutina para visualizarla mejor en el calendario.

4.3.3. Documentación de la API

Figura 17. Documentación de la API.

En la figura 17, cada uno de los nombres representa el controlador de las distintas entidades de la aplicación.

Cada uno de estos controladores disponen de cierto número de Endpoints que permiten a los clientes realizar operaciones sobre recursos y situaciones específicas. Cada Endpoint está asociado a una URL única, compuesta por un enlace proporcionado por el servidor y la ruta específica del Endpoint. Es a través de estos que los clientes solicitan peticiones al *Backend*, enviando solicitudes HTTP y recibiendo archivos de respuesta JSON.

Swagger nos ayuda a realizar peticiones de manera sencilla, dando como entrada los datos necesarios para la petición según su tipo y representando en un cuadro de texto la respuesta del servidor.

En la figura 18 se muestra los Endpoints utilizados para lograr cumplir los requisitos funcionales (mayoritariamente) definidos anteriormente.

Figura 18. Endpoints de cada caso de uso

Hay ciertos Endpoints que sirven para gestionar ciertas herramientas utilizadas en el *Frontend*, como es el caso de los Endpoints del controlador “ColorController”, el cual dispone ayuda a la gestión de colores asociados a una rutina para mejorar el diseño del *Frontend*.

4.3.4. Base de Datos

Como base de datos se ha utilizado Supabase y se han creado las tablas necesarias para gestionar los datos de *Social Lift* mostradas en la figura 19.

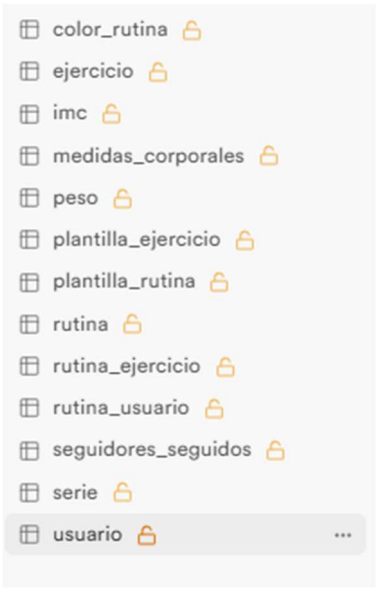

Figura 19. Tablas de la base de datos.

Gracias a Spring Boot no ha sido necesario crear explícitamente cada una de tablas que representan los datos y las relaciones, esto se debe a que Spring contiene anotaciones que simplifica el proceso sin necesidad de escribir el SQL explícitamente.

```
18 usages  Juan Pablo Bueno Fuentes +1
@Entity
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Serie {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long idSerie;

    @Column(nullable = false)
    private int numeroSerie;

    @Column(nullable = false)
    private double peso;

    @Column(nullable = false)
    private String tipoPeso;

    @Column(nullable = false)
    private int numeroRepeticiones;

    @ManyToOne
    @JoinColumn(name = "idEjercicio", nullable = false)
    @JsonIgnoreProperties("series")
    private Ejercicio ejercicio;

    private String video;
```

Figura 20. Entidad Serie

La figura 20 muestra el código de la entidad serie. En esta se pueden ver distintas anotaciones en amarillo. La anotación @Entity permite declarar la clase como entidad

de JPA, de forma que los atributos de la clase serán las columnas de las tablas de la base de datos. Las relaciones se crean gracias a las anotaciones @OneToMany, @ManyToOne, @ManyToMany y @OneToOne que mapean las asociaciones correspondientes en la base de datos

4.4. programación

4.4.1. Aplicación de patrones

Un patrón sirve como una posible solución probada y recomendada para un problema común en el mundo del desarrollo de software. Se basan en la experiencia acumulada por la comunidad de desarrollo de software y se considera como buena praxis para escribir código limpio.

Para el desarrollo del *Backend* de *Social Lift* se aplicaron varios patrones de diseño para mejorar la estructura, modularidad y mantenibilidad del código, que son los siguientes:

4.4.1.1. Patrón MVC (Modelo-vista-controlador).

El patrón de diseño Modelo-Vista-Controlador [11] es utilizado para separar la lógica de negocio, la presentación y el controlador de las soluciones. Facilita la gestión y la escalabilidad del código.

En el contexto de *Social Lift*, se ha implementado el patrón MVC para estructurar la API REST, en el que cada componente tiene un papel específico.

El controlador es el responsable de manejar las solicitudes HTTP que llegan a la API. Cuando una solicitud es recibida, el controlador procesa la entrada e invoca la lógica de negocio necesaria a través del modelo.

El Modelo representa los objetos de datos de la aplicación. Incluye las definiciones de las entidades y es la encargada de la manipulación de datos.

La vista es la representación de los recursos en formato JSON que son enviados al *Frontend* para su posterior uso en la aplicación. Se encarga de presentar los datos de manera estructurada y coherente, facilitando la renderización de la información correctamente.

4.4.1.2. Patrón Repositorio.

El patrón Repositorio [12] se utiliza para abstraer el acceso a la base de datos y proporcionar una capa de persistencia, facilitando la separación de la lógica de negocio con la base de datos y permitiendo un código más limpio.

El repositorio hereda sus funcionalidades de JPA el cual encapsula todas las operaciones CRUD necesarias. De esta manera, cualquier interacción con la base de datos se realiza mediante los métodos predefinidos de los repositorios y no es necesario crearlos explícitamente. Si queremos realizar una solicitud específica a la base de datos, JPA ofrece ciertas ayudas, de forma que solo es necesario definir el método con un nombre con palabras clave en el repositorio y JPA ya se encarga de realizar la petición, esto se puede ver en la figura 21 con los métodos “findByNombreUsuario” y “findByNombreUsuarioStartingWith”.

```
@Repository
public interface UsuarioRepository extends JpaRepository<Usuario, Long> {
    4 usages  Juan Pablo Bueno Fuentes
    public Usuario findByNombreUsuario(String nombreUsuario);
    4 usages  Juan Pablo Bueno Fuentes
    List<Usuario> findByNombreUsuarioStartingWith(String nombreUsuario);
}
```

Figura 21. Repositorio de usuario.

De esta forma, toda la persistencia de datos se encuentra desacoplada de la lógica de negocio, permitiendo la reutilización de código en diferentes partes de la aplicación y facilitando la realización de pruebas sobre la base de datos.

4.4.2. Refactorizaciones

Durante el desarrollo del *Backend* de *Social Lift*, se han realizado varias refactorizaciones con el objetivo de mejorar la optimización de los servicios y mejorar ciertos errores ocasionados en el proceso de desarrollo, de los cuales cabe destacar los siguientes.

Primero se identificó la necesidad de optimizar la respuesta de los archivos JSON, porque, a medida que la aplicación y los datos crecían, la información de estos archivos aumentaba y en muchas ocasiones traían información que no era relevante, varios objetos se repetían constantemente. Con el fin de mejorar la eficiencia, se implementó una estrategia de refactorización que incluyo la aplicación de la anotación @jsonIgnoreProperties para evitar la sobrecarga de datos innecesarios en las respuestas JSON.

Esta anotación permitía ignorar ciertas propiedades al serializar un objeto, lo que nos permite optimizar significativamente el rendimiento de las llamadas y las respuestas de la API. Para ello era necesario tener un gran conocimiento de la lógica de la aplicación, de manera que solo se implementara en atributos cuya información no es relevante en ciertas entidades.

En la figura 22 se muestra una comparación entra la captura de un usuario con la refactorización y la captura de un usuario sin la refactorización, de manera que los atributos como seguidores, pesos, IMC y medidas corporales no se capturan en el momento de capturar un usuario específico, ya que, si la información creciera en alguno de estos campos, sobre todo en el de seguidores, se propagaría un gran número de datos irrelevante por toda la aplicación. Para capturar estos atributos se utilizan Endpoints específicos.

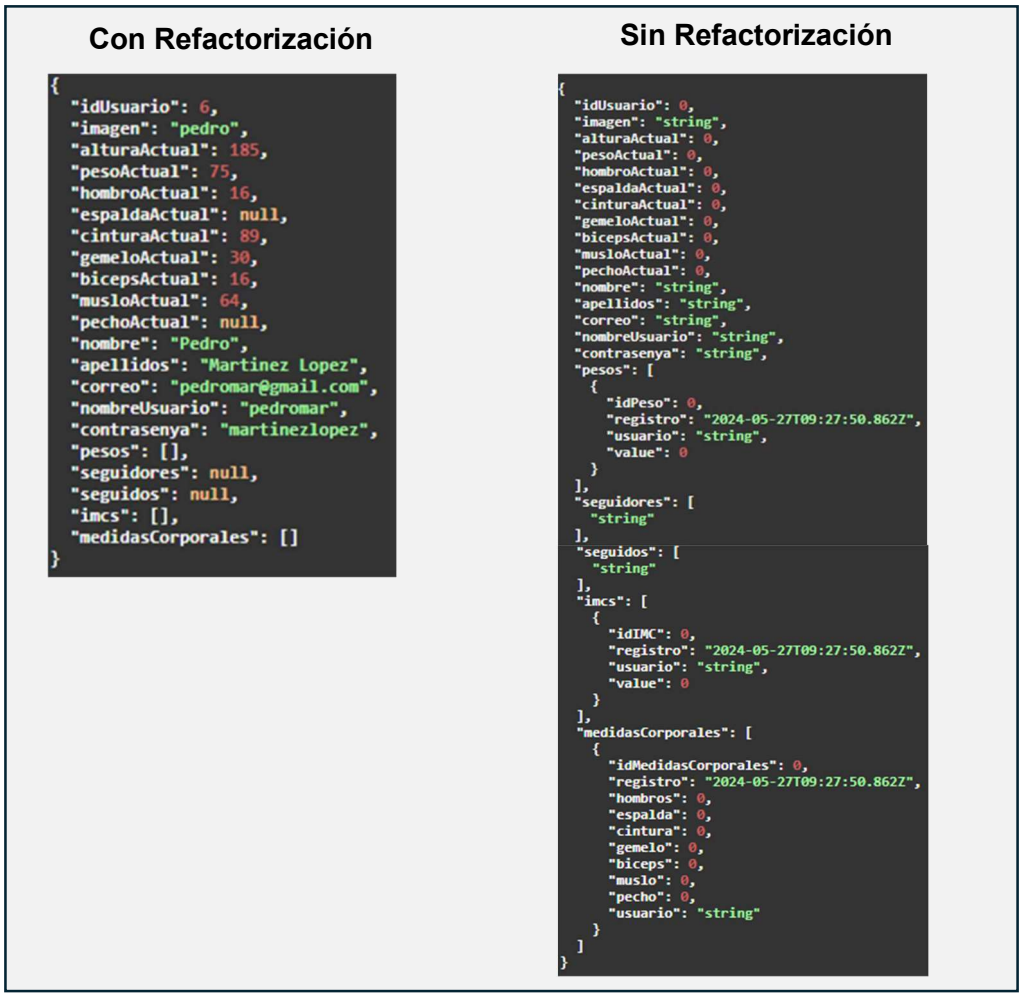

Figura 22. Refactorización Json

Se llevó a cabo una segunda refactorización con el objetivo de mejorar la obtención y envío de las estadísticas, debido a que dicha funcionalidad necesita de varios datos provenientes de varios campos de información lo que provocaba que su carga y respuesta fuese demasiado grande, por lo que se propuso hacer él envió con la información relevante organizada directamente desde el *Backend*. Para lograr esto, se implementó una estructura de datos más eficiente mediante el uso de mapas.

Figura 23. Respuesta de llamada de estadísticas.

La figura 23 muestra la respuesta de las estadísticas, los ejercicios de un usuario. En un mapa principal, la clave representa el ejercicio realizado, mientras que el valor asociado es otro mapa que contiene las fechas en que se ha realizado dicho ejercicio con su máximo peso registrado. Esta mejora simplifica el proceso de obtención de estadísticas y también agiliza la entrega devolviendo solo la información relevante mapeada directamente desde el *Backend*.

Aparte de estas dos refactorizaciones grandes, lo largo del proyecto se han realizado varios cambios con el fin de mejorar la eficiencia del servicio como pueden ser realizar Endpoints específicos para solicitudes concretas, como es el caso del buscador de personas que dispone de un Endpoint exclusivo para su servicio.

4.4.3. Despliegue

Para el despliegue de la aplicación, se ha optado por una solución basada en contenedores Docker y un servidor de alojamiento en la nube llamado Render. La aplicación se encuentra empaquetada garantizando su portabilidad. Se ha configurado un proceso automatizado que, al realizar un *Commit* en el repositorio *GitHub* en la rama de desarrollo, genera el archivo ejecutable .jar que contiene la aplicación, que posteriormente, es incorporada al contenedor Docker. Finalmente, el contenedor Docker se carga y ejecuta en el servidor alojado en Render, donde la aplicación hará que esté disponible para su uso público. Esta configuración proporciona un flujo de desarrollo ágil, permitiendo desplegar cambios y mejoras en la aplicación.

4.4.4. Desafíos de programación

El correcto desarrollo de la API REST para *Social Lift* conllevó ciertos desafíos y retos que tuvimos que superar.

Para empezar, la conexión correcta del *Backend* localmente con la aplicación móvil fue un inconveniente, debido a ciertos problemas de conexión de puertos. Esta dificultad llevó a buscar la solución de desplegar un servidor *online* para que así, el servicio de *Frontend*, siempre esté conectado a la dirección del servidor sin necesidad de modificar la dirección IP y los puertos. Esta solución también ha traído otras ventajas, por ejemplo, nos facilita el acceso a cualquier lugar, dándonos disponibilidad para poder usar siempre la aplicación móvil sin necesidad de tener una copia del *Backend* ejecutándose, al igual que nos permite comprobar que la aplicación se integra correctamente

con todos los cambios que se han realizado, ya que por cada despliegue se compila y se construye la aplicación, por lo que la versión actual en el servidor es una versión que se asegura que funciona correctamente.

Por otro lado, para el tema de inicio de sesión de la aplicación se buscó tener seguridad, por lo que se investigó un servicio proporcionado por auth0[26] para comprobar la autenticación de los usuarios y también darles autorización a realizar ciertas funcionalidades dependiendo de los roles que tengan. Esta solución presentó un reto muy grande de investigación que al principio mostró ciertos errores con los roles y la autenticación debido a que no lo realizaba correctamente. Pero después de varios intentos, se consiguió que los Endpoints solo sean visibles por los usuarios dependiendo de la llamada del Endpoint. Si el URL de la llamada contiene por "*public*" es accesible por cualquier usuario de la aplicación, si contiene por "*admin*" la función solo es posible si la realiza el administrador y si contiene "Client" la función es posible para usuarios registrados de la aplicación. Los roles son verificados mediante un Token, el cual representa una cadena de caracteres que se verifica a sí misma y contiene cierta información, en este caso, la información es el rol del usuario. El *Backend* solo acepta llamadas provenientes del *Frontend* en el cual se espera una petición a un Endpoint con sus respectivos datos de entrada y el Token de autorización.

Esta función no está del todo implementada en la aplicación porque necesita más pruebas para validar su correcto funcionamiento, por lo que su desarrollo se dejó en una rama aparte de la de desarrollo subida en el servidor.

4.5. Pruebas

Las pruebas en el desarrollo software representan un papel fundamental, por lo que para comprobar la fiabilidad y el rendimiento de la aplicación se realizó un proceso de pruebas y así validar el correcto funcionamiento del *Backend*.

4.5.1. Pruebas unitarias

Las pruebas unitarias nos sirven para verificar que cada unidad individual de código funcione correctamente de manera aislada, por lo que se centran en probar unidades de código de forma independiente. De esta manera podemos detectar errores de forma temprana y mejorar la calidad de código.

Para implementar las pruebas de la aplicación se utiliza las tecnologías Junit y Mockito. Cada una de las clases del *Backend* tiene una clase correspondiente de prueba que comprueba cada uno de los métodos de dicha clase.

Para los modelos se prueban los métodos *getters* y *setters*, constructores y la adición y eliminación de elementos en atributos que son de tipo listas.

Para los servicios se prueba que cada uno de los métodos cumple con su respectivo propósito y se mockea la llamada a los repositorios, de forma que se prueben explícitamente las funciones que se realizan en el servicio y no sea por un problema externo.

Para los repositorios se comprueban los métodos que hacen llamadas a la base de datos, por lo que se prueban las acciones CRUD con la base de datos sin dejar

[26] https://auth0.com/

constancia. También se prueban métodos personalizados creados para mostrar su correcto funcionamiento.

Figura 24. Test unitario del repositorio de usuario.

En la figura 24 se muestra una prueba sobre el repositorio “UsuarioRepository” en el que se comprueba el correcto funcionamiento del método “testFindByNombreUsuarioStartingWith”. Para empezar se realiza un setUp() en el que se crea un usuario y se guarda en la base de datos con el nombre de usuario ”usuario_prueba” luego se ejecuta el test en el que se pasa el String “usuario_prueba” y se ejecuta el método. El test tendría que salir correcto si captura el usuario. Al final de las pruebas se elimina el usuario creado en el setUp para no modificar la base de datos.

Por último, para los controladores se comprueba que las peticiones HTTP funcionan correctamente, simulando los servicios, de forma que solo se comprueba que las llamadas y las respuestas son correctas.

Estas pruebas se realizaron sobre las 12 entidades de la aplicación, probando cada una de las clases descritas anteriormente.

Las pruebas unitarias ayudan a prevenir errores y fallos en la lógica individual de cada componente, por lo que estas pruebas son ejecutadas en total mediante la línea de comandos “mvn test” al finalizar cada Sprint. Las pruebas se ejecutaban manualmente y si todas pasaban correctamente se podía preparar el siguiente Sprint. Esto ayudo mucho en el mantenimiento de la aplicación.

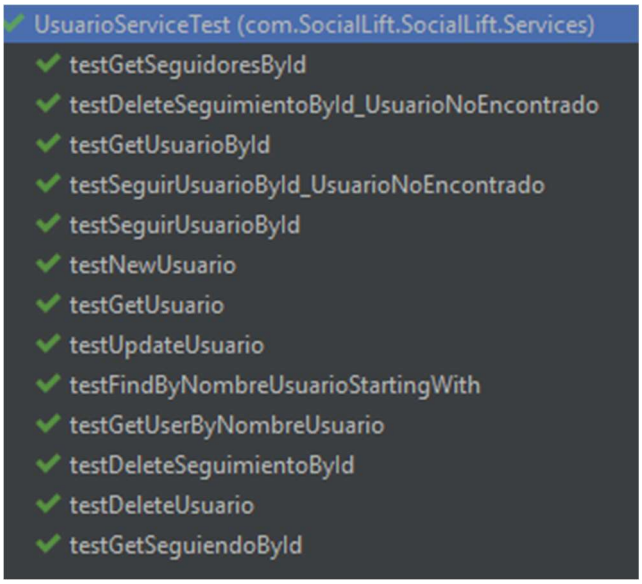

Figura 25. Resultado de pruebas de UsuarioService

En la figura 25 se muestra el resultado de las pruebas unitarias realizadas sobre la clase usuarioService que se realización correctamente.

4.5.2. Pruebas de integración

Para las pruebas de integración se hicieron *tests* sobres los distintos Endpoints utilizados en la API REST, realizando sus respectivas llamadas y comprobando su correcto funcionamiento.

En la figura 26 y 27 se aprecia el código de las pruebas de la entidad Usuario sobre el Endpoint que crea un usuario. En cada Endpoint se prueba el caso correcto y los casos que darían que darían una respuesta incorrecta y se comprueba su respuesta esperada. Este proceso se ha repetido en los demás Endpoints de la aplicación

```
@Test
public void testNewUsuario() throws Exception {
    Usuario usuario = new Usuario();
    usuario.setNombre("Juan");
    usuario.setApellidos("Pérez");
    usuario.setCorreo("juan@example.com");
    usuario.setNombreUsuario("juanperez");
    usuario.setContrasenya("password");

    mockMvc.perform(post( urlTemplate: "/api/usuario")
                    .contentType(MediaType.APPLICATION_JSON)
                    .content(objectMapper.writeValueAsString(usuario)))
            .andExpect(status().isOk());

    Usuario createdUsuario = usuarioService.GetUserByNombreUsuario(usuario.getNombreUsuario());
    assert createdUsuario.getNombreUsuario() == usuario.getNombreUsuario();
}
```

Figura 26. Test de integración newUser

```java
@Test
public void testNewUsuarioName() throws Exception {
    Usuario usuario = new Usuario();
    usuario.setApellidos("Pérez");
    usuario.setCorreo("juan@example.com");
    usuario.setNombreUsuario("juanperez");
    usuario.setContrasenya("password");

    mockMvc.perform(post( urlTemplate: "/api/usuario")
                    .contentType(MediaType.APPLICATION_JSON)
                    .content(objectMapper.writeValueAsString(usuario)))
            .andExpect(status().isOk());

    Usuario createdUsuario = usuarioService.GetUserByNombreUsuario(usuario.getNombreUsuario());
    assert createdUsuario == null;
}
```

Figura 27. test integración newUser incorrecto por nombre

Para asegurar que en el momento de finalizar las pruebas la base de datos no presente ningún cambio, Spring Boot nos ofrece una anotación denominada @Transactional, la cual, realiza un *Rollback* en la base de datos al finalizar las pruebas, es decir, realiza las operaciones pertinentes para ejecutar las pruebas y cuando termina deshace todas las operaciones.

Estas pruebas se realizaron sobre los 12 controladores de la aplicación, probando cada uno de los Endpoints.

Estas pruebas nos ayudan a prevenir errores de funcionalidades, por lo que se usaron como pruebas de regresión en el momento de entrega de un MVP, de esta manera se probaban todas las funcionalidades creadas hasta el momento, asegurando que cada una seguía teniendo un funcionamiento correcto. Estas pruebas, al igual que las otras, se ejecutan mediante la línea de comando “mvn test”.

Cabe destacar la posibilidad de automatizar las pruebas, por ejemplo, en el momento de realizar un Commit y despliegue de la aplicación. Para ello se puede utilizar herramientas como Jenkins[27] y así ofrecer prácticas de desarrollo software útiles como Integración Continua (CI). Este aspecto se implementará en un futuro.

4.5.3. Pruebas de rendimiento

Para las pruebas de rendimiento se utilizó una herramienta de prueba de carga basada en la nube llamada Loader.io, la cual simula grandes volúmenes de tráfico en sus aplicaciones web para evaluar su capacidad de manejo.

Antes de empezar con estas pruebas es necesario aclarar que el servidor que se está usando en Render tiene unas prestaciones muy bajas, debido a que se está utilizando el servicio gratuito, las prestaciones se muestran en la figura 28.

[27] https://www.jenkins.io/

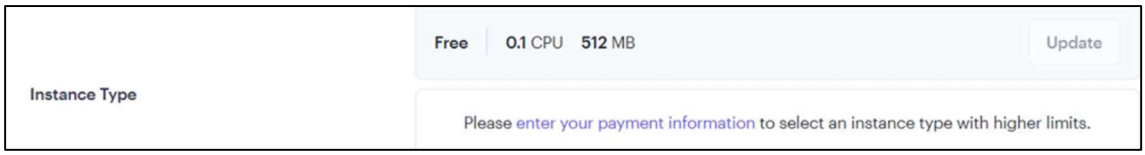

Figura 28. Prestaciones servidor Render.

Esto afectará considerablemente al número de personas que puedan estar usando la aplicación a la vez.

Las pruebas consisten en aportar el URL de un Endpoint de la aplicación y Loader.io realizar un cierto número de peticiones con diferentes parámetros personalizados.

Para empezar, se realizó la prueba en dos Endpoints que da como respuesta un archivo de datos muy grande, los cuales son:

- El Endpoint que captura todos los ejercicios de la aplicación
- El Endpoint que captura todas las rutinas de la aplicación.

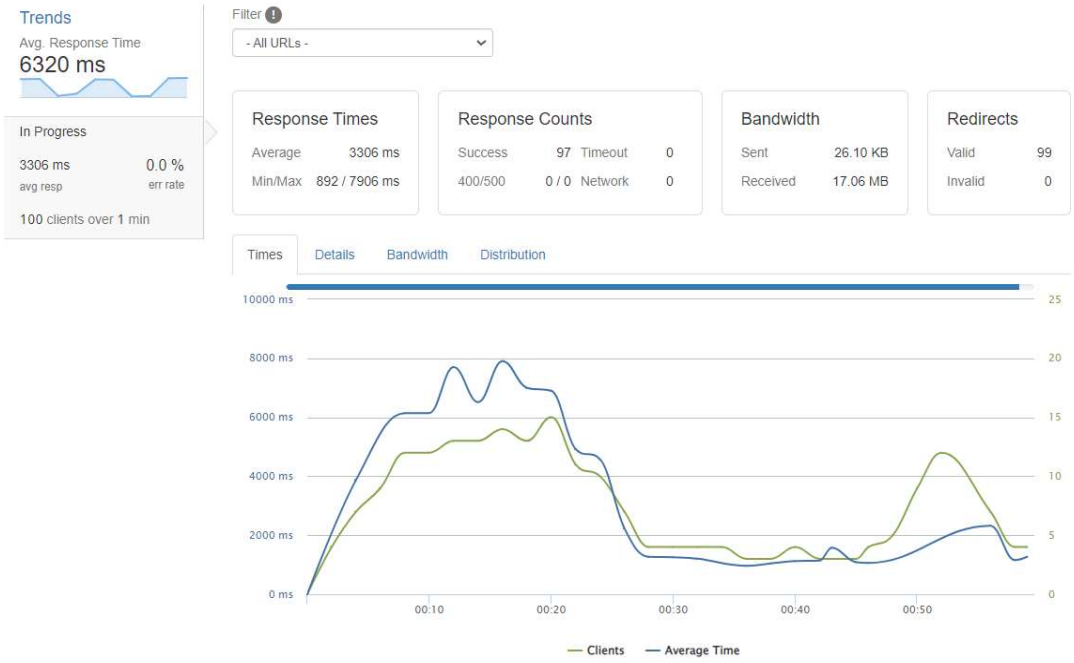

Figura 29. Prueba para la captura de ejercicios.

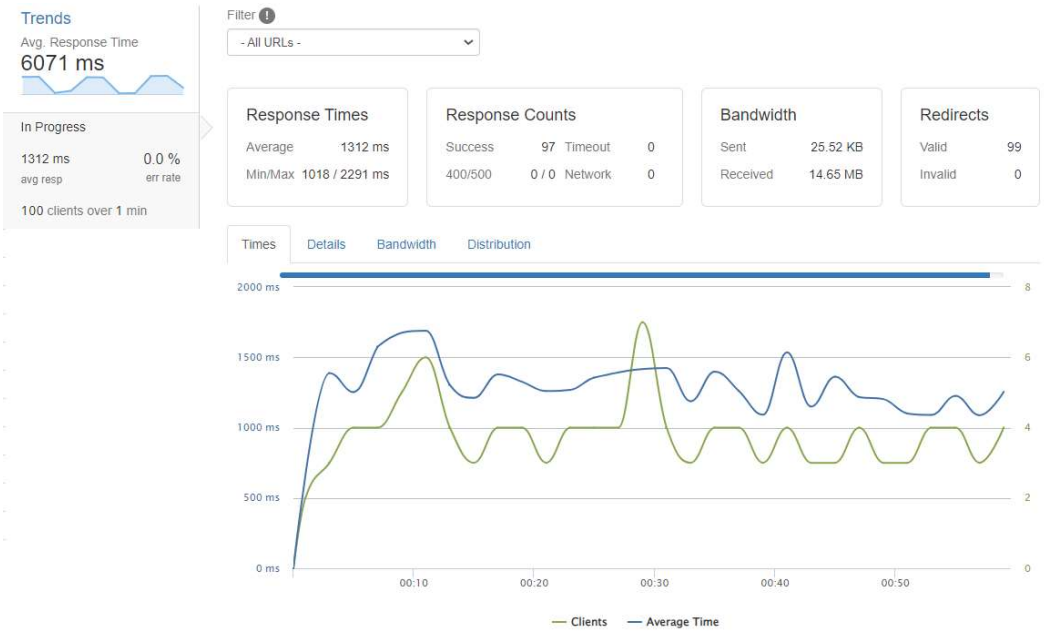

Figura 30. Prueba para la captura de rutinas.

La prueba consiste en que se simula el acceso de 100 clientes a dicha llamada por un minuto. La línea verde en la figura 29 y 30 representa el número de clientes en una franja de tiempo y la línea azul el tiempo medio en milisegundo que tarda la respuesta para dicho número de clientes en su respectiva franja horaria, como se puede observar las dos pruebas pasaron correctamente con un tiempo de media de los ejercicios de 6320 ms y 6071 ms para las rutinas.

El pico de clientes consecutivos fue de 16 para los ejercicios y 7 para las rutinas. Ahora vamos a elevar esos picos enviándoles siempre el mismo número de usuarios en un minuto. Esta prueba solo se va a realizar sobre la captura de los ejercicios, ya que es la que más media de tiempo tuvo.

25 clientes consecutivos

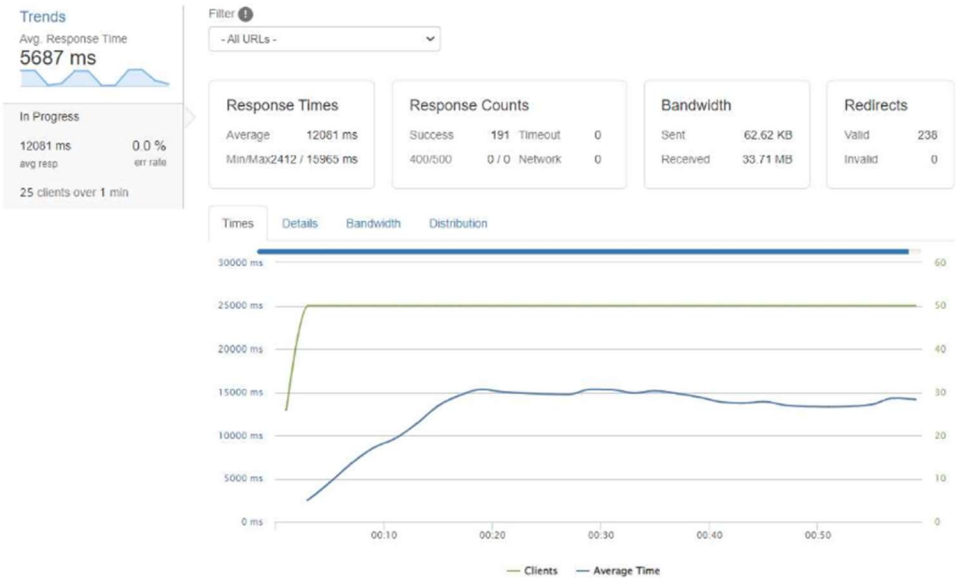

Figura 31. Prueba para la captura de ejercicios con 25 clientes.

Con 25 clientes, en la figura 31, se obtiene una media de 5687 ms y no presenta ningún error en las respuestas por lo que se puede considerar correcto.

50 clientes consecutivos

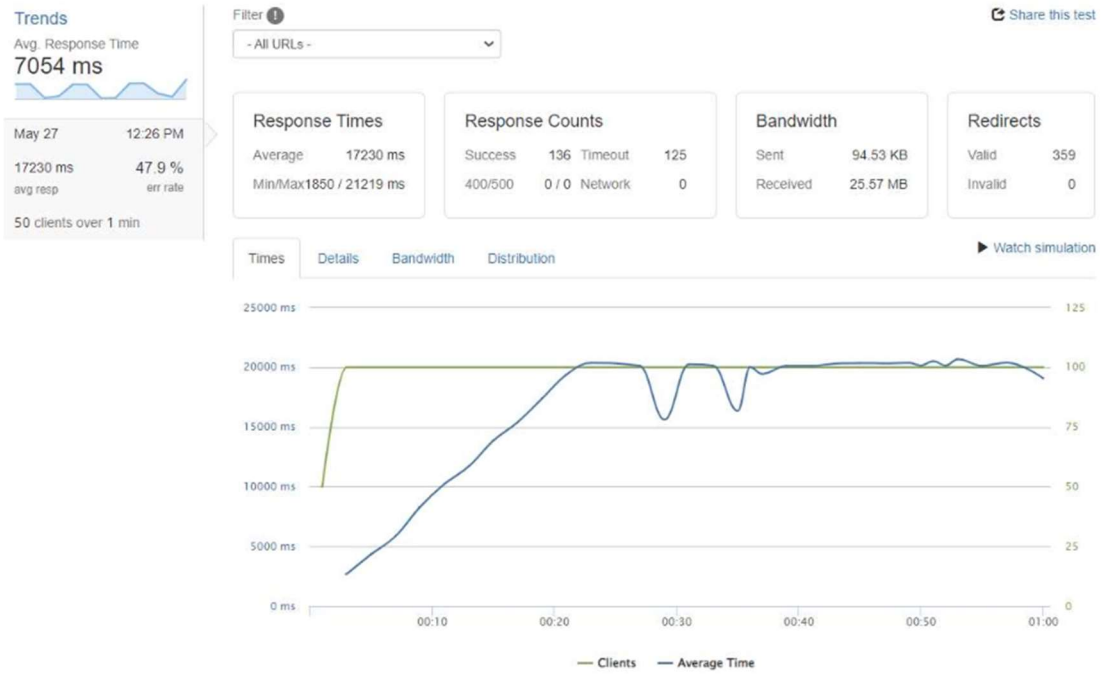

Figura 32. Prueba para la captura de ejercicios con 50 clientes.

Con 50 clientes, en la figura 32, la media es de 7042 ms, pero el porcentaje de error ha subido a un 47.9%. Este porcentaje simboliza la cantidad de peticiones que se dan como fallidas porque presentan un *timeout* de más de 10 segundos. Esta situación ya se podría considerar crítica para la aplicación, por lo que es necesario realizar un planteamiento de mejora, ya sea mejorando las presentaciones del servidor o realizar una lógica más eficiente en la respuesta de la información.

Para terminar, se realizó una prueba en sobre una funcionalidad interesante de la aplicación que es la captura de las estadísticas, con el fin de comprobar si la refactorización fue efectiva o, por el contrario, es necesario realizar otra refactorización más eficiente, dicha prueba se muestra en la figura 33.

50 clientes consecutivos

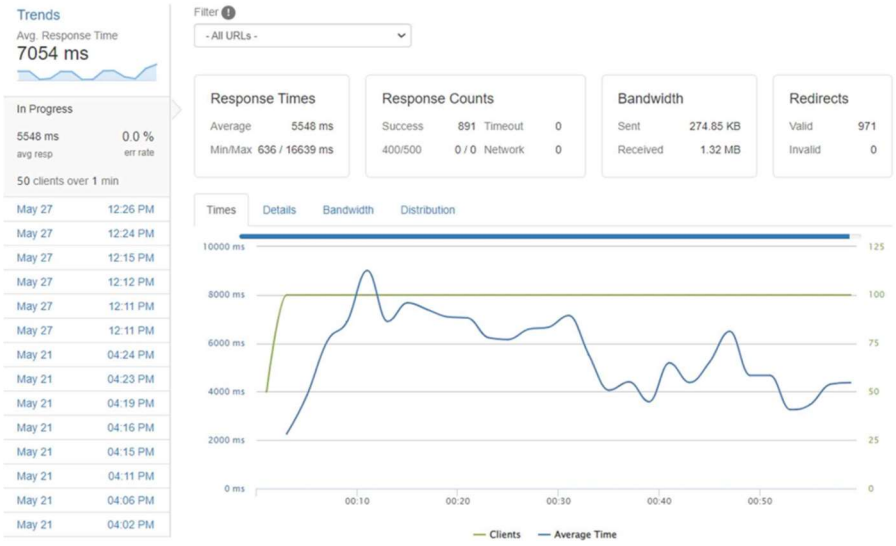

Figura 33. Prueba para la captura de estadísticas con 50 clientes.

En la prueba se obtuvo un tiempo medio de respuesta de 7054 ms que es el mismo tiempo de respuesta medio obtenido sobre los ejercicios, sin embargo, no ha presentado ningún error por lo que el servicio se ha mantenido constante.

100 clientes consecutivos

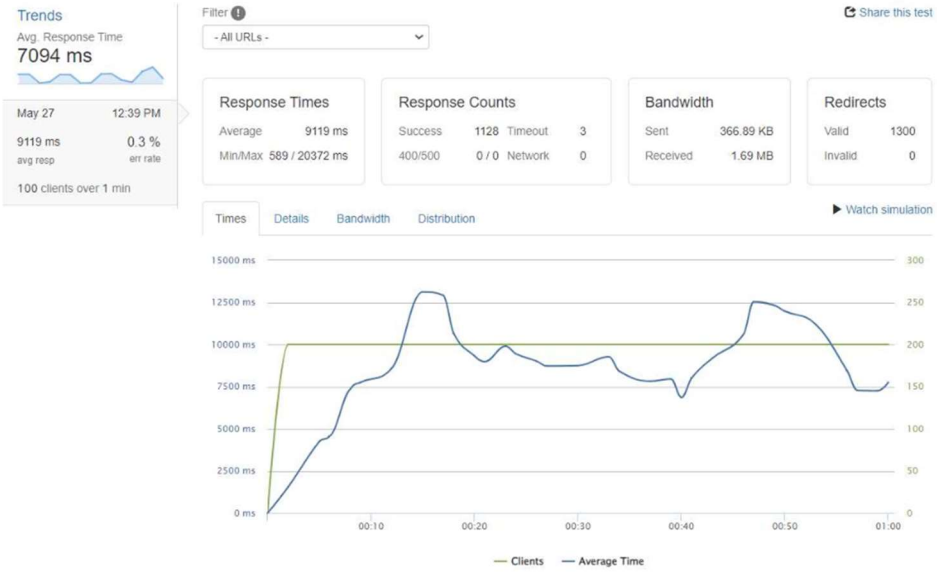

Figura 34. Prueba para la captura de estadísticas con 100 clientes.

Para 100 clientes consecutivos, en la figura 34, se han obtenido muy buenos resultado, con un tiempo de respuesta media similar al obtenido con 50 clientes consecutivos y con un porcentaje de error muy bajo, por lo que podemos afirmar que la refactorización ha sido un éxito.

4.5.4. Pruebas de aceptación.

Las pruebas de aceptación son realizadas para determinar si un sistema cumple o no con los requisitos especificados, lo cual es crucial en el proceso de desarrollo software.

Se han desarrollado pruebas de aceptación para cada una de las unidades de trabajo del proyecto siguiendo el esquema mostrado en la figura 35.

Calendario de rutinas	
Jesús Fierrez	Se mostrará un calendario, en el que en el dia correspondiente, si el usuario tiene una rutina asignada, debe aparecer en ese dia
22-03-24	

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

Figura 35. Prueba de aceptación.

En el momento de crear una tarea, se definen las pruebas necesarias que esa tarea debe cumplir. Cuando una tarea termine su proceso de desarrollo de programación, pasará al proceso de pruebas en las que se evaluará cada una de las pruebas propuestas, teniendo como resultado un “OK” si la prueba pasó correctamente o un “FAIL” si no cumple con los requisitos acordados.

En el caso de que contenga un “FAIL”, la tarea pasará otra vez al proceso de desarrollo para arreglar su funcionamiento, de manera que, al momento en el que se vuelvan a evaluar las pruebas, pase correctamente.

Cuando todas las pruebas se evalúen como “OK” se podrá dar como finalizada la tarea.

5. Cronología del TFG

Tras definir los casos de uso de *Social Lift* se han creado épicas en Jira, mostradas en la figura 36, una por cada caso de uso más una de flecos y mejoras.

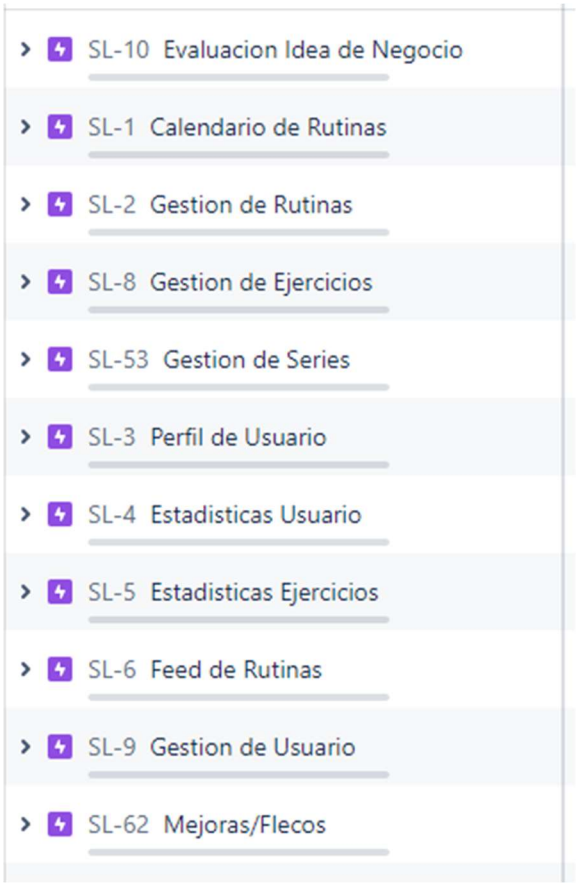

Figura 36. Épicas de Jira.

A partir de estas épicas, se iban creando dentro de ellas unidades de trabajos para desarrollar incrementalmente cada una.

5.1. Backlog Inicial

Se comenzó definiendo un Backlog inicial, el cual sirve como punta de partida para la planificación de nuestro proyecto. Este Backlog inicial comprende una lista de funcionalidades y unidades de trabajo globales del proyecto.

Cada elemento del Backlog fue evaluado y clasificado según su importancia relativa para los objetivos del proyecto, de forma que cada tarea tiene una prioridad asignada que permite enfocar el esfuerzo en las tareas más críticas e importantes de la aplicación.

Figura 37. Backlog inicial.

Cada una de estas unidades de trabajo tiene asociado una épica, la cual se ilustra mediante un texto con un color de relleno en la figura 37.

Jira nos ayuda a priorizar las unidades de trabajo asignando un campo de prioridad en el que se escoge entre 4 diferentes opciones: Muy alta, alta, baja, muy baja. Dependiendo de qué prioridad se asigne, te ordena el Backlog, mostrando las unidades de trabajo con más alta prioridad primero.

Como criterio de priorización se han tenido en cuenta las siguientes medidas:

- **Highest**: Unidades de trabajo que son completamente necesarias para cumplir los requisitos básicos de la aplicación o unidades de trabajo que su resolución no depende de otras tareas pero que otras UT sí dependen de su resolución.
- **High**: Unidades de trabajo necesarias para cumplir con los requisitos básicos relacionadas con la prioridad highest que dependen de que otras unidades de trabajo ya estén realizadas.
- **Medium**: Unidades de trabajo interesantes a realizar, en la aplicación pero que no comprenden como requisito fundamental básico de la aplicación.
- **Low**: Unidades de trabajo interesantes a realizar pero que en un principio no entran dentro del alcance del TFG.

Cada una de estas UT están asignadas a una unidad de trabajo más grande denominada “Épica”, la cual representa cada una de las características esenciales de la aplicación.

5.2. Línea temporal.

Figura 38. cronología del proyecto.

La línea temporal de la figura 38 nos ofrece una visión general de la evolución del proyecto. Dividida en acontecimientos importantes del proyecto.

En el periodo de formación primero se investigó por encima sobre distintas tecnologías para montar una API REST las cuales fueron NestJS[28], ASP.NET Core[29] y Spring Boot, siendo este último el que se seleccionó por su amplia variedad de herramientas automatizadas y por el lenguaje Java el cual se ha utilizado con recurrencia en el grado de ingeniería informática. A partir de videos informativos y cursos online se obtuvo la formación necesaria para empezar el proyecto y a medida que avanzaba se iba investigando más.

Otras ciertas tecnologías como Supabase, Github o Render ya se habían utilizado anteriormente en otros proyectos del grado, por lo que no fue necesario formarnos en dichas tecnologías

i. Sprint 0 – Inicio del proyecto

Durante el Sprint cero, se dedicó a realizar la evaluación de la idea de negocio del proyecto para examinar su viabilidad y mercado potencial. Además, se dedicó tiempo a investigar sobre las tecnologías que se utilizaron a lo largo del proyecto. Esto implica evaluar las diferentes opciones disponibles, sus ventajas y desventajas. Al igual que seleccionar las herramientas y librerías que mejor se adaptarán a nuestras necesidades.

[28] https://nestjs.com/
[29] https://dotnet.microsoft.com/es-es/learn/aspnet/what-is-aspnet-core

ii. Desarrollo del primer MVP y primer experimento

Para el desarrollo del primer MVP nos enfocamos en implementar las características más esenciales de la aplicación, asegurando su funcionalidad básica para satisfacer las necesidades fundamentales de los usuarios. Para lograr esto, se llevó a cabo un proceso de programación que abarco dos Sprints de tres semanas de duración cada uno. Cada Sprint contiene las respectivas unidades de trabajo que se seleccionaron dependiendo de su priorización del Backlog. Los Sprints 1 y 2 se representan en la figura 39 y 40 respectivamente

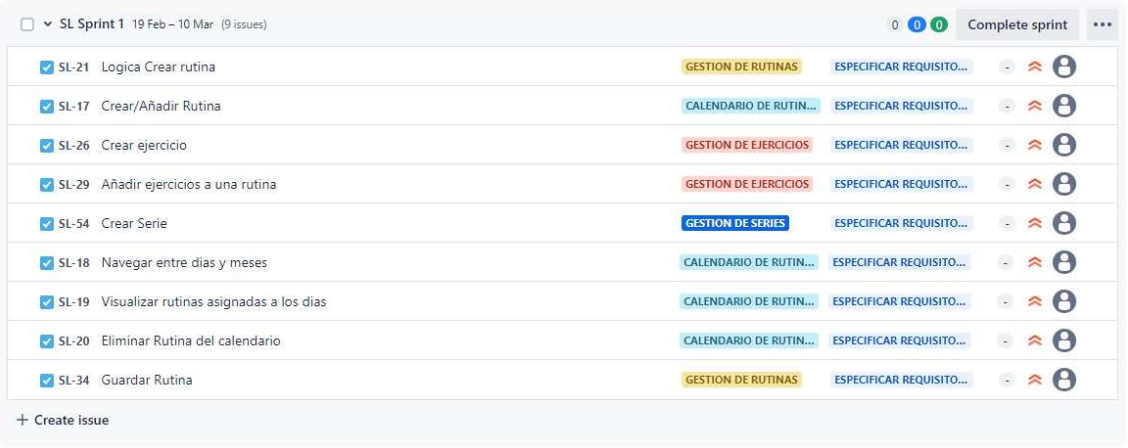

Figura 39. Planificación Sprint 1.

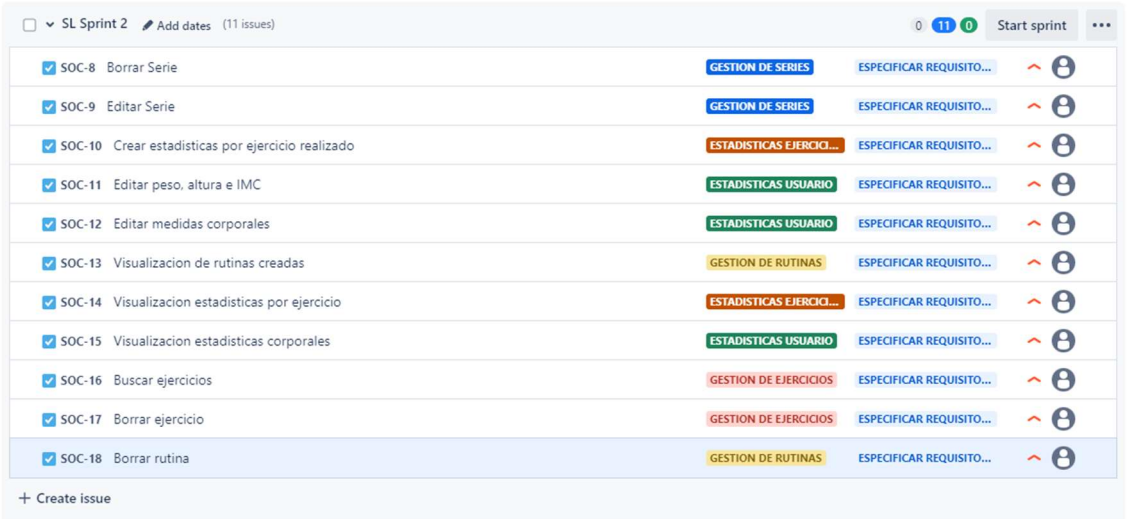

Figura 40. Planificación Sprint 2.

Cada una de las tareas sigue el *Workflow* anteriormente comentado, con el propósito de que al finalizar el Sprint estén todas las tareas en *Listo*.

Al finalizar cada Sprint se realiza una reunión de retrospectiva para proponer distintas mejoras en los siguientes Sprints, en las cuales cabe destacar:

- Mejorar la buena práctica de la metodología, como la asignación de responsables de las tareas al momento de realizarlas y así tener una mejor gestión.
- Realizar pruebas de aceptación más exhaustivas y completas.

Tras la finalización de estos 2 Sprints, se entregó el primer MVP y se llevó a cabo el primer experimento que consistía en la recopilación de información por medio de una encuesta dirigida a usuarios que asisten constantemente a gimnasios.

La encuesta se divide en 8 secciones. La primera sección para comprobar que son usuarios potenciales para el experimento con preguntas sobre su experiencia en el gimnasio, las 6 secciones siguientes relacionadas con las interfaces más relevantes de la aplicación y en las cuales se realizan preguntas para validar los requisitos de cada interfaz. Por último, la última sección corresponde a preguntas finales sobre la aplicación en general.

Sección 1

Preguntas

Las preguntas realizadas en la sección 1 a los usuarios potenciales son las siguientes:

- ¿Con qué frecuencia realizas entrenamientos de gimnasio?
- ¿Cuál es tu fuente de información para añadir nuevos ejercicios en tus rutinas?
- ¿Dónde apuntas tu progreso diario en los diferentes ejercicios de tus rutinas?
- ¿Conoces alguna aplicación útil para gestionar tus entrenamientos de gimnasio?
- Si tu respuesta a la anterior pregunta fue Sí, indícanos qué aplicación conoces.

Resultados

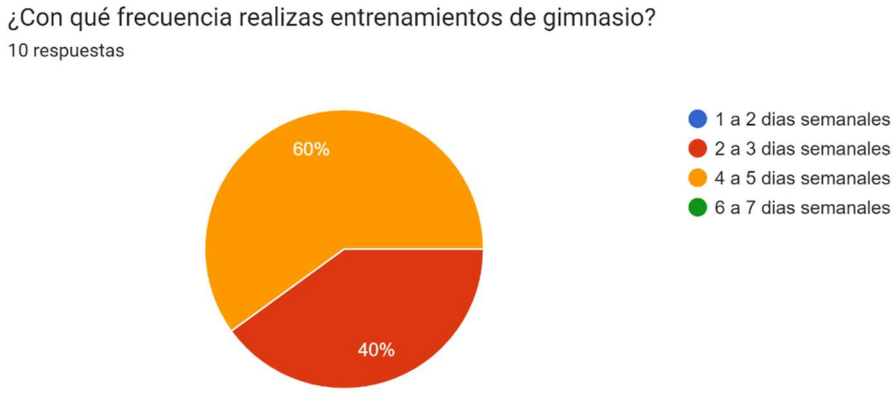

Figura 41. Resultado frecuencia de entrenamientos.

Una forma habitual de contabilizar la frecuencia con que la gente va al gimnasio es hacerlo por semana. Ir de 1 a 2 días es una frecuencia un poco insuficiente, así que buscamos personas que vayan entre 3 a 5 días. Gracias a la encuesta podemos observar en la figura 41 que todos cumplen este rango, y que la mayoría con un 60% asisten al gimnasio entre 4 a 5 días por semana.

Figura 42. Resultado fuentes de información.

Figura 43. Resultado apunte de progreso.

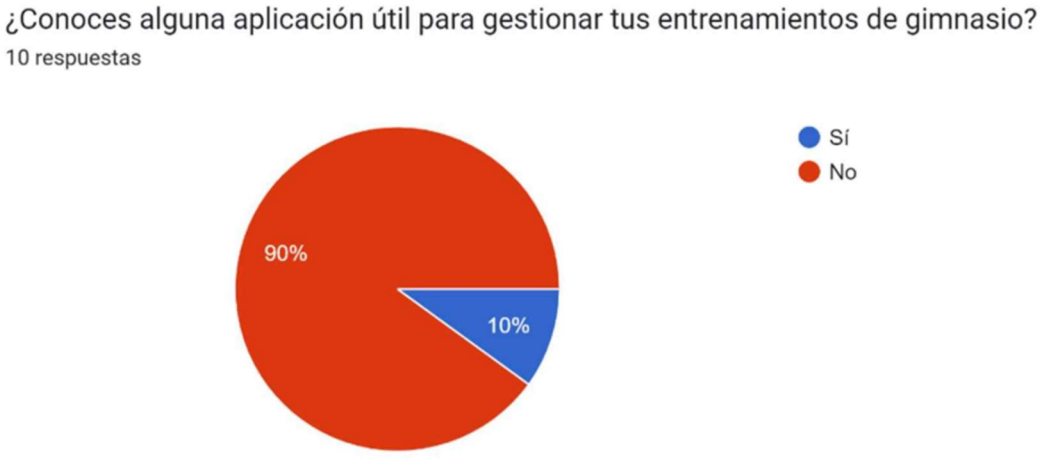

Figura 44. Resultado aplicación.

En el proyecto se busca ofrecer un servicio único y nuevo a estos usuarios, por lo que nos interesa saber dónde recopila la información y apuntan su progreso del gimnasio. En la figura 42 se muestra que mayoritariamente se informan a partir de internet,

asimismo, no recopilan información por medio de una aplicación móvil. Por otro lado, Se puede comprobar en la figura 43 que la mayoría de la gente con un 60% del total no apunta su progreso, por lo que nos presenta una oportunidad para mostrar nuestro servicio. El otro 40% sí controla su seguimiento. En la figura 44, Un 10% del total conoce otra aplicación de gestión de entrenamientos, que gracias a la última pregunta de esta sección podemos saber que la aplicación es “Hevy”, la cual es un principal competidor.

Sección 2 a 6

Preguntas

En cada una de las secciones se realizaron las mismas preguntas para 6 distintas interfaces. Las preguntas son:

- ¿Qué impresión tienes de la pantalla?
- ¿Te resulta fácil entender cómo utilizar las funcionalidades de esta pantalla?
- ¿Consideras que la información presentada en la pantalla es clara y concisa?
- ¿Qué funcionalidades te gustaría ver añadidas a esta pantalla en futuras versiones de la aplicación?

Se evaluaron la interfaz del calendario principal, la interfaz de creación de una rutina, la interfaz de la selección de ejercicios, la interfaz de las rutinas creadas y la interfaz del perfil de usuario.

Resultados

La impresión de las pantallas de las interfaces fue mayoritariamente positiva, obteniendo resultados con una media general de todas las interfaces de 4,5 sobre 5.

Figura 45. Resultado interfaz selección de ejercicios.

En la figura 45 se muestra que el nivel de compresión de las pantallas es ligeramente inferior, con una media de 4,46 sobre 5 en general. La pantalla que ha resultado más difícil de comprender ha sido la de la interfaz de selección de ejercicios con una media de 4,3 sobre 5 que igualmente tiene un valor muy positivo.

Gracias a esta información podemos decir que los requisitos se validaron correctamente y que la aplicación va cumpliendo su objetivo. Respecto a la última pregunta de esta sección, se crearon nuevos requisitos que se comentaran en la sección 7.

Sección 7

Preguntas

En esta sección se realizaron preguntas generales sobre la aplicación, que son las siguientes:

- ¿En qué aspectos crees que la aplicación podría mejorar para satisfacer mejor tus necesidades de gestión de rutinas de entrenamiento en el gimnasio?
- ¿Qué funcionalidades adicionales te gustaría ver implementadas en futuras versiones de la aplicación?
- ¿Qué otras aplicaciones o herramientas similares has utilizado anteriormente y qué características de ellas crees que podrían ser útiles para integrar en Social Lift?
- ¿Cómo calificarías, en una escala del 1 al 10, tu nivel de satisfacción general con el estilo de las pantallas de la aplicación?
- ¿Tienes alguna sugerencia adicional o comentario final que te gustaría compartir para mejorar la aplicación?

Gracias a estas preguntas, los usuarios aportaron nuevas posibles funcionalidades que consideran interesantes para implementar en la aplicación, de las cuales caben destacar las siguientes:

- La aparición de la duración de la rutina de entrenamiento.
- La posibilidad de subir videos en los ejercicios para su visualización posterior.
- La implementación de un *feed* en el que se puedan publicar rutinas de diferentes usuarios.
- Verificar usuarios.
- Añadir el gimnasio en el que se ha realizado la rutina.
- Opciones predeterminadas en el tipo de rutina y de ejercicios.

Estas nuevas funcionalidades son ahora nuevos posibles requisitos de la aplicación para un futuro desarrollo.

Por último, podemos decir que en general los resultados obtenidos a partir de la encuesta son bastante positivos, con una media de satisfacción general de 8,1 sobre 10, validando nuestros requisitos iniciales y destacando áreas de mejora.

iii. Desarrollo del segundo MVP y segundo experimento

Para el segundo MVP se propuso, por un lado, implementar los nuevos requisitos propuestos en el anterior experimento y por el otro lado, desarrollar ciertas tareas priorizadas que se consideran relevantes en el desarrollo de la aplicación y que entraban en el alcance de los Sprints.

Este segundo MVP se realizó en dos Sprints de tres semanas de duración, igual que el anterior.

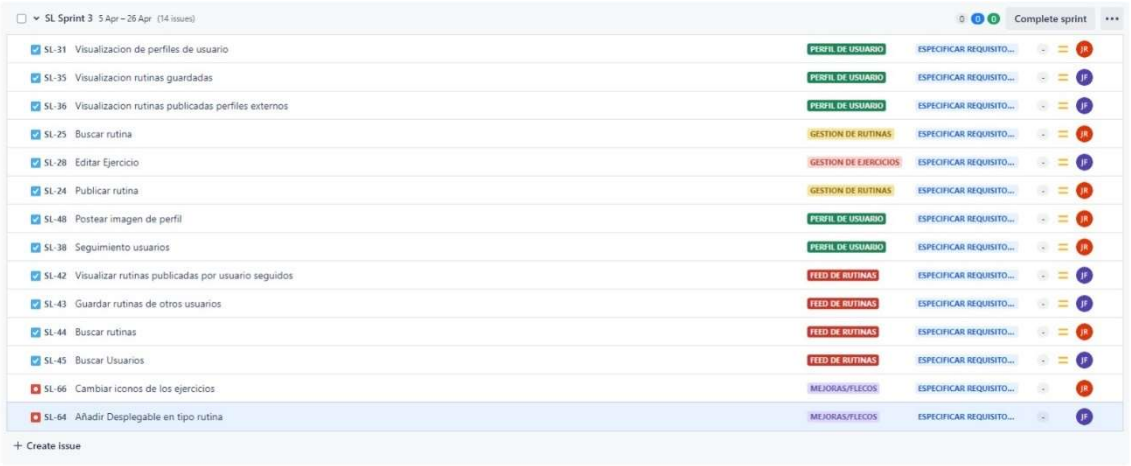

Figura 46. Planificación Sprint 3.

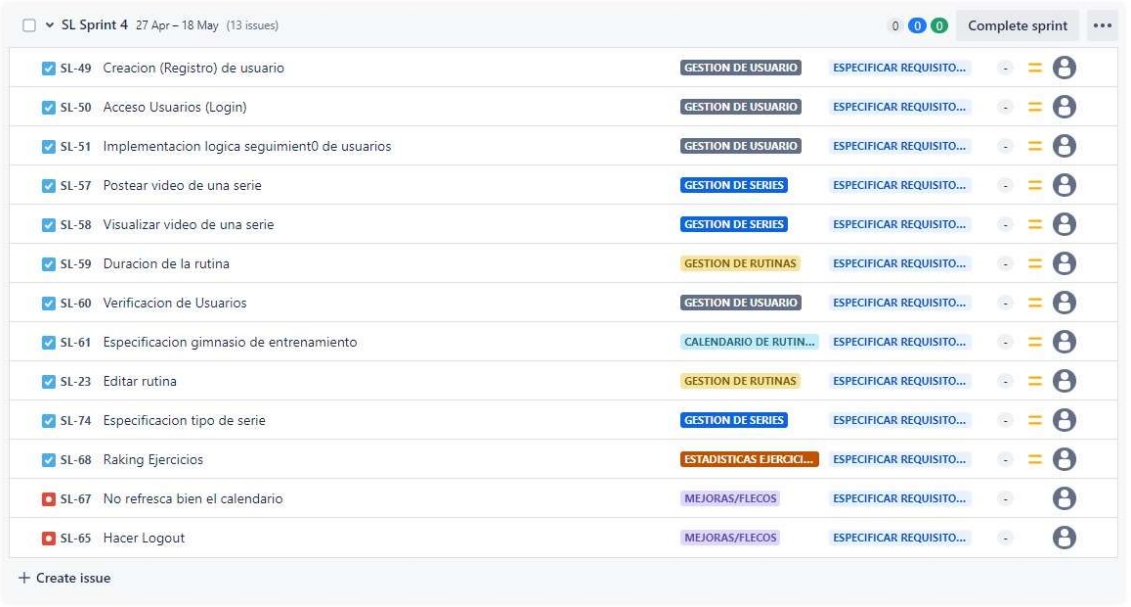

Figura 47. Planificación Sprint 4.

En estos Sprints entraron las tareas que tenían mayor prioridad del Backlog, los nuevos requisitos propuestos a partir del primer experimento y la corrección de ciertos bugs en la aplicación. En la figura 46 y 47 se muestran los Sprints 3 y 4 respectivamente.

Al finalizar cada Sprint se realizaron reuniones de retrospectiva para mejorar el funcionamiento de la metodología, de las cuales cabe destacar:

- Mejorar el proceso de desarrollo y de pruebas de una tarea, de forma que no pase tanto tiempo en un estado sin revisión.

Al finalizar los dos Sprints se realizó la entrega del segundo MVP, el cual contenía una aplicación sólida con ciertas funcionalidades capaces de competir en el mercado. Con la entrega de este MVP se realizó el segundo experimento.

Este segundo experimento consiste en una encuesta que responderán los mismos usuarios que participaron en el primer experimento y otros nuevos usuarios principiantes y avanzados en el mundo del *fitness*.

Esta encuesta está divida en dos secciones. La primera sección encargada de la validación de los nuevos requisitos implementados, en la que se realizan preguntas para comprobar si las nuevas funcionalidades cumplen o no su función. La segunda sección consiste en una demo en la que se les proporciona a los encuestados la aplicación desarrollada para que puedan probar las distintas funcionalidades. Después, los usuarios tienen que contestar un cierto número de preguntas sobre la aplicación y el servicio aportado, de forma que se pueda evaluar el *Frontend* y el *Backend* de la aplicación. En este TFG nos centraremos en la evaluación del *Backend*.

Sección 1

Preguntas

En esta sección se repite una pregunta sobre 5 nuevas funcionalidades, la pregunta es la siguiente:

- ¿Crees que la implementación de este nuevo requisito se adecua a las necesidades del usuario?

Las funcionalidades implementadas son:

- La aparición de la duración de la rutina de entrenamiento y opciones predeterminadas en el tipo de rutina y de ejercicios.
- La posibilidad de subir videos en los ejercicios para su visualización posterior.
- La implementación de un *feed* en el que se puedan publicar rutinas de diferentes usuarios.
- Verificar usuarios.
- Añadir el gimnasio en el que se ha realizado la rutina.

Resultados

Gracias a la encuesta, podemos validar correctamente las nuevas funcionalidades,

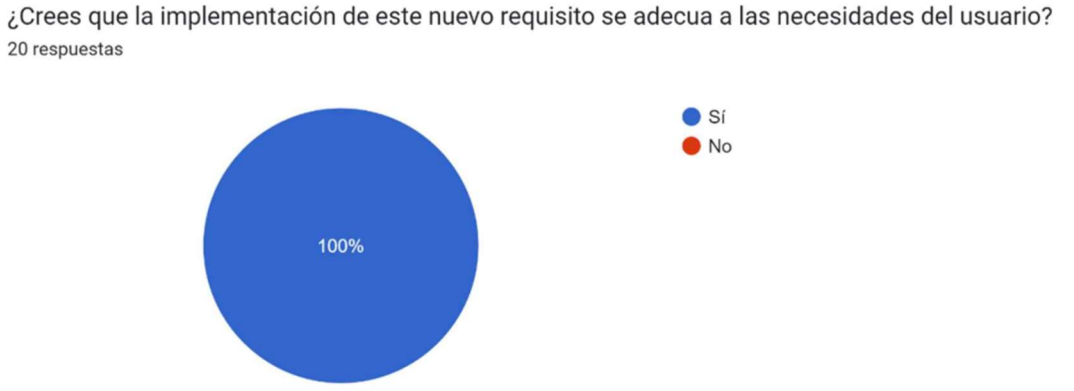

Figura 48. resultado validación de requisito 1.

Se ha obtenido un resultado de “Sí” con un 100% de verificación, mostrado en la figura 48, en los requisitos:

- La aparición de la duración de la rutina de entrenamiento y Opciones predeterminadas en el tipo de rutina y de ejercicios
- La posibilidad de subir videos en los ejercicios para su visualización posterior
- La implementación de un feed en el que se puedan publicar rutinas de diferentes usuarios

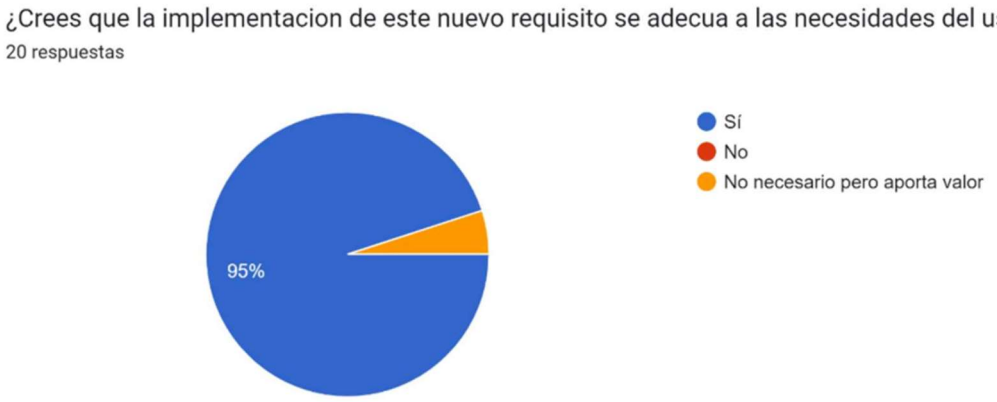

Figura 49. Resultado validación de requisito 2.

Mientras que se ha obtenido un 95% con la respuesta “Sí”, mostrado en la figura 49, en los requisitos:

- Verificar usuarios
- Añadir el gimnasio en el que se ha realizado la rutina

Debido a que el 5% restante considera que estos requisitos se califican como: “No necesario, pero aporta valor”

Sección 2

Preguntas

En esta sección se realizaron preguntas sobre la aplicación en general. Para este TFG destacaremos las siguientes preguntas:

- ¿Has experimentado interrupciones o caídas en el servicio mientras usabas la aplicación?
- ¿Cuánto tiempo suele tardar la aplicación en responder a tus acciones (por ejemplo, al abrir una página o al realizar una búsqueda)?
- ¿Cuánto tiempo suele tardar la aplicación en responder a tus acciones (por ejemplo, al abrir una página o al realizar una búsqueda)?
- ¿Cómo calificarías la estabilidad general del servicio durante tu uso de la aplicación?

Por otro lado, también había preguntas para buscar nuevas funcionalidades o posibles mejoras, al igual que recibir realimentación (*feedback*) y distintas opiniones sobre la aplicación:

- ¿Qué aspectos, tras el uso de la aplicación, destacarías (tanto buenos como malos)? ¿Y qué crees que deberíamos tener en cuenta para futuras versiones de la aplicación?

- ¿Cómo evaluarías la variedad y utilidad de las funcionalidades ofrecidas por la aplicación en general?
- ¿Qué tan satisfecho/a estás con la diversidad de características y opciones que ofrece la aplicación para satisfacer tus necesidades?

Resultados

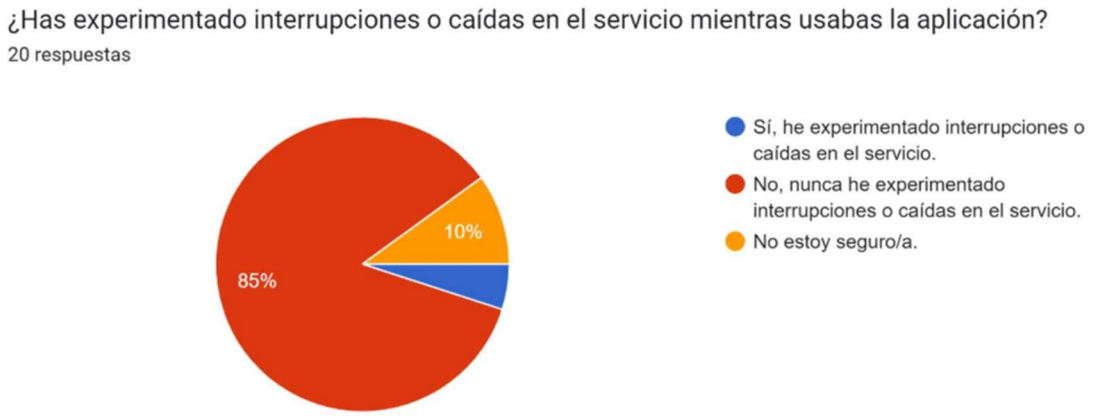

Figura 50. Resultado interrupciones.

Figura 51. Resultado tiempo de respuesta.

Un requisito fundamental para el *Backend* es que el servicio ofrezca disponibilidad constante y tiempos de respuesta cortos, por lo que resulta muy importante que el servicio no tenga ningún tipo de interrupción o caída en el momento en el que un usuario use la aplicación.

Como se puede observar en la figura 50, el 85% de los usuarios que usaron la aplicación no presentaron ningún tipo de caída, por lo que es un resultado muy positivo. Además, tan solo el 5% de los usuarios puede afirmar que presentó una interrupción en el momento de uso de la aplicación.

Para los tiempos de respuesta, al ser un servidor ajeno a nuestro control total, buscamos una franja de tiempo entre 0 y 3 segundos de respuesta. En la figura 51, el 90% de los usuarios encuestados respondieron en dicho rango. El otro 10% no presentó los resultados esperados.

Figura 52. Resultado errores de datos.

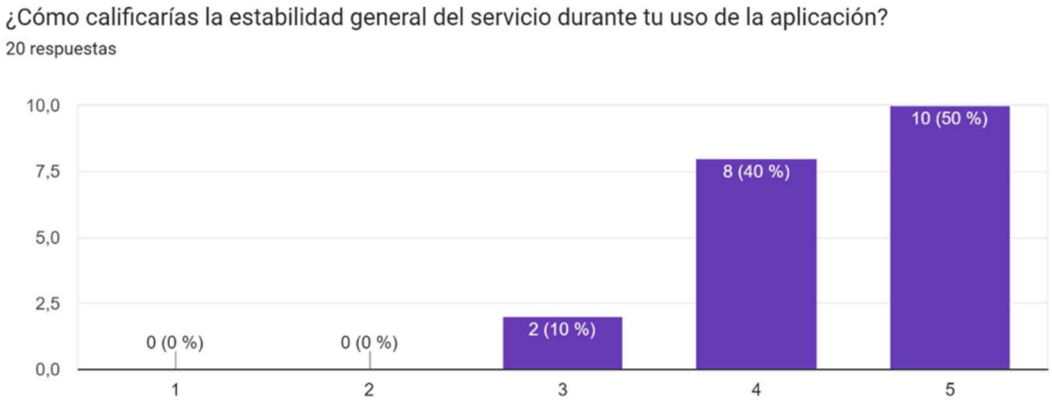

Figura 53. Resultado estabilidad.

Por otro lado, para el *Backend* se busca que el servicio aportado no disponga de ningún error en la lógica de funcionamiento y que los datos mostrados siempre sean los esperados.

En la figura 52, tan solo un 5% de los encuestados presentó un error de datos en el uso de la aplicación, el otro 95% tuvo los resultados esperados. Por último, la estabilidad del producto, mostrado en la figura 53, obtuvo una calificación de 4,4 sobre 5, por lo que se puede considerar que el *Backend* de *Social Lift* ofrece un servicio estable.

Como conclusión, se obtiene que el servicio de la API de *social Lift* por el momento es el adecuado, pero no es perfecto. El ideal sería reducir los tiempos de espera de respuesta del servidor y buscar una disponibilidad casi perfecta. Como se ha visto en las pruebas de rendimiento, es necesario mejorar el servidor en el que se encuentra alojada la API REST y así sea una posible solución a los tiempos de carga y de respuesta de la aplicación.

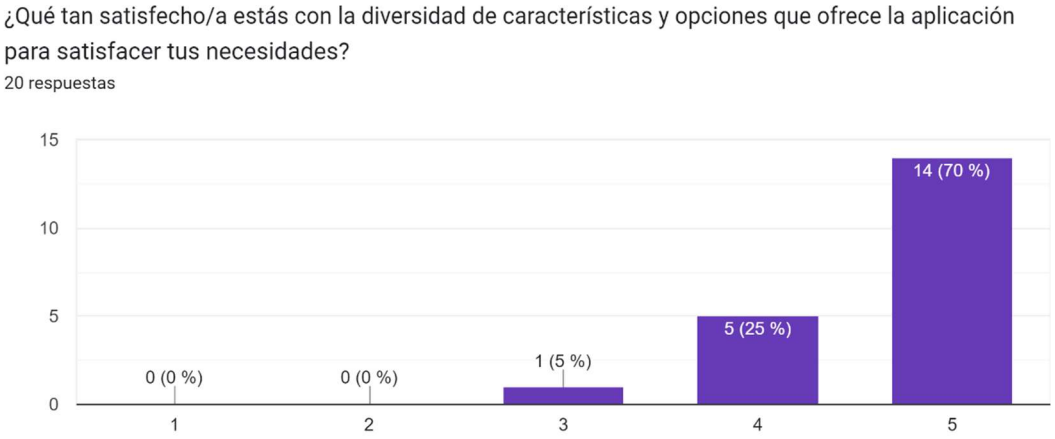

Figura 54. Resultado satisfacción.

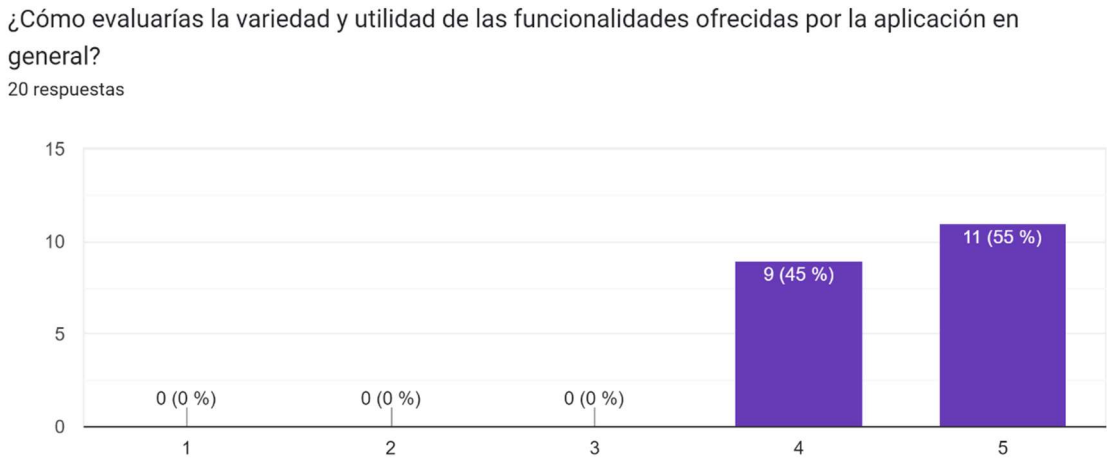

Figura 55. Resultado utilidad de las funcionalidades.

Por último, se puede ver que los usuarios evaluaron positivamente la aplicación. Con un 4,65 sobre 5 de media en la satisfacción de la diversidad de características en la figura 54 y con un 4,55 sobre 5 en la utilidad de las funcionalidades en la figura 55.

Los usuarios también aportaron nuevas mejoras interesantes como:

- “Un aspecto a mejorar, que no significa que ahora mismo sea malo, es la parte de las estadísticas de los ejercicios, estaría bien añadir más información, acerca de los PR o RM”
- “Me gustaría que de cara a futuras versiones se pudiera realizar un ranking filtrando por gimnasio o ejercicio”.

Y destacaron el uso del calendario y el *feed* de publicaciones.

5.3. Dedicación

Para visualizar la dedicación aproximada que se tuvo para realizar este TFG se muestra la siguiente tabla 6.

Actividad	Marco de trabajo	Fecha inicio	Fecha Fin	Tiempo invertido
Formación e investigación		22/01/2024	04/02/2024	32 h
Evaluación de la idea de negocio *	Sprint 0	04/02/202	17/02/2024	26 h
Desarrollo del primer MVP	Sprint 1	19/02/204	09/03/2024	52 h
	Sprint 2	11/03/2024	01/04/2024	52 h
Primer experimento y análisis *	Experimento 1	02/04/2024	05/04/2024	6 h
Desarrollo del segundo MVP	Sprint 3	05/04/2024	26/04/2024	49 h
	Sprint 4	27/04/2024	18/05/2024	49 h
Segundo experimento y análisis *	Experimento 2	19/05/2024	21/05/2024	6 h
Memoria				80 h
Total				346

Tabla 6. Dedicación

Cabe resaltar que las actividades marcas con un asterisco (*) son actividades que se han realizado en conjunto con mi compañero Jesús Fiérrez Ruipérez.

6. Conclusiones y trabajo futuro.

El proyecto de S*ocial Lift* nace con la idea de solucionar la falta de herramientas y funcionalidades para tener seguimiento en los entrenamientos. Esta aplicación se presenta como solución que facilita la organización y gestión de rutinas, fomentando la colaboración entre distintos usuarios de la aplicación.

La evaluación de *Social Lift* como idea de negocio reveló que es un producto innovador y le permitirá posicionarse favorablemente para atraer gran audiencia y establecer colaboraciones estratégicas con más campañas o empresas de este sector.

Los experimentos realizados con *early adopters* revelaron datos positivos para el proyecto, como la aceptación general de la idea y de las funcionalidades proporcionadas por *Social Lift*, dando también *feedback* y nuevas mejoras que enriquecen a la aplicación y demuestran compromiso con los usuarios.

Por parte del desarrollo del *Backend*, se han conseguido los objetivos. Se logró desarrollar un servicio que sustente el *Frontend* de la aplicación de manera eficiente, el servicio está disponible para varios dispositivos móviles y no presenta caídas, además la gestión de rutinas, ejercicios, series y usuarios se realizan correctamente sin fallos aparentes como se han demostrado en los experimentos. Se realizaron experimentos en un entorno real con usuarios potenciales. El objetivo de desarrollar un producto que sustente una gran cantidad de usuarios es mejorable, como se ha mencionado en el apartado de pruebas, las refactorizaciones han cumplido su función, sin embargo, es indispensable aumentar las prestaciones del servidor en el que está alojada la API. El desarrollo de este proyecto ha aportado una base para futuros proyectos de emprendimiento, dando una gran experiencia tanto en la gestión de proyectos como en las tecnologías utilizadas, ampliando mis conocimientos en nuevos lenguajes y marcos de trabajo.

Aparte de los conocimientos obtenidos a lo largo del proyecto, también se han utilizado conocimientos de diferentes ámbitos adquiridos a lo largo de la carrera, como entre los cuales destacan las siguientes:

- La gestión de procesos y proyectos software: PIN y PSW
- Formación en el desarrollo software con código limpio y eficiente: DDS e ISW
- Proceso de análisis de requisitos necesarios en un proyecto: DDM y AER
- Bases de programación en java y conocimiento de bases de datos: IIP, PRG y BDA.

Después de todo el trabajo realizado, considero que he logrado conseguir mis objetivos personales gracias al desarrollo del proyecto. Mi experiencia tanto en tecnologías para desarrollar una API REST como en la puesta en marcha de un *Backend* ha aumentado, he mejorado la práctica de metodología Scrum, el trabajo en equipo y la gestión de tiempo en un proyecto de emprendimiento.

La realización del TFG es simplemente el inicio de un proyecto que le espera un continuo desarrollo para seguir mejorando la aplicación y así conseguir su consolidación en el mercado.

Por otro lado, Aunque los experimentos a lo largo del TFG fueron realizados por usuarios experimentados en gimnasios, se busca conseguir una gran cantidad de usuarios para incentivar a las personas a seguir una vida *fitness* y así conseguir una comunidad grande en *Social Lift*. Se pretende lograr este objetivo a través de campañas de marketing y promociones para captar a un mayor público y hacer sostenible este proyecto.

De igual manera, se busca optimizar el rendimiento general de la aplicación, asegurando tiempos de respuestas cortos y así asegurar una experiencia de usuario fluida y sin errores.

Actualmente, la aplicación no se encuentra en ningún mercado de aplicaciones móviles, para ello es necesario mejorar ciertos aspectos de seguridad y añadir sistemas de pago para conseguir los ingresos previstos en el modelo de negocio.

Social Lift está en la búsqueda activa de colaboradores e inversores que compartan la visión de promover el estilo de vida *fitness* a través de la tecnología para así acelerar su crecimiento, mejorar su infraestructura y expandir nuestras campañas de marketing.

Aún hay varias funcionalidades descritas en la tabla comparativa del estudio de mercado que faltan por implementar y que se encuentran en el Backlog, al igual que, es necesario resolver los constantes bugs e incidencias que se presenten en la aplicación, resulta interesante también implementar pruebas automatizadas como se mencionó en su respectivo apartado. Este inicio presenta una base sólida para un proyecto innovador con alto potencial para competir en el mercado.

Referencias

[1] Informe global de la IHRSA 2022. Disponible en: https://es.ihrsa.org/publications/the-2022-ihrsa-global-report/, consultada en 02/2024.

[2] Statista. (s.f.). El sector del fitness en España. Disponible en https://es.statista.com/temas/6047/el-sector-del-fitness-en-espana/, consultada en 04/2024.

[3] Valgo. Evolución y perspectivas del sector fitness: del desafío de la pandemia al renacer económico. Disponible en: https://www.valgo.es/blog/evolucion-y-perspectivas-del-sector-fitness-del-desafio-de-la-pandemia-al-renacer-economico?elem=315524, consultada en 03/2024.

[4] Instituto de Formación Profesional. (2023). Análisis DAFO: Qué es y para qué sirve. Disponible en https://www.ifp.es/blog/analisis-dafo-que-es-para-que-sirve, consultado en 02/2024.

[5] Maurya, A. (n.d.). What is Lean Canvas?. Disponible en https://leanstack.com/lean-canvas, Consultada 03/2024.

[6] Spring Boot. "Spring Boot". Disponible en: https://spring.io/projects/spring-boot#learn, consultada en 03/2024.

[7] Ries, E. (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business. consultada en 02/2024,

[8] Atlassian. (n.d.). What is a minimum viable product (MVP). Disponible en: https://www.atlassian.com/agile/product-management/minimum-viable-product, consultada en 03/2024.

[9] Scrum.org. (2022). The Scrum Guide. Disponible en: https://www.scrum.org/resources/scrum-guide , consultada en 02/2024.

[10] Reactive Programming. (n.d.). Capas: Estilos arquitectónicos. Disponible en https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/capas, consultada en 04/2024

[11] CampusMVP. (2023). ¿Qué es el patrón MVC en programación y por qué es útil? Disponible en https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx, consultado el 03/2024.

[12] Our Academy. (2023). El patrón Repository: Implementación y buenas prácticas. Disponible en https://our-academy.org/posts/el-patron-repository:-implementacion-y-buenas-practicas , consultado el 03/2024.

Anexo A

Guía de uso de *Social Lift*.

La aplicación dispone de 3 ventanas principales:

Feed de Rutinas

Desde esta ventana de la aplicación, en la parte de ***Publicaciones***, el usuario podrá visualizar rutinas y guardar plantillas rutinas para su uso posterior. Dicha función se muestra en la figura 56.

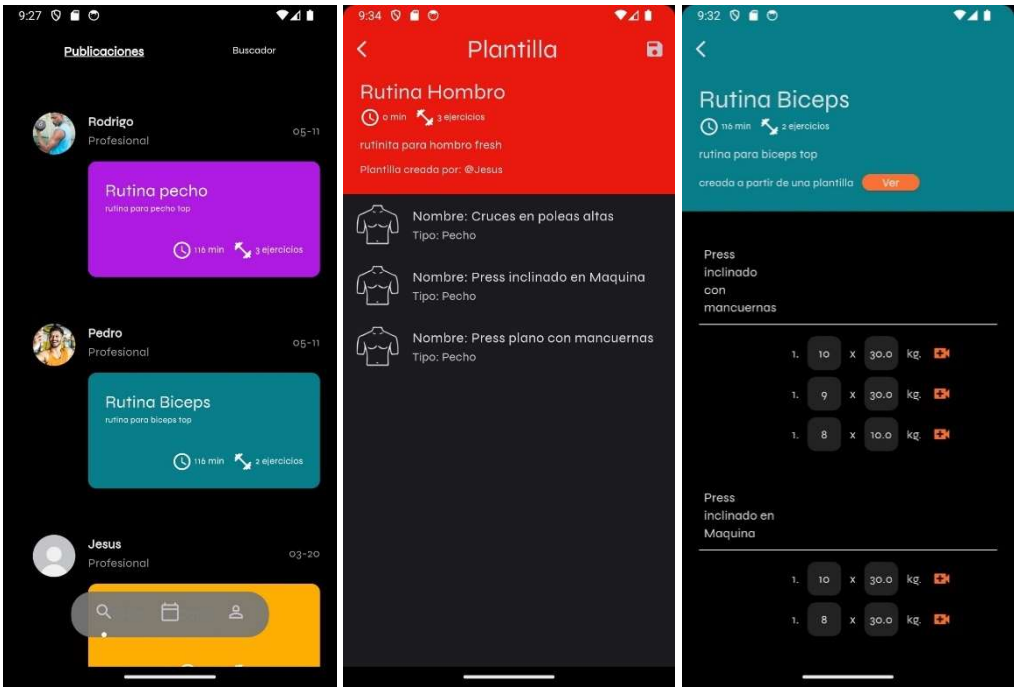

Figura 56. interfaz feed

Solo podrás guardar plantillas rutinas creadas por otros usuarios

Dentro del Feed en el apartado de ***Buscador,*** mostrado en la figura 57, podrás buscar tanto **usuario** como **plantillas de rutinas** que quieras realizar en tus entrenamientos.

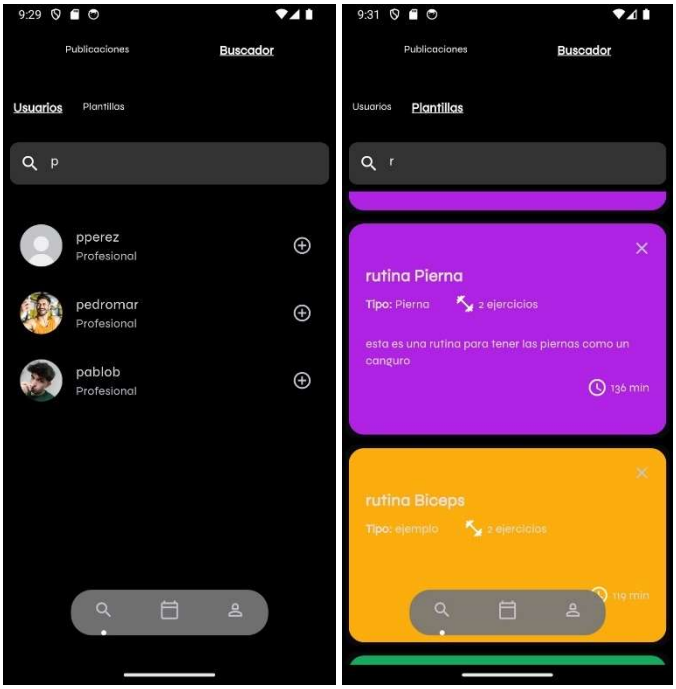

Figura 57. buscador

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.

Calendario de Rutinas

En la figura 58 se muestra la sección del calendario mensual en el cual se pueden añadir rutinas a partir de plantillas guardadas o crear rutinas personalizadas. Estas se muestran mediante un punto de color en la parte inferior del y al pulsar el día, se podrán visualizar.

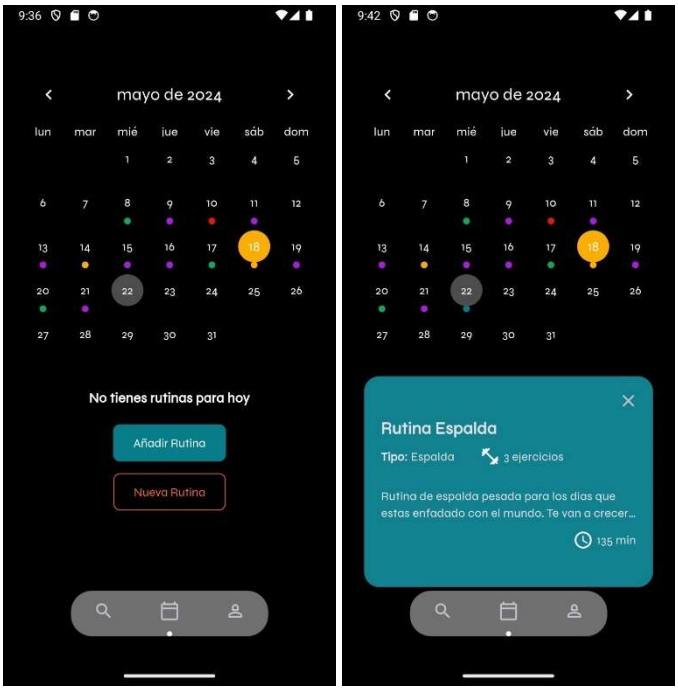

Figura 58. Calendario interactivo

Al seleccionar el botón ***Añadir***, te redirige a una ventana donde puedes seleccionar tus plantillas guardadas y tus plantillas creadas. Esta interfaz está mostrada en la figura 59.

Figura 59. Plantillas

.

Al seleccionar el botón ***Crear*** podrás crear tu rutina personalizada, como está mostrada en la figura 60.

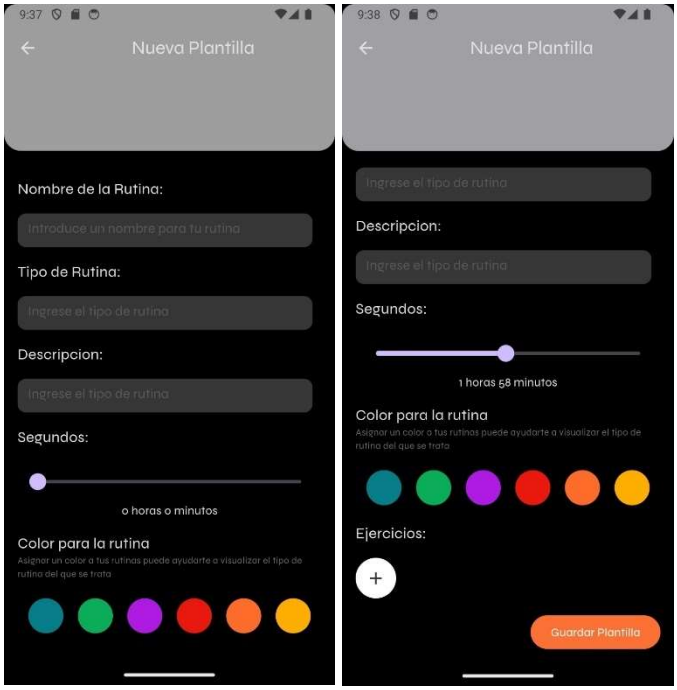

Figura 60. Crear rutina

En el apartado de ***Ejercicios***, le podrás añadir los ejercicios que tú quieras, seleccionando entre tus ejercicios creados personalizados o todos los ejercicios de la aplicación. Figura 61.

Figura 61. Selector ejercicios

Al crear la rutina, tras añadir los ejercicios, quedará de la siguiente manera en la figura 62.

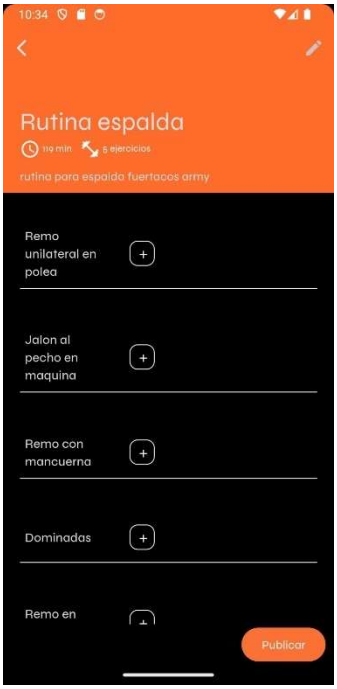

Figura 62. Series

Pulsando el botón de **+** podrás añadir tus series, definiendo el peso, las repeticiones y teniendo la posibilidad de subir un video por cada serie creada.

Perfil de Usuario

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

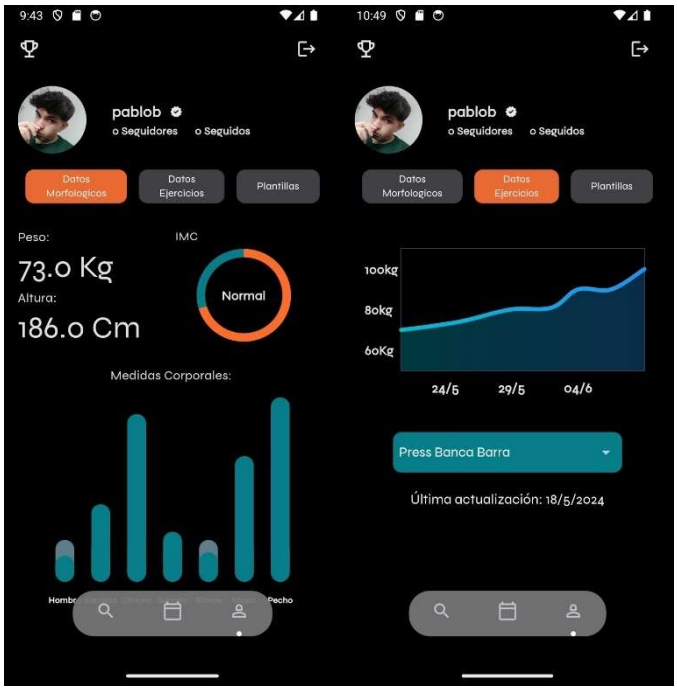

Figura 63. Perfil usuario

Dentro de datos morfológicos, en la figura 63 te describen tus estadísticas personales como peso, altura, IMC y medidas corporales.

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

Dentro del apartado de ***Plantillas,*** mostrado en la figura 64, el usuario podrá visualizar y editar tanto sus rutinas guardadas, las cuales ha guardado de otro usuario o sus propias rutinas creadas.

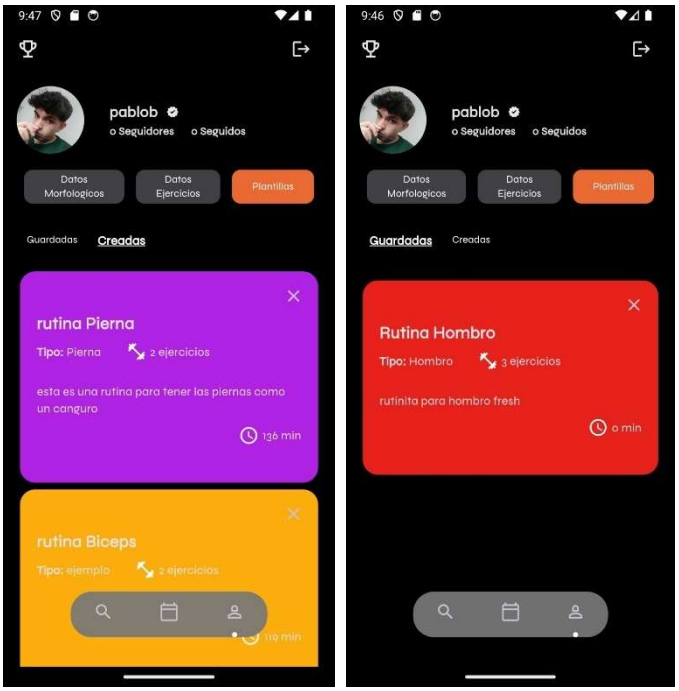

Figura 64. Rutinas guardadas

Esta sección de la aplicación muestra un ***ranking*** estadístico del número de entrenamientos realizados por los usuarios de la aplicación. Figura 65.

Figura 65. 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. **Fin de la pobreza.**				
ODS 2. **Hambre cero.**				
ODS 3. **Salud y bienestar.**				
ODS 4. **Educación de calidad.**				
ODS 5. **Igualdad de género.**				
ODS 6. **Agua limpia y saneamiento.**				
ODS 7. **Energía asequible y no contaminante.**				
ODS 8. **Trabajo decente y crecimiento económico.**				
ODS 9. **Industria, innovación e infraestructuras.**				
ODS 10. **Reducción de las desigualdades**.				
ODS 11. **Ciudades y comunidades sostenibles**.				
ODS 12. **Producción y consumo responsables.**				
ODS 13. **Acción por el clima.**				
ODS 14. **Vida submarina.**				
ODS 15. **Vida de ecosistemas terrestres.**				
ODS 16. **Paz, justicia e instituciones sólidas.**				
ODS 17. **Alianzas para lograr objetivos.**				

Tabla 7. ODS

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

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*. El TFG de desarrollo de la aplicación *Social Lift* brinda una oportunidad para analizar sobre el cómo contribuye a los Objetivos de Desarrollo Sostenible (ODS) propuestos por la Organización de las Naciones Unidas (ONU). En este análisis se estudia en detalle la conexión entre Social Lift y los diversos ODS, destacando áreas de impacto, contribuciones indirectas y metas con las que la relación no es tan clara.

ODS 3: Salud y Bienestar

La principal relación de Social Lift es con el ODS 3, el cual busca asegurar una vida saludable y fomentar el bienestar para todos en todas las edades. *Social Lift* promueve la salud y el bienestar de varias maneras:

- **Motivación a la actividad física**: Brinda herramientas para que los usuarios pueden hacer seguimiento de sus entrenamientos y progreso, al igual que, proporciona una gran variedad de ejercicios y rutinas. Esto promueve hábitos saludables y una vida activa, elementos cruciales para el bienestar físico.
- **Educación y conocimiento**: mediante un muro social integrado de rutinas, los usuarios expertos pueden intercambiar conocimientos y experiencias de rutinas y ejercicios a otros usuarios, lo que incrementa el conocimiento sobre prácticas de ejercicio eficaces y seguras.
- **Feedback y mejora continua**: La aplicación ofrece herramientas para que los usuarios verificados puedan dar retroalimentación a otros usuarios, de esta manera, ayuda a mejorar la técnica y prevenir lesiones, promoviendo una práctica de ejercicio más segura y efectiva.

Social Lift ofrece herramientas que facilitan la salud física y mental para el bienestar general de manera personalizada y accesible.

ODS 4: Educación de calidad

Social Lift contribuye al ODS 4 proporcionando un espacio de educación informal en el intercambio de conocimientos sobre el sector del *fitness.* De esta forma crea un espacio de educación no formal en donde se comparten 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 un sistema inclusivo donde todos los usuarios, independientemente de su género, tienen acceso equitativo a herramientas de *fitness* y bienestar.

Fomentar un entorno inclusivo, equitativo y justo 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 tecnología innovadora de *Social Lift* en salud y *fitness* cumple con el ODS 9, el cual impulsa el desarrollo de infraestructuras sólidas, 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

Por último, Social Lift puede promover asociaciones y colaboraciones en la comunidad fitness, creando una colaboración colectiva para mejorar la salud y bienestar. Estas alianzas son esenciales para maximizar el impacto de la aplicación y contribuir al logro de objetivos de desarrollo sostenible.

En resumen, Social Lift está estrechamente relacionado con varios Objetivos de Desarrollo Sostenible, destacando el ODS 3, 4, 5, 9, 10, 17. La aplicación no solo fomenta la salud y el bienestar, sino que también promueve la educación, igualdad, innovación y la reducción de desigualdades mediante su plataforma inclusiva y colaborativa. Social Lift puede ayudar al desarrollo sostenible en diversas áreas, ofreciendo un gran potencial tecnológico y mostrando la importancia en integrar aspectos de sostenibilidad en el diseño y desarrollo de aplicaciones tecnológicas para maximizar su impacto positivo en la sociedad.