



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una red social fitness: Back-end.

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Picó Rico, Jesús

Tutor/a: Molina Marco, Antonio

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Desarrollo de una red social fitness móvil: Back-End

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Jesús Picó Rico

**Tutor:** Antonio Molina Marco

2022/2023



# Resumen

---

Fit n Fed es una aplicación multiplataforma que se basa en crear y compartir difusiones, rutinas y recetas. Los usuarios podrán seguirse mutuamente, ver las recetas y rutinas que los demás usuarios hayan publicado y buscar las que se adapten mejor a ellos. Además, los usuarios podrán planificar su semana mediante un calendario semanal, entre otras cosas. El objetivo de la aplicación es unir alimentación y deporte en una sola aplicación además de fomentar los hábitos saludables de las personas.

La parte de crear y compartir recetas y rutinas está terminada, por lo que ahora se continuará implementando la planificación semanal, se añadirán ajustes adicionales y se mejorará la información de las recetas y rutinas para dar una mejor experiencia al usuario añadiendo valoraciones e información adicional entre otras cosas.

La aplicación estará basada en .NET MAUI. Para el Back-End necesitaremos una API RESTful HTTP que estará basada en ASP.NET Core. Desde esta se gestionará el acceso a datos mediante Entity Framework y la lógica del negocio de la aplicación. Finalmente, utilizamos una base de datos SQL Server para la persistencia de datos. Tanto la Web API como la base de datos están hospedadas en Azure. Por último, como IDE utilizaremos Visual Studio.

**Palabras clave:** C#, API REST, Back-End, Fitness, Red Social



## Abstract

---

Fit n Fed is a multiplatform application based on creating and sharing diffusions, workout routines and recipes. Users will be able to follow each other, view the recipes and routines that other users have posted, and search for ones that best suit them. Additionally, users will be able to plan their week using a weekly calendar, among other things. The goal of the application is to combine nutrition and exercise in a single app, as well as promote healthy habits among people.

The part of creating and sharing recipes and routines is almost finished, so now we will continue implementing the weekly planning, private messaging between users, adding additional settings, notifications, and improving the information of the recipes and routines to provide a better user experience by including ratings and additional information, among other things.

The application will be based on .NET MAUI. For this, we need an API that should be a RESTful HTTP service based on ASP.NET Core. It will handle data access using Entity Framework and the business logic of the app. Finally, we will use a SQL Server database for data persistence. Both the Web API and the database are hosted on Azure. Lastly, we will use Visual Studio as the IDE.

**Keywords:** C#, API REST, Back-End, Fitness, Social Network

# Tabla de contenidos

---

1. Introducción .....	7
1.1. Motivación .....	7
1.2. Objetivos .....	8
1.3. Estructura de la memoria .....	9
2. Generación de la idea de negocio .....	10
2.1. Idea inicial .....	10
2.2. Evolución de la idea de negocio .....	11
3. Evaluación de la idea de negocio .....	14
3.1. Estudio de mercado .....	14
3.1.1. Aplicaciones relacionadas con la alimentación, el deporte y redes sociales.	14
3.1.2. Tabla comparativa .....	18
3.2. Análisis DAFO .....	20
3.3. Modelo de negocio y proyección económica .....	22
3.4. Lean Canvas .....	25
3.5. Conclusiones de la evaluación .....	26
4.1. Mapa de características .....	27
4.2. Cronología del proyecto .....	31
4.3. Prueba con <i>early adopters</i> .....	33
5.1. Visual Studio 2022 .....	36
5.2. ASP.NET Core .....	38
5.3. Entity Framework Core .....	39
5.4. SQL Server .....	40
5.6. Swagger .....	42
6.1. Estado inicial .....	43
6.2. Primer Sprint .....	45
6.2.1. Valoración de publicaciones .....	45
6.2.2. Planificación semanal .....	47
6.3. Segundo Sprint .....	48
6.3.1. Macronutrientes .....	48
6.3.2. Ajustes .....	49



6.4. Tercer Sprint .....	52
6.4.1. Nuevos ajustes .....	52
6.4.2. Mejora de la planificación .....	53
6.5. Estado final .....	55
7. Conclusiones .....	56
8. Trabajo futuro .....	57
9. Bibliografía.....	58
Anexo .....	59
Objetivos de Desarrollo Sostenible.....	59

# 1. Introducción

---

## 1.1. Motivación

Hace poco, la humanidad acaba de pasar una pandemia global, la pandemia del COVID-19, la cual ha afectado negativamente a la alimentación y el ejercicio de las personas, entre otras cosas. Esto produce que la mala alimentación y el sedentarismo sean unos de los grandes problemas a nivel físico que existen hoy en día.

Y es que según el INE (Instituto Nacional de Estadística), cerca del 53% de las mujeres y del 42% de los hombres a partir de 15 años comían verduras antes de la pandemia, dato que cambia a partir de la llegada del COVID-19, donde ahora las estadísticas caen hasta el 50% en mujeres y el 38% en hombres.

Como podemos observar, existe una caída aproximada del 3% y 4% respectivamente en consumo de vegetales diarios. Además, el consumo de dulces, comida rápida y alimentos ultra procesados ha aumentado durante estos años, por lo que la alimentación general de las personas se ha visto perjudicada.

Por otro lado, durante el COVID-19 ha aumentado el sedentarismo de manera drástica. Según un estudio hecho por la OMS (Organización Mundial de la Salud), cerca del 28% de los adultos y del 81 %de adolescentes no llegaban al nivel recomendado de actividad física durante 2022 por culpa de la pandemia (OMS, 2022).

Por todo esto, podemos concluir que hay grave problema en la sociedad actual en cuanto a la salud física de las personas y si no intentamos ponerle solución la situación solo podrá empeorar.

Por otra parte, si tenemos si tuviéramos que decir cuál es nuestra herramienta más importante no dudaríamos en elegir nuestro teléfono móvil. Se ha convertido en algo imprescindible y en algo que usamos para realizar cualquier tarea, ya sea hablar con familiares o amigos, jugar a videojuegos o incluso buscar nuevas maneras de hacer ejercicio y nuevas formas de mejorar en nuestra alimentación.

Por ello, la propuesta de nuestro proyecto se basa en promover la actividad física y la buena alimentación mediante una aplicación móvil y red social que ayude a los usuarios a conseguir estos objetivos creando y compartiendo sus rutinas de ejercicio y recetas de cocina con otros usuarios.

Si nos ponemos a pensar en la gran cantidad de aplicaciones móviles que hay, vemos que existe por un lado un mundo de la alimentación donde hay aplicaciones que nos ayudan a descubrir nuevas recetas y trucos de cocina y, por otro lado, existe una gran cantidad de aplicaciones que nos proporcionan rutinas de ejercicio de todo tipo.

Sin embargo, no existe ninguna aplicación que combine alimentación y deporte, sabiendo que ambos mundos son totalmente necesarios para nuestra salud física. Por este motivo decidimos empezar este proyecto, con la intención de que las personas puedan tener a su alcance una aplicación donde puedan crear y compartir tanto rutinas

como recetas, así como planificar su semana para conseguir sus objetivos de la forma más eficiente posible.

Pero sobre todo queremos enfocarnos en un punto muy importante, en compartir esa experiencia. Queremos que nuestros usuarios tengan la opción de aprender unos de otros y evolucionar conjuntamente, ya sea en el ámbito de la alimentación, en el del deporte o en ambos.

Por todo lo mencionado anteriormente, la motivación principal del proyecto es incitar y ayudar a que la sociedad no siga empeorando su salud física después de los efectos de la pandemia y con nuestra aplicación queremos conseguir darles ese empujón inicial para que puedan alcanzar sus metas.

## 1.2. Objetivos

Durante nuestras conversaciones con los miembros del proyecto (en la última etapa Javier Ruano y yo, Jesús Picó) llegamos a la conclusión de que debíamos fusionar los mundos de la alimentación y el *fitness*.

El objetivo principal de nuestro proyecto consiste en desarrollar una aplicación que permita a los usuarios crear y compartir recetas de cocina y rutinas de ejercicio, además de facilitar la comunicación entre ellos mediante publicaciones o posts. Este Trabajo de Fin de Grado (TFG), en particular, se centra en documentar el *Back-End* de la aplicación, como se mencionó en el resumen. Esto se debe a que el proyecto fue llevado a cabo por un equipo de dos personas, y en el TFG de mi compañero se documenta el *Front-End* de la aplicación.

Con el fin de mantener la claridad, dividiremos los objetivos en dos categorías: los objetivos relacionados con el proyecto de desarrollo (la aplicación en sí) y los objetivos específicos de este TFG.

Objetivos de aplicación:

- Ayudar a los usuarios a mejorar su bienestar.
- Hacer que los usuarios puedan a conocer gente en su misma situación.
- Unir ambos mundos, nutrición y ejercicio.
- Permitir la planificación de recetas y rutinas de forma sencilla.

Objetivos específicos del TFG (*Back-End*):

- Mejorar la mantenibilidad de la API.
- Separar el modelo de datos de *Back-End* y *Front-End*
- Continuar ampliando la lógica de negocio del proyecto.

### 1.3. Estructura de la memoria

La memoria está estructurada en varios capítulos:

El primer capítulo abarca la introducción, donde hemos abordado las razones que nos inspiraron a emprender este proyecto, junto con los objetivos a lograr tanto en términos de desarrollo como en el marco de nuestro Trabajo de Fin de Grado (TFG).

En el segundo capítulo, exploraremos la génesis de nuestra idea de negocio, su origen y los motivos subyacentes que la respaldan. También examinaremos su evolución desde sus inicios hasta la concepción actual del negocio.

En el tercer capítulo, nos adentraremos en la evaluación y análisis de la idea de negocio. En esta sección, analizaremos aspectos como las diferencias y similitudes con otras aplicaciones relacionadas con la alimentación y el deporte, el análisis DAFO, el modelo de negocio y las proyecciones económicas del proyecto. Además, presentaremos el Lean Canvas y concluiremos el capítulo con un resumen breve de la evaluación realizada.

El cuarto capítulo se centrará en el desarrollo de la idea de negocio, donde detallaremos el mapa de características de la aplicación y proporcionaremos una explicación breve de cada una de estas funcionalidades. También abordaremos la cronología del proyecto, siguiendo las distintas etapas de desarrollo y la producción del MVP (Producto Mínimo Viable).

En el quinto capítulo, hablaremos sobre las tecnologías empleadas en la ejecución del proyecto. Aquí, discutiremos las herramientas necesarias para el desarrollo del Back-End de la aplicación, y describiremos sus funciones y roles en el proceso de desarrollo.

El sexto capítulo se centrará en la implementación de nuevas características en la aplicación, dividiéndolo en tres secciones correspondientes a cada sprint de desarrollo. En cada sprint, explicaremos como hemos abordado los diferentes retos a la hora de implementar las nuevas funcionalidades. Además, en este capítulo expondremos el estado inicial y el estado final.

Por último, en el séptimo y octavo capítulo, presentaremos las conclusiones del proyecto y detallaremos las tareas pendientes para el futuro desarrollo de la aplicación.

## 2. Generación de la idea de negocio

---

En este capítulo, exploraremos la génesis de la idea de negocio inicial para el proyecto y rastreamos su transformación hasta la configuración actual de la propuesta empresarial.

### 2.1. Idea inicial

Basándonos en los objetivos previamente establecidos, se implementaron sesiones de *brainstorming* con el objetivo de definir los requisitos iniciales que servirían como fundamento para el desarrollo de la aplicación. Estas sesiones se llevaron a cabo de manera periódica hasta dar forma completa a la aplicación.

Después de un exhaustivo debate y la presentación de diversas ideas, se llegó a la conclusión de que existían cuatro pilares fundamentales sobre los cuales se sustentaría la aplicación:

- **Feed de Contenido:** Dado que la aplicación tenía un enfoque de red social, se consideró esencial incluir un *feed*, entendido como la sección principal donde los usuarios pueden visualizar y participar en las publicaciones de otros usuarios.
- **Gestión de Usuarios:** Se propuso que los usuarios pudieran crear cuentas y realizar un proceso de inicio de sesión. Además, se contempló la posibilidad de que los usuarios gestionaran sus propios perfiles, incluyendo la edición de información personal.
- **Creación de Contenido:** Uno de los objetivos principales era implementar un mecanismo sencillo para que los usuarios pudieran crear fácilmente rutinas de ejercicio y recetas de cocina. Se diseñó un botón de acceso rápido que permitiera crear publicaciones en tres categorías principales: recetas, rutinas y difusiones (una forma de publicación que consiste en texto, utilizada para informar o intercambiar mensajes con seguidores).
- **Interfaz Amigable:** Se prestó especial atención al diseño de la interfaz de usuario, con el objetivo de garantizar que fuera fácil de usar y comprender. Se buscaba evitar cualquier confusión en la experiencia del usuario, y se decidió que toda la interfaz de la aplicación debía mantener un estilo coherente y amigable.

Además de estos pilares iniciales, se contemplaba en un principio la posibilidad de conectar a usuarios novatos con profesionales, como una extensión del proyecto. Esta idea se basaba en establecer acuerdos con universidades y crear un sistema donde los recién graduados pudieran ofrecer sus servicios a través de la aplicación, y la plataforma actuaría como intermediaria, recibiendo un porcentaje de los pagos realizados a los profesionales.

En resumen, la visión general de la aplicación era ofrecer una plataforma totalmente gratuita donde los usuarios pudieran conectar, crear contenido y acceder a servicios profesionales para ayudarles a alcanzar sus objetivos.

## 2.2. Evolución de la idea de negocio

La mejor forma para entender la evaluación de la idea de negocio inicial de la aplicación es observando algunos *mockups* de la aplicación, donde se puede apreciar mejor la evolución y mejora de las funcionalidades. Cabe decir que estos *mockups* son el resultado de una colaboración con estudiantes de bellas artes.

Inicialmente, debatimos sobre cuál sería el color principal del tema de la aplicación, ya que deseábamos que fuera fácilmente reconocible. De estas conversaciones, llegamos a la conclusión de que el color principal sería principalmente naranja, inspirado en los tonos de las naranjas de Valencia. Luego, debimos seleccionar un color secundario para la paleta, y tras diversas reuniones, optamos por un tono oscuro, inclinándonos hacia el negro. Esta elección nos permitía la posibilidad de desarrollar más adelante dos temas diferentes, uno con un fondo oscuro y otro con un fondo claro. Para este propósito, colaboramos con compañeras de la facultad de bellas artes para que le dieran estilo y aplicaran la paleta de colores, lo cual posteriormente nos entregaron a través de la plataforma Figma.

A continuación, observaremos cómo quedaron algunos *mockups* de la aplicación en comparación con el estado de las interfaces en ese momento, así como algunas de las adiciones que realizamos a los requisitos debido a nuevas ideas para las interfaces:

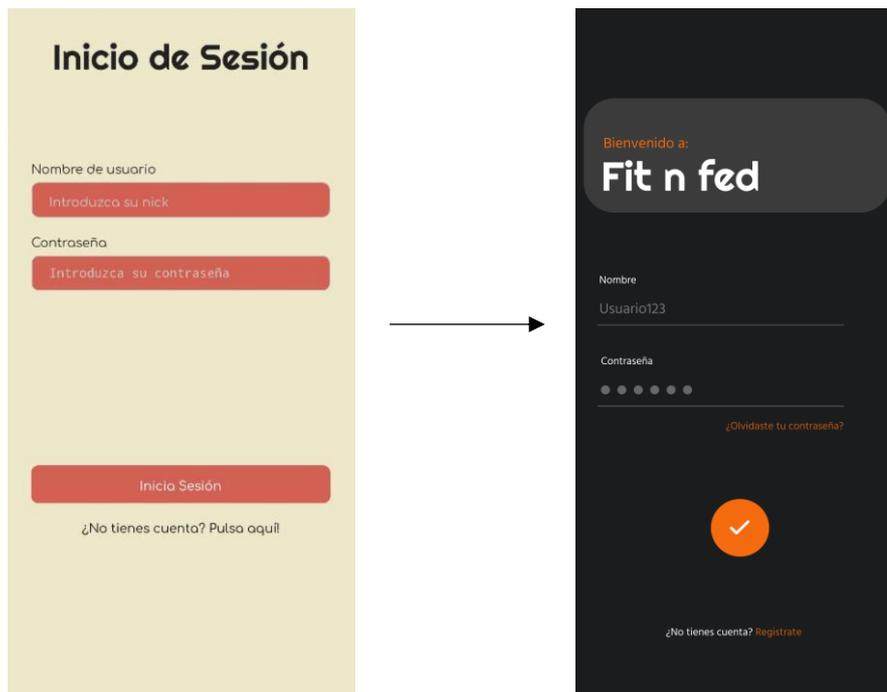


Figura 1: Antes y después de cerrar el estilo de la aplicación para el inicio de sesión

En la Figura 1, podemos observar que la interfaz presenta principalmente un cambio visual, mientras que las funcionalidades previamente diseñadas se mantienen intactas.

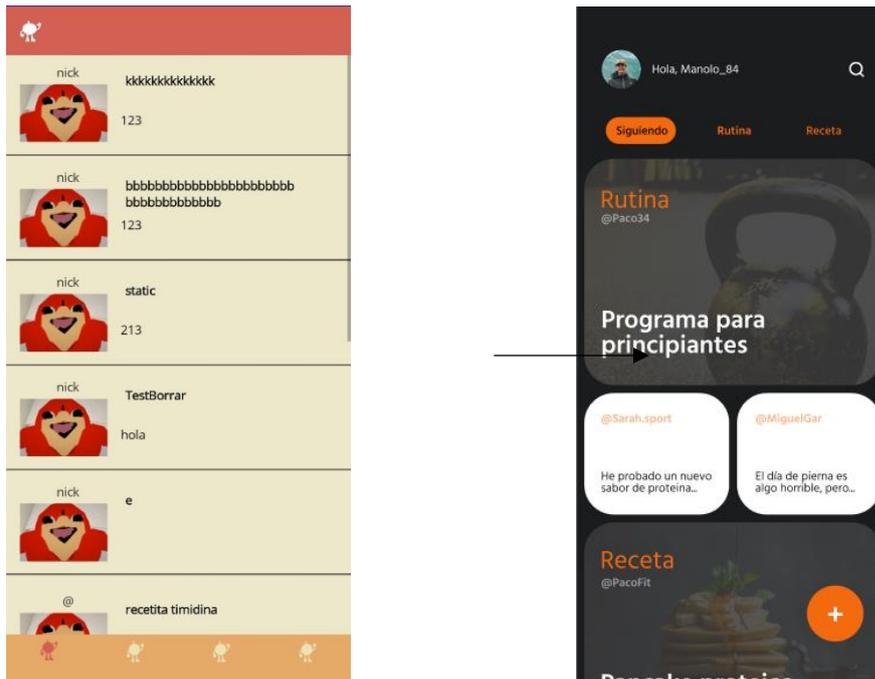


Figura 2: Antes y después de cerrar el estilo de la aplicación para el feed principal

En la Figura 2, observamos un cambio drástico en la interfaz, donde el *feed* ha experimentado una transformación significativa, adaptándose al nuevo estilo de la aplicación y siguiendo la nueva paleta de colores.

Sin embargo, es importante destacar que esta no es la versión final de la interfaz debido a ciertas limitaciones tecnológicas que no pudimos abordar en el tiempo disponible. Estas limitaciones afectan especialmente a las "difusiones," ya que en la nueva interfaz se presentan dos difusiones contiguas en columna, pero en la versión final se muestra solo una.

En términos de funcionalidades, se puede apreciar la incorporación de un botón en la esquina inferior derecha, que permite acceder a la creación de publicaciones. Este botón incluye una pequeña animación que despliega tres opciones desde el lado derecho de la pantalla: receta, rutina y difusión.

Además de los cambios mencionados, la pequeña barra superior se reemplazó por un menú desplegable que se abre desde la izquierda de la pantalla o haciendo clic en la pequeña foto de perfil en la parte superior del *banner*. El menú inferior fue sustituido por pequeños botones ubicados debajo de la foto de perfil, lo que permite cambiar el *feed*. Con esta modificación, se añadieron otros dos tipos de *feed* a la aplicación: uno exclusivamente para recetas y otro exclusivamente para rutinas.

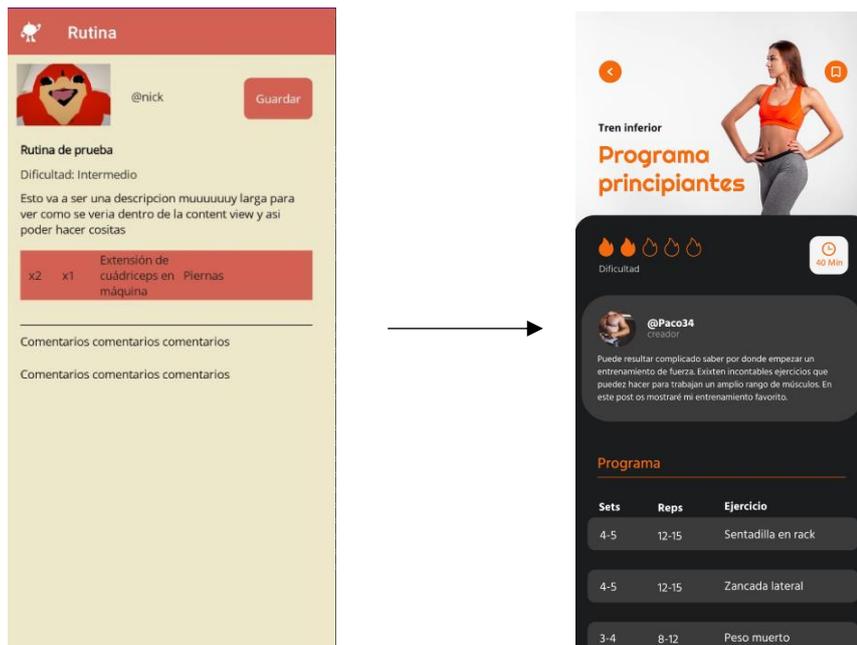


Figura 3: Antes y después de cerrar el estilo de la aplicación para ver las rutinas

En la Figura 3, se aprecian una serie de cambios significativos en la interfaz de la aplicación:

- **Foto de Perfil Diferenciada:** La foto del *post* ahora es una imagen que distingue entre rutina y receta, lo que facilita la identificación de la categoría de la publicación.
- **Descripción y Dificultad:** La descripción se encuentra encapsulada debajo de la foto y la información de dificultad. Esto ayuda a mantener una presentación ordenada y clara.
- **Lista de Series y Repeticiones:** En el caso de las rutinas, se incluye una lista de series y repeticiones, proporcionando detalles esenciales para los usuarios interesados en ejercicios.
- **Botón de Guardar:** El botón de guardar ahora se presenta como un pequeño icono en la esquina superior derecha, lo que permite a los usuarios guardar publicaciones. Esta funcionalidad se complementa con una nueva característica que permite a los usuarios consultar sus publicaciones guardadas desde una interfaz exclusiva.
- **Comentarios en Todas las Publicaciones:** Se ha añadido la capacidad de realizar comentarios en todas las publicaciones, lo que fomenta una mayor interacción entre los usuarios.

En cuanto al modelo de negocio, se decidió cambiar la opción de contacto con profesionales debido a limitaciones prácticas. La aplicación pasó a ser completamente gratuita, con la posibilidad de una suscripción mensual que desbloquea funcionalidades

adicionales y elimina anuncios que se introducirían más adelante. Esta nueva estrategia convierte a la aplicación en un modelo *freemium*, lo que significa que es gratuita para su uso básico, pero ofrece mejoras con la versión premium. Las mejoras en la versión premium se centran en mejorar la comodidad y eliminar anuncios, sin agregar funcionalidades exclusivas para los usuarios de pago.

Un ejemplo de esta estrategia es el acceso a los macronutrientes de una receta. Los usuarios que no son premium pueden ver un anuncio de vídeo para acceder a esta información, mientras que los usuarios que pagan pueden acceder directamente. Este es solo uno de los ejemplos, y se deja abierta la posibilidad de realizar cambios adicionales a medida que se continúa trabajando en la aplicación.

## 3. Evaluación de la idea de negocio

---

Para saber si la idea de negocio es viable debemos realizar una evaluación de esta. Para ello realizaremos un estudio de mercado en el cual analizaremos las aplicaciones de la competencia y las compararemos con la nuestra, realizaremos el análisis DAFO del proyecto, veremos el modelo de negocio que se implementará, el *Lean Canvas* y, para finalizar, expondremos las conclusiones determinando la viabilidad del proyecto.

### 3.1. Estudio de mercado

Es este apartado se analizarán las aplicaciones de la competencia y realizaremos una tabla comparativa entre estas aplicaciones y *Fit n Fed*.

#### 3.1.1. Aplicaciones relacionadas con la alimentación, el deporte y redes sociales.

Nuestra aplicación, *Fit n Fed*, está enfocada en atraer la atención de aquellos usuarios a los que les interesa el mundo de la alimentación y el deporte sea cual sea su grado de implicación en el ámbito *Fitness*.

Hoy en día existen millones de aplicaciones móviles de todo tipo que cubren cualquier necesidad que una persona se pueda imaginar. En concreto, dentro del ámbito *Fitness* existe una gran cantidad de aplicaciones tanto en el ámbito de la alimentación como en el ámbito del deporte.

Por ello, lo primero que debemos hacer para saber cómo debemos diseñar nuestra aplicación es investigar que aplicaciones de ámbito *Fitness* existen y en concreto cuales

son las más parecidas a nuestra idea final de *Fit n Fed* para compararlas y descubrir que cosas son esenciales en una aplicación de este tipo y cómo podemos conseguir que nuestra aplicación se diferencie del resto y tenga un valor añadido.

Además, debemos de fijarnos también en las aplicaciones móviles de redes sociales dado que nuestra aplicación también será una red social.

A continuación, explicaremos cuales son las aplicaciones que podrían hacernos competencia.

## *Cookpad*

Primero hablaremos de *Cookpad*, una red social de cocina basada en crear, compartir y encontrar recetas y trucos de cocina.

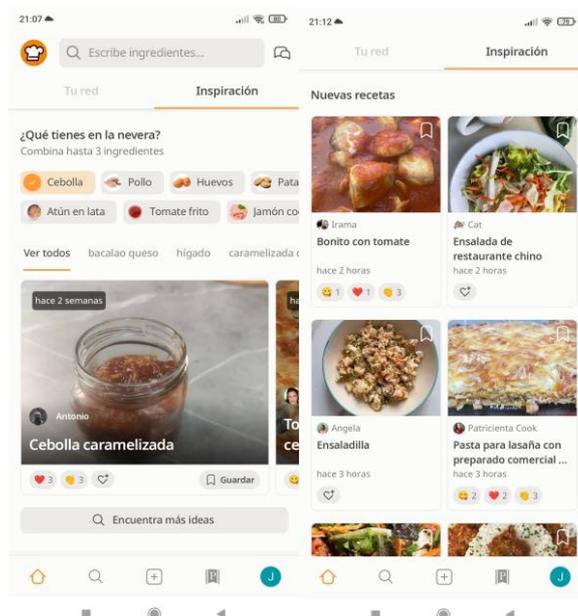


Figura 4: Capturas de pantalla de Cookpad

Esta aplicación ofrece un buscador de recetas bastante completo, ya que se puede buscar por ingrediente, tipo de comida, tipo de cocina y palabras clave. En nuestra aplicación también implementaremos un buscador de recetas similar.

Otra similitud es que *Cookpad* es una aplicación *freemium*, por lo que tiene una suscripción mensual con la que desbloquearemos diferentes funcionalidades adicionales además de dejar de ver publicidad al usar la aplicación.

La primera diferencia es que además de recetas podemos encontrar trucos de cocina bastante interesantes y *cooksnap*s, publicaciones de fotos con una descripción y comentarios de los usuarios. En nuestra aplicación no existirán estos objetos como tal,

pero habrá difusiones que podrán hacer la función tanto de los trucos como de los *cooksnap*s.

Otra diferencia es que a pesar de ser una red social en la cual se puede ver los perfiles de otros usuarios y seguirlo o bloquearlos, no existe la posibilidad de tener un chat privado con ellos, cosa que sí que queremos implementar en nuestra aplicación.

Además, en esta aplicación puedes guardarte las recetas, pero no existe ningún tipo de planificación, es decir, no puedes asociar las recetas a una planificación o día en concreto. En *Fit n Fed* sí que se podrá hacer, para así conseguir facilitar la implementación de una planificación *Fitness* semanal en la vida de los usuarios.

Finalmente, la principal diferencia es que *Cookpad* está únicamente orientada a la alimentación por lo que si también te interesa el ámbito del deporte deberás tener otra aplicación aparte. Con *Fit n Fed* esto no pasa dado que busca unir alimentación y deporte.

### *Calisteniapp*

Ahora hablaremos de *Calisteniapp*, una aplicación basada en rutinas de calistenia, ejercicios con nuestro peso corporal.

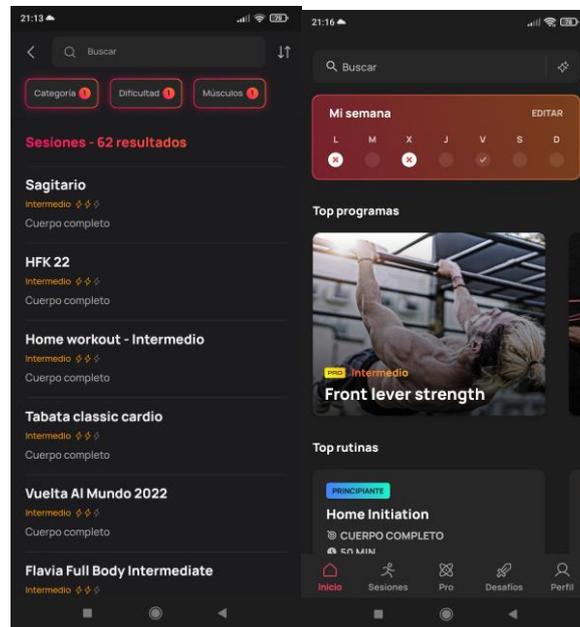


Figura 5: Capturas de pantalla de *Calisteniapp*

Igual que la aplicación anterior, *Calisteniapp* ofrece un buscador muy completo, esta vez para buscar las rutinas de calistenia que existen en la aplicación en base a diferentes

parámetros. En nuestra aplicación también contaremos con un buscador de rutinas muy similar.

Otra similitud tanto con *Fit n Fed* como con *Cookpad* es que la aplicación es de tipo *freemium*. En este caso desbloqueamos las opciones de Programas y *Smart Progress*. Esto son diferentes planificaciones de rutinas con un objetivo específico que van cambiando según pasa el tiempo para conseguir dicho objetivo.

En esta aplicación podemos observar que sí que existe una planificación semanal, al igual que existirá en *Fit n Fed*, que nos permite planificar nuestras rutinas distribuyéndolas en varios días. Una diferencia con nuestra aplicación es que nosotros dispondremos de más de una planificación semanal por usuario y en *Calisteniapp* solo podemos tener una.

Otra diferencia es que las rutinas pueden ser ejecutadas, es decir, a la hora de entrenar puedes hacer un seguimiento de la rutina, sabiendo así el ejercicio en el que estás y el tiempo que llevas. Esto no estará de momento en *Fit n Fed* dado que impide que uses las demás funcionalidades de la aplicación mientras se hace el seguimiento de la rutina y no creemos que sea algo necesario.

Esta vez la aplicación no es una red social, no existe ningún tipo de interacción entre usuarios, a pesar de sí tener un perfil de usuario donde se ven estadísticas relacionadas con la cantidad de rutinas realizadas y tiempo invertido.

Finalmente, al igual que *Cookpad* solo se centra en la alimentación *Calisteniapp* solo se centra en el deporte, en concreto en la calistenia, por lo que necesitarías usar ambas aplicaciones para unir los dos ámbitos principales del *Fitness*. Con nuestra aplicación conseguiremos unir alimentación y deporte en la misma aplicación.



### 3.1.2. Tabla comparativa

Un estudio de mercado no está completo sin realizar una tabla comparativa de las aplicaciones mencionadas anteriormente con *Fit n Fed*.

Para realizar esta tabla se han probado todas las aplicaciones para descubrir si cubrían las características más interesantes a incluir en nuestra aplicación.

Característica	Cookpad	Calisteniapp	Fit n Fed
Perfil de usuario	Sí	Sí	Sí
Seguir usuarios	Sí	No	Sí
Crear recetas	Sí	No	Sí
Crear rutinas	No	Sí	Sí
Compartir recetas	Sí	No	Sí
Compartir rutinas	No	No	Sí
Crear y compartir difusiones	Sí (Cooksnaps)	No	Sí
Añadir multimedia	Sí	No	Sí
Comentarios	Sí	No	Sí
Buscador de recetas	Sí	No	Sí
Buscador de rutinas	No	Sí	Sí
Buscador de usuarios	Sí	No	Sí
Planificación	No	Sí	Sí
Guardar recetas	Sí	No	Sí
Guardar rutinas	No	No	Sí
Espacio privado	Sí	Sí	Sí
Valorar publicaciones	Sí	Sí	Sí
Notificaciones	Sí	Sí	Sí
Chat privado	No	No	Sí
Seguimiento de las rutinas	No	Sí	No
Desafíos de rutinas	No	Sí	No
Subscripción mensual	Sí	Sí	Sí
Sugerencias personalizadas	Sí	Sí	No
Información nutricional	No	No	Sí
Ajustes	Sí	Sí	Sí
Multiplataforma	Sí	Sí	Sí
Publicidad	Sí	No	Sí

Figura 6: Tabla comparativa de características entre aplicaciones similares.

En primer lugar, debemos comentar que solo hemos comparado nuestra aplicación con *Calistiapp* y *Cookpad* dado que las demás aplicaciones en el ámbito de alimentación y deporte eran muy parecidas a estas y ofrecían muy pocas funcionalidades o no tenían nada que ver con lo que ofrecemos en nuestro proyecto, en cambio estas dos aplicaciones ofrecen una gran diversidad de funcionalidades muy completas e interesantes.

Lo primero que observamos es que al buscar que *Fit n Fed* combine los ámbitos de la alimentación y el deporte y que sea lo más completa posible, nuestra aplicación ofrecerá prácticamente todas las funcionalidades que ofrecen estas dos aplicaciones, pero existen tres funcionalidades que no pensamos implementar, al menos de momento. Los desafíos de rutinas y las sugerencias personalizadas no las meteremos dado que nuestra intención es que los usuarios interactúen entre sí, buscando y seleccionando las recetas y rutinas sin que nosotros intervengamos en sus decisiones. Por otro lado, el seguimiento de las rutinas no lo implementaremos dado que si en un futuro ofrecemos la funcionalidad de poder contactar con profesionales en nuestra aplicación serán los propios profesionales los que lleven el seguimiento de la alimentación y el ejercicio de los usuarios.

En cuanto a la funcionalidad de poder obtener la información nutricional de las recetas es una funcionalidad que creemos que es muy útil que *Cookpad* en este caso no ofrece. Por otro lado, *Fit n Fed* ofrecerá un chat privado para poder interactuar de una manera más directa y privada con otros usuarios, cosa que ninguna de las otras dos aplicaciones ofrece.

Si comparamos los modelos de negocio observamos que ambas aplicaciones siguen el modelo *Freemium*, es decir, son gratuitas, pero ofrecen una suscripción mensual.

La principal diferencia entre sus modelos de negocio reside principalmente en que *Cookpad* contiene anuncios mientras que *Calisteniapp* no. Otra diferencia es en los tipos de suscripciones, *Cookpad* solo ofrece una suscripción mensual de 1.99€ mientras que *Calisteniapp* ofrece tres posibilidades: una suscripción mensual de 9.99€, una trimestral de 24.99€ y finalmente una suscripción anual con un coste de 69.99€.

## 3.2. Análisis DAFO

El análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades) es una herramienta de estudio sobre la situación de un proyecto analizando su situación interna y externa.

Las características internas del proyecto se representan mediante las debilidades y fortalezas de este y su situación externa se representa mediante las amenazas y las oportunidades del proyecto.

### Debilidades

- Falta de experiencia en el desarrollo de aplicaciones
- Falta de experiencia en proyectos de gran envergadura
- Recursos limitados

### Amenazas

- Aplicaciones que puedan sustituir partes de nuestra aplicación
- Aumento de la competencia

### Fortalezas

- Poco coste de desarrollo
- Diseño intuitivo y minimalista
- Descarga gratuita
- Buena coordinación del equipo
- Producto de uso diario

### Oportunidades

- Mercado en auge
- Amplio espectro de consumidores
- Unión de dos sectores en una misma aplicación.

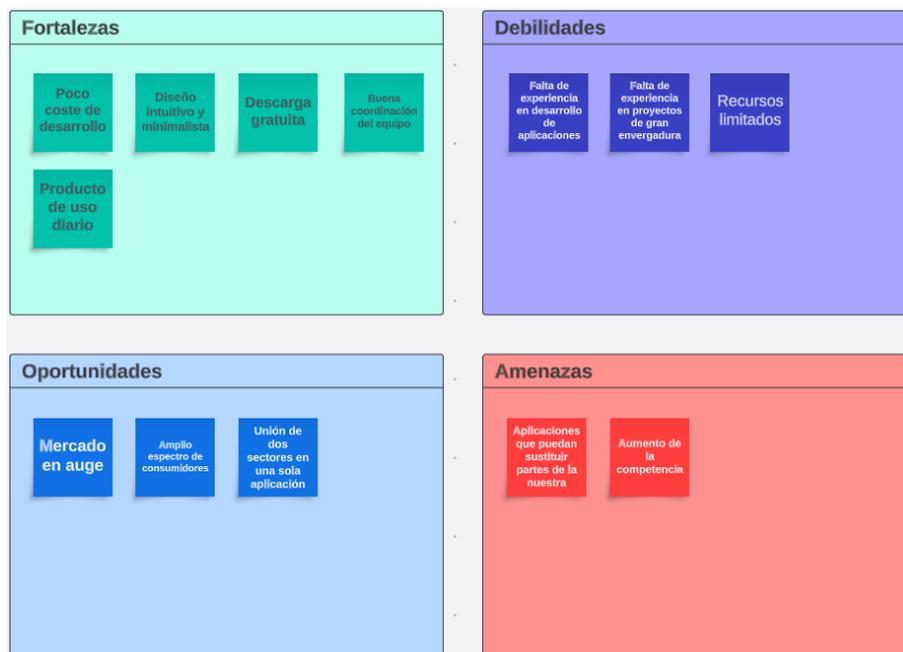


Figura 7: Análisis DAFO de Fit n Fed.

En cuanto a las características internas, observamos que nuestra mayor debilidad es la falta de experiencia tanto en desarrollo de aplicaciones móviles como en el desarrollo de un proyecto bastante ambicioso como el de una red social *Fitness*. Por otro lado, nuestras mayores fortalezas son que no se necesita mucho coste en el desarrollo inicial de la aplicación y al ser un producto de uso diario podremos mantener a los usuarios activos.

Por otro lado, en lo que respecta a la situación externa observamos que una gran oportunidad es que cada vez la gente está más interesada en comer sano y hacer deporte y nuestra aplicación une los dos sectores, pero el hecho de que sea un sector en auge hace que existan aplicaciones que cumplan funcionalidades de nuestra aplicación.

### 3.3. Modelo de negocio y proyección económica

El modelo de negocio y la proyección económica son unos de los aspectos más importantes para el desarrollo de un proyecto exitoso. Para ello estimaremos y analizaremos los ingresos y gastos del proyecto durante los próximos 5 años. Con ello estudiaremos la viabilidad del proyecto y podremos estimar los beneficios de este.

*Fit n Fed* se basa en modelo de negocio *freemium*, es decir, es una aplicación totalmente gratuita con funcionalidades bloqueadas o limitadas, las cuales se podrán obtener al pagar una suscripción mensual.

Por ello contaremos con dos fuentes de ingresos principales. La primera fuente de ingresos son los anuncios. Cuando el usuario no ha pagado la suscripción premium deberá ver anuncios para poder realizar diferentes funcionalidades.

Por otro lado, la segunda fuente de ingresos es una suscripción premium que costará 4.99€ al mes. Con ella se desbloquearán diferentes funcionalidades, como tener más de una planificación semanal para ir intercambiándolas cuando el usuario quiera y obtener la información nutricional de las recetas sin tener que ver un vídeo publicitario, dado que con la suscripción premium se eliminan los anuncios dentro de la aplicación.

Para realizar la estimación de ingresos se presupone que durante el primer año la aplicación seguirá en desarrollo y los únicos usuarios serán los early adopters de la aplicación y amigos y familiares que irán probándola para ayudarnos a que salga al mercado lo mejor posible.

Para el crecimiento de usuarios hemos realizado una estimación inicial de 1000 usuarios el primer trimestre después de lanzarlo al mercado y un aumento del 200% de usuarios al trimestre durante el primer año en el mercado. A partir de ahí se estabiliza y aumenta un 50%, excepto en el segundo trimestre de cada año que aumenta un 75% la cantidad de usuarios, ya que invertimos más en publicidad dado que llega el verano y esto nos beneficia al ser una aplicación con la que ayuda a mejorar el aspecto físico.

En cuanto a los ingresos calculamos que ingresaremos 0.09€ de publicidad por usuario en un trimestre y que tendremos un 10% de usuarios con suscripción premium.

En la Figura 9 podemos observar los ingresos durante los 2 primeros años.

Número de Trimestre	1	2	3	4	5	6	7	8
<b>Tipos de suscripciones acumuladas</b>								
Suscripciones mensuales durante el trimestre	Año 1 / T1	Año 1 / T2	Año 1 / T3	Año 1 / T4	Año 2 / T1	Año 2 / T2	Año 2 / T3	Año 2 / T4
	0	0	0	0	100	300	900	2700
<b>Usuarios de la aplicación</b>								
Cantidad de usuarios trimestral	25	38	56	84	1000	3000	9000	27000
<b>Ingresos Trimestrales</b>								
(Ventas nuevas + Renovaciones) * Precio	0,00 €	0,00 €	0,00 €	0,00 €	499,00 €	1.497,00 €	4.491,00 €	13.473,00 €
(Cantidad de usuarios no prémium) * Anuncios	2,25 €	3,38 €	5,06 €	7,59 €	90,00 €	270,00 €	810,00 €	2.430,00 €
<b>Total Ingresos</b>	<b>2,25 €</b>	<b>3,38 €</b>	<b>5,06 €</b>	<b>7,59 €</b>	<b>589,00 €</b>	<b>1.767,00 €</b>	<b>5.301,00 €</b>	<b>15.903,00 €</b>

Figura 8: Resumen de ingresos de los dos primeros años de vida de Fit n Fed.

Si siguiendo el mismo crecimiento al final del tercer año podríamos tener unos 159469 usuarios llegando a ingresar 93927.09€ en un trimestre llegando así a conseguir ingresar 3276532.93€ en el último trimestre del quinto año como se muestra en la siguiente figura.

Número de Trimestre	11	12	13	14	15	16	17	18	19	20
<b>Tipos de suscripciones acumuladas</b>										
Suscripciones mensuales durante el trimestre	Año 3 / T3	Año 3 / T4	Año 4 / T1	Año 4 / T2	Año 4 / T3	Año 4 / T4	Año 5 / T1	Año 5 / T2	Año 5 / T3	Año 5 / T4
	10631	15947	23920	41861	62791	94186	141279	247239	370858	556287
<b>Usuarios de la aplicación</b>										
Cantidad de usuarios trimestral	106313	159469	239203	418605	627908	941862	1412793	2472389	3708583	5562874
<b>Ingresos Trimestrales</b>										
(Ventas nuevas + Renovaciones) * Precio	53.049,94 €	79.574,91 €	119.362,36 €	208.884,13 €	313.326,19 €	469.989,29 €	704.983,94 €	1.233.721,89 €	1.850.582,83 €	2.775.874,24 €
(Cantidad de usuarios no prémium) * Anuncios	9.568,13 €	14.352,19 €	21.528,28 €	37.674,49 €	56.511,74 €	84.767,61 €	127.151,41 €	222.514,97 €	333.772,45 €	500.658,68 €
<b>Total Ingresos</b>	<b>62.618,06 €</b>	<b>93.927,09 €</b>	<b>140.890,64 €</b>	<b>246.558,62 €</b>	<b>369.837,93 €</b>	<b>554.756,90 €</b>	<b>832.135,35 €</b>	<b>1.456.236,86 €</b>	<b>2.184.355,28 €</b>	<b>3.276.532,93 €</b>

Figura 9: Resumen de los ingresos desde el tercer año hasta el quinto de Fit n Fed.

En cuanto a los gastos, observamos en la Figura 10 que el primer año, mientras la aplicación está en desarrollo no hay gastos, dado que los únicos desarrolladores seremos mi compañero Javier y yo y no necesitamos ningún gasto extra. En cuanto a la infraestructura en la nube usaremos la versión gratuita de Azure. A partir del segundo año contrataremos a un desarrollador Senior a un experto en ciberseguridad, la cual no puede faltar en una red social.

Número de Trimestre	1	2	3	4	5	6	7	8	9	10	11
<b>Gastos Anuales</b>											
Infraestructura Cloud	0 €	0 €	0 €	0 €	1.000 €	1.500 €	2.250 €	3.375 €	5.063 €	7.594 €	11.391 €
Ordenadores y equipamiento	0 €	0 €	0 €	0 €	4.000 €	0 €	0 €	0 €	2.000 €	0 €	0 €
Alquiler oficina e instalaciones	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €
Marketing	0 €	0 €	0 €	0 €	1.000 €	1.000 €	1.000 €	1.000 €	2.000 €	5.000 €	2.000 €
Gestoría	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €	0 €
Desarrolladores Senior	0 €	0 €	0 €	0 €	6.000 €	6.000 €	6.000 €	6.000 €	12.000 €	12.000 €	12.000 €
Ciberseguridad	0 €	0 €	0 €	0 €	6.000 €	6.000 €	6.000 €	6.000 €	6.000 €	6.000 €	6.000 €
<b>Total Gastos</b>	<b>0 €</b>	<b>0 €</b>	<b>0 €</b>	<b>0 €</b>	<b>18.000 €</b>	<b>14.500 €</b>	<b>15.250 €</b>	<b>16.375 €</b>	<b>27.063 €</b>	<b>30.594 €</b>	<b>31.391 €</b>
<b>Resultado Trimestral</b>											
	2,25 €	3,38 €	5,06 €	7,59 €	-17.411,00 €	-12.733,00 €	-9.949,00 €	-472,00 €	-3.208,00 €	11.151,63 €	31.227,44 €
<b>Resultado Trimestral Acumulado</b>											
	2,25 €	5,63 €	10,69 €	18,28 €	-17.392,72 €	-30.125,72 €	-40.074,72 €	-40.546,72 €	-43.754,72 €	-32.603,09 €	-1.375,66 €

Figura 10: Resumen de los gastos trimestrales hasta el final del año 3 de Fit n Fed.



En la siguiente figura podemos observar como a medida que pasa el tiempo aumentan los gastos en todos los ámbitos.

Número de Trimestre	11	12	13	14	15	16	17	18	19	20
	Año 3 / T3	Año 3 / T4	Año 4 / T1	Año 4 / T2	Año 4 / T3	Año 4 / T4	Año 5 / T1	Año 5 / T2	Año 5 / T3	Año 5 / T4
<b>Gastos Anuales</b>										
Infraestructura Cloud	11.391 €	17.086 €	25.629 €	38.443 €	57.665 €	86.498 €	129.746 €	194.620 €	291.929 €	437.894 €
Ordenadores y equipamiento	0 €	0 €	6.000 €	0 €	0 €	0 €	12.000 €	0 €	0 €	0 €
Alquiler oficina e instalaciones	0 €	0 €	6.000 €	6.000 €	6.000 €	6.000 €	12.000 €	12.000 €	12.000 €	12.000 €
Marketing	2.000 €	2.000 €	2.000 €	5.000 €	2.000 €	2.000 €	4.000 €	10.000 €	4.000 €	4.000 €
Gestoría	0 €	0 €	3.000 €	3.000 €	3.000 €	3.000 €	3.000 €	3.000 €	3.000 €	3.000 €
Desarrolladores Senior	12.000 €	12.000 €	24.000 €	24.000 €	24.000 €	24.000 €	48.000 €	48.000 €	48.000 €	48.000 €
Ciberseguridad	6.000 €	6.000 €	12.000 €	12.000 €	12.000 €	12.000 €	24.000 €	24.000 €	24.000 €	24.000 €
<b>Total Gastos</b>	<b>31.391 €</b>	<b>37.086 €</b>	<b>78.629 €</b>	<b>88.443 €</b>	<b>104.665 €</b>	<b>133.498 €</b>	<b>232.746 €</b>	<b>291.620 €</b>	<b>382.929 €</b>	<b>528.894 €</b>
<b>Resultado Trimestral</b>										
	31.227,44 €	56.841,16 €	62.261,73 €	158.115,26 €	265.172,89 €	421.259,34 €	599.389,01 €	1.164.617,35 €	1.801.426,02 €	2.747.639,04 €
<b>Resultado Trimestral Acumulado</b>										
	-1.375,66 €	55.465,50 €	117.727,23 €	275.842,50 €	541.015,39 €	962.274,73 €	1.561.663,74 €	2.726.281,08 €	4.527.707,11 €	7.275.346,14 €

Figura 11: Resumen de los gastos a partir del cuarto año de Fit n Fed.

Si observamos el primer trimestre del tercer año podemos ver que es último trimestre donde no obtendremos beneficios trimestrales y que tendremos una pérdida de 43754.72€, por lo que esta es la inversión mínima que debemos realizar para empezar el proyecto según la estimación.

Por otro lado, al terminar el tercer año observamos que por fin recuperaremos todo el dinero invertido y a partir de ese mes será rentable.

### 3.4. Lean Canvas

Otra herramienta estratégica para un proyecto de una empresa es el Lean Canvas. Esta herramienta permite analizar de manera sencilla y visual un modelo de negocio con el objetivo de aumentar el éxito del proyecto.

El *Lean Canvas* nos ayuda a entender de qué se trata nuestro proyecto y la viabilidad del negocio.

En la figura 12 se muestra el *Lean Canvas* de *Fit n Fed*.

<p><b>2 Problema</b></p> <p>La pandemia del COVID-19 ha agravado seriamente el bienestar físico de las personas, pues hay estudios y estadísticas que demuestran que los niveles de sedentarismo ha aumentado, a la par que también el consumo de productos con azúcar y ha disminuido el de frutas y verduras.</p> <p>Todo esto junto ocasiona que la gente haya empeorado su estado de forma y su alimentación, es por este motivo que creemos que es un gran momento para dar motivos y alentar a la gente a moverse y comer mejor, más equilibrado.</p>	<p><b>4 Solución</b></p> <p>Desarrollo de una red social móvil y multiplataforma, donde los usuarios puedan crear y ver tanto recetas como rutinas, a la vez que se comunican con otros usuarios de forma fácil y sencilla. También queremos ofrecer a los usuarios la posibilidad de planificar su semana.</p> <p><b>8 Métricas</b></p> <p>En cuanto a las métricas, serán simples. Utilizaremos el número total de usuarios y el número de usuarios premium.</p>	<p><b>3 Propuesta de valor</b></p> <p>Nuestra propuesta se basa en ofrecer a todas aquellas personas que buscan cambiar su alimentación y la manera en la que se ejercitan rutinas de ejercicio y recetas de cocina donde aprender y poner en práctica lo aprendido, a la vez que conocer gente que pueda estar en su misma situación, para ayudar a estas personas a cumplir sus objetivos y mejorar su bienestar físico.</p>	<p><b>9 Ventaja competitiva</b></p> <p>No existe una aplicación que junte de esta manera el fitness y la buena alimentación, es por esto que tenemos una ventaja clara respecto a otras apps del mercado.</p> <p><b>5 Canales</b></p> <p>Daremos a conocer la aplicación primero por conocidos, y después mediante una cuenta de TikTok y otras redes donde poder publicitar nuestra aplicación. Creemos que de esta manera podemos llegar a la mayoría de la población joven, y que sean los propios usuarios quien, por la naturaleza de la aplicación, difundan e inciten al uso de la aplicación a potenciales usuarios de mayor edad.</p>	<p><b>1 Clientes</b></p> <p>Los usuarios de la aplicación son todas aquellas personas que buscan mejorar su estilo de vida en relación a su calidad y calidad del ejercicio que realizan. Es por esto que el rango de edad de los usuarios es muy amplio.</p> <p>Los Early Adopters que probarán la aplicación en sus versiones anteriores a la salida de esta serán personas interesadas en la aplicación, es decir, son potenciales usuarios una vez se termine el desarrollo.</p> <p>Puesto que la aplicación es gratuita de uso, los usuarios finales no pagarán por ella, a menos que deseen pagar por la versión premium.</p>
<p><b>7 Costes</b></p> <p>El coste del proyecto es realmente el coste del desarrollo del proyecto, por lo que se necesitará costear únicamente alguien que se encargue del desarrollo (programador) y el servidor en la nube que hospede la base de datos de la aplicación.</p>		<p><b>6 Ingresos</b></p> <p>Los ingresos vendrán principalmente de los anuncios integrados en la aplicación, los cuales son evadibles mediante un pago mensual, es decir, una versión premium, que desbloquea mejoras a las funciones ya existentes dentro de la aplicación.</p>		

Figura 12: *Lean Canvas* de *Fit n Fed*.



### 3.5. Conclusiones de la evaluación

Analizando toda la información mencionada en los apartados anteriores podemos concluir que el proyecto es viable y rentable a largo plazo.

Primero de todo, nuestro producto es único en el mercado *fitness*, dado que a pesar de que existen varias aplicaciones de alimentación y deporte no existe ninguna que una ambas y junte lo mejor de los dos sectores.

En segundo lugar, observamos en el análisis DAFO que nuestras debilidades son de falta de experiencia por lo que al esperar un año a sacar el producto al mercado y seguir desarrollándolo ganaremos esa experiencia que nos falta. Por otro lado, nuestras amenazas son aplicaciones que cumplen funcionalidades de nuestra aplicación y que nuestra oportunidad es que no existe ninguna que cumpla con todas las características que ofrecemos.

Como vimos en la proyección económica la inversión mínima es de 43754.72€, pero no es nada comparado con los beneficios que puede llegar a ofrecer la aplicación a largo plazo. Por ello, el riesgo económico no es muy grande.

Teniendo en cuenta todo lo mencionado anteriormente, *Fit n Fed* es un proyecto capaz de generar una gran cantidad de ingresos, llegando a poder generar más de 2 millones de euros como beneficio en un trimestre.

En definitiva, *Fit n Fed* es un proyecto más que viable.

# 4. Desarrollo de la idea de negocio

En este capítulo veremos cómo se ha ido desarrollando la idea de negocio hasta convertirse en el MVP (mínimo producto viable) que es actualmente la aplicación. Veremos de qué características se compone el MVP y posibles nuevas características (aunque entraremos en profundidad en eso más adelante), los plazos y tiempos en los que se ha ido desarrollando la aplicación y que metodología hemos empleado para llevarlo a cabo. También se verá una pequeña prueba de la aplicación, seguida de una encuesta donde recibiremos *feedback* de un grupo reducido de *early adopters*.

## 4.1. Mapa de características

Un mapa de características es un pequeño gráfico donde se pueden ver la lista de funcionalidades de un proyecto, en este caso de *Fit n Fed*, nuestra aplicación. En dicho mapa encontraremos las características ya incluidas (*MUST*), las características a incluir (*SHOULD*), las que podrían llegar a estar (*COULD*) y las que se han descartado (*WONT*).



Figura 13: Mapa de características de Fit n Fed.

Una vez ya presentado el mapa de características, es momento de hablar de las funcionalidades de la aplicación. En este capítulo únicamente hablaremos de las ya presentes, pues el resto serán valoradas en futuros capítulos. Marcaremos las ya hechas con un subrayados, para que quede claro el punto de partida del TFG. También veremos los requisitos funcionales relacionados con las características que han sido implementadas para la memoria del proyecto.

Dicho esto, vamos allá con las características ya incluidas en *Fit n Fed*:

- **Iniciar sesión**: característica del tipo *What You See Is What You Get*, es la manera del usuario que se ha registrado de entrar en la aplicación.
- **Registro usuario**: función que permite a un nuevo usuario crear una cuenta personal introduciendo sus datos.
- **Feed principal**: primer tipo de *feed*. Consiste en una lista de todos los posts ordenados por tiempo de creación de la gente a la que sigues, en caso de no seguir a nadie aparece vacío.
- **Feed rutinas**: tercer tipo de *feed*. Al igual que el *feed* recetas, este *feed* muestra rutinas de usuarios no bloqueados, de nuevo ordenadas por tiempo de creación.
- **Feed recetas**: segundo tipo de *feed*. En este *feed* se visualizan las recetas creadas por cualquier tipo de usuario no bloqueado, ordenadas por tiempo.
- **Buscadores**: herramienta que sirve a los usuarios para poder encontrar publicaciones y usuarios mediante palabras clave y filtros.
- **Gestión de rutinas**: compendio de acciones donde el usuario puede crear y eliminar sus rutinas. Para crear una rutina se especifican los ejercicios a realizar, el número de repeticiones y series y la descripción de esta, al igual que el nombre y la dificultad.
- **Gestión de recetas**: al igual que el anterior, permite crear y eliminar recetas. Para crear una receta se añade el nombre, dificultad, ingredientes, cantidad de ingredientes, descripción y pasos a realizar.
- **Gestión de difusiones**: crear y eliminar difusiones. Para crear una difusión simplemente hace falta rellenar el texto a publicar.
- **Gestión de usuario**: conjunto de acciones que van desde la edición de los datos personales de un usuario hasta el cierre de sesión y la posibilidad de borrar tu cuenta de usuario. También se puede bloquear y desbloquear a otros usuarios, además de poder consultar la lista de bloqueados.
- **Comentarios**: posibilidad de comentar en publicaciones de otros usuarios, al igual que una difusión, simplemente se rellena el texto a comentar y se envía el comentario.

- **Guardar posts:** acción que permite a un usuario guardar publicaciones ajenas en un espacio especial, donde poder consultarlas más adelante, ya sea para planificarlas o simplemente volver a verlas.
- **Añadir contenido multimedia:** característica referida a la adición de fotos y/o vídeos dentro de una publicación. En el MVP únicamente se podrá cambiar la foto de la receta/rutina, pero más adelante se pretende mejorar a añadir todo tipo de contenido auxiliar.
- **Interacción con otros usuarios:** funcionalidad que permite a los usuarios seguir, dejar de seguir y bloquear y desbloquear a otros usuarios, todo esto se lleva a cabo entrando al perfil de otro usuario, lo que se realiza haciendo clic en su foto de perfil en una publicación o buscando su perfil en el buscador.
- **Interfaz amigable:** como ya hemos explicado anteriormente, queremos que la interfaz sea funcional, agradable y fácil de entender.
- **Planificación semanal:** el usuario puede acceder mediante el *flyout* desplegable a un área donde puede consultar que rutinas y/o recetas tiene asignadas a que día de la semana. Desde la página principal de la planificación se puede ver un pequeño resumen con la cantidad de rutinas y/o recetas asignadas a qué día de la semana, una vez dentro del día se pueden ver las horas concretas en las que están planificados cada uno de los elementos. Además, se puede editar la hora y quitar de la planificación desde dentro de la propia interfaz. Finalmente, existen sets de planificación, que permite tener planificadas 1 o más semanas, de manera que sean intercambiables. Los requisitos funcionales asociados a esta característica son visualizar planificación, ver resumen de planificación, comprobar día planificado, planificar publicación, editar hora planificación, eliminar publicación de la planificación y acceder a publicación planificada.
- **Valorar posts:** nueva característica que permite a los usuarios poner nota a una publicación, con valores que varían de 0.5 en 0.5, con un mínimo de 0 y un máximo de 5. Un usuario puede valorar una publicación todas las veces que quiera, sin embargo, solamente será la última valoración la que cuente. La valoración total de una publicación se muestra en la página principal de esa misma publicación, con un contador de estrellas desde el 0 al 5, con precisión decimal. Los requisitos son visualizar valoración media, valorar publicación, revalorar publicación.

- **Cálculo de macronutrientes:** permite a los usuarios visualizar los macronutrientes asociados a una receta, tanto en cantidad relativa (100g), como en cantidad total. Estos datos se muestran en la propia página de la receta, por debajo de la dificultad y la valoración. Por último, los macronutrientes se calculan automáticamente según los ingredientes elegidos para la receta. En cuanto a los requisitos, tenemos visualizar información nutricional relativa y visualizar información nutricional total.
- **Ajustes:** apartado accesible desde el *flyout* desplegable, donde el usuario puede comprobar la lista de usuarios bloqueados (y desbloquearlos o ver su perfil), borrar su cuenta permanentemente, cambiar el modo de ordenación de las recetas/rutinas entre tiempo de publicación y valoración y cambiar el tema de la aplicación (claro/oscuro). La pestaña de ajustes contendrá estas opciones para el MVP, no obstante, pretendemos que sea más completa en el futuro. Los requisitos asociados a los ajustes son visualizar ajustes, visualizar modo de ordenación por tiempo y por relevancia, cambiar de tema, ver lista de bloqueados y eliminar cuenta.
- **Multiplataforma:** .NET MAUI nos brinda la capacidad de hacer que la aplicación sea completamente multiplataforma, aunque ya explicaremos cómo más tarde. Sin embargo, pese a que el MVP está pensado para ser multiplataforma, no hemos podido probar Fit n Fed en *iOS*, pues no disponemos de las herramientas necesarias para ello.

## 4.2. Cronología del proyecto

En esta sección del capítulo 4, explicaremos cómo ha evolucionado la organización del proyecto con el tiempo, qué metodología hemos empleado y cómo hemos planificado y ejecutado el trabajo en el proyecto a lo largo de un período determinado.

Después de revisar y explicar las características del proyecto que forman parte del MVP, es importante describir cómo hemos estructurado el desarrollo para cumplir con las fechas establecidas. También mencionaremos un pequeño experimento que realizaremos más adelante, que incluirá una prueba de la aplicación y una breve encuesta.

Tras consultar con nuestro tutor del TFG, llegamos a la conclusión de que debíamos emplear una metodología ágil para llevar a cabo el proyecto. Optamos por SCRUM, ya que teníamos experiencia previa con ella durante nuestros años académicos. Establecimos tres *sprints*, cada uno con una duración de tres semanas. Acordamos que, al finalizar cada *sprint*, llevaríamos a cabo una reunión para revisar lo que se había logrado y considerar posibles mejoras.

En cuanto a la programación temporal de los tres *sprints*, el *Sprint 1* abarcó desde el 5 de junio hasta el 25 de junio; el *Sprint 2* se desarrolló desde el 26 de junio hasta el 16 de julio; y finalmente, el *Sprint 3* transcurrió desde el 17 de julio hasta el 6 de agosto.

Con la metodología y los *sprints* establecidos, era el momento de crear un *backlog* que incluyera las Unidades de Trabajo (UT) correspondientes. Para esta tarea, necesitábamos una plataforma de gestión que facilitara la organización. Durante nuestros estudios, habíamos explorado diversas herramientas como Trello o Worki. Sin embargo, dado que nuestra base de datos estaba alojada en Azure, decidimos utilizar esta última para organizar el proyecto. Creamos los tres *sprints* y comenzamos a crear UT relacionadas con las características que debíamos implementar.

Es importante destacar que *Fit n Fed* fue una aplicación cuyo desarrollo empezó durante la asignatura de PIN (Proyecto de Ingeniería del Software), por lo que ya contábamos con algunas características implementadas. Como resultado, el trabajo del TFG se centró en las nuevas funcionalidades de la aplicación, ya que aspectos como la creación de recetas y rutinas, así como los diferentes *feeds*, ya formaban parte de la aplicación desde el inicio. Dado que este proyecto ya estaba organizado previamente, tuvimos que "migrar" las UT previamente añadidas a Worki. Sin embargo, solo tuvimos en cuenta las UT que estaban en el *backlog*, ya que estábamos enfocados en darle un giro diferente a la aplicación, transformándola de un proyecto académico en una idea de negocio que pudiéramos explotar.

Una vez establecido cómo organizaríamos el proyecto, procedimos a crear las UT necesarias para las características que debíamos implementar, junto con el proceso por el cual pasarían las Unidades de Trabajo durante su desarrollo. A continuación, describiremos las etapas que decidimos que seguirían las UT durante la fase de desarrollo.



- **Registrar:** la primera etapa, aquí se detalla la UT y se asigna al *backlog*.
- **Especificar Requisitos:** Aquí se especifica que componentes tiene la UT y que es necesario para completarla a la hora de llevar a cabo el trabajo.
- **Esperar Sprint:** Una vez la UT está especificada y registrada, queda en este estado hasta que se asigne a un *Sprint* concreto.
- **Diseñar:** En esta etapa se discute cómo se va a llevar a cabo la UT, a nivel de implementación.
- **Programar:** Con los requisitos definidos y el diseño hecho, es el momento de implementar la UT
- **Pruebas:** Ahora que la UT está implementada, es momento de hacer pruebas de aceptación, para comprobar que lo programado funciona correctamente. En caso de que las pruebas fallen se deberá volver a la etapa anterior.
- **Terminada:** La última etapa, cuando las pruebas son correctas se pasa a esta etapa, donde se cierra la UT y se considera terminada.

Optamos por estas etapas de trabajo debido a nuestra experiencia previa en la carrera, a pesar de que Azure no permitiera una especificación tan detallada de las etapas (en Azure se trabaja con las categorías *To Do*, *Doing* y *Done*). Preferimos trabajar de esta manera con anotaciones en las Unidades de Trabajo (UT) ya que nos resultaba más cómodo.

Una vez que definimos cómo funcionarían las etapas dentro de las UT, llegó el momento de asignar estas tareas a los diferentes *sprints*. Para el primer *sprint*, planificamos la tarea de "Planificación Semanal" y "Valoración de Publicaciones". Sin embargo, después de nuestra primera reunión con el tutor, se nos recomendó realizar mejoras en la "Planificación Semanal". A pesar de que la tarea estaba lo suficientemente completa como para cerrarla, surgió la idea de mejorarla significativamente y le dimos prioridad sobre otras tareas, como las "Notificaciones".

En cuanto al segundo *sprint*, logramos completar las tareas de "Ajustes" y "Macronutrientes" con éxito. Sin embargo, en la reunión posterior al *sprint*, llegamos a la conclusión de que los "Ajustes", a pesar de estar terminados, tenían un mayor potencial de mejora del que habíamos considerado previamente. Recibimos comentarios valiosos de nuestro tutor y agregamos una UT adicional en el *backlog* para mejorar los "Ajustes". Como resultado, "Espacio Privado" y "Notificaciones" quedaron fuera del alcance en este *sprint*.

Finalmente, en el tercer y último *sprint*, trabajamos en las mejoras de la "Planificación" y "Ajustes" respectivamente, siguiendo las recomendaciones recibidas. Una vez finalizadas, y dado que las fechas no permitían una reunión en persona, enviamos un breve video al tutor para que diera su opinión sobre el trabajo realizado en el último *sprint*. Con esto, concluyó la fase de desarrollo de la aplicación por el momento.

### 4.3. Prueba con *early adopters*

Una vez ya acabado el desarrollo, llegaba el momento de probar Fit n Fed con usuarios realistas, de diferentes edades, ya que nuestro público objetivo puede tener un rango bastante amplio de edades. Se les facilitó la aplicación a usuarios de Android durante unos 5 días, donde se les pidió que usaran la aplicación sin darles indicaciones más allá de que siguieran al resto de encuestados y crearan al menos una rutina o una receta.

Para esta pequeña prueba se preparó una breve encuesta compuesta de 10 preguntas de respuesta cerrada (una valoración del 1 al 5, siendo el 5 muy de acuerdo y el 1 muy en desacuerdo) junto con una pregunta de respuesta abierta para que los *early adopters* comentaran que les ha parecido la aplicación y si echan algo en falta, ya sean funcionalidades, aspectos de la interfaz, etc.

En cuanto a las preguntas, esta es la selección que hemos escogido:

Pregunta	Muy en desacuerdo	En desacuerdo	NS/NC	De acuerdo	Muy de acuerdo
¿Piensas que la aplicación es fácil de usar?	1	2	3	4	5
¿Piensas que la aplicación ha repercutido/ puede repercutir de forma positiva para ti?	1	2	3	4	5
¿Recomendarías esta aplicación a alguien que busca mejorar su alimentación/ejercicio?	1	2	3	4	5
¿Te parece útil que una receta tenga sus macronutrientes?	1	2	3	4	5
En caso de que usaras esta aplicación regularmente ¿Pagarías la mensualidad?	1	2	3	4	5
En caso de ser alguien con una buena alimentación y/o que hace ejercicio ¿Te resulta útil la aplicación?	1	2	3	4	5
¿Crees que podrías usar esta aplicación de forma regular?	1	2	3	4	5
¿Te gustaría que las rutinas/recetas tuvieran vídeos instructivos para hacerlas?	1	2	3	4	5
¿Consideras que 4,99€ es un precio justo por la mensualidad de la aplicación?	1	2	3	4	5
¿Te gustaría poder contactar con profesionales relacionados con los ámbitos de la aplicación desde dentro de la propia aplicación?	1	2	3	4	5

Figura 14: Tabla de la encuesta sobre la aplicación Fit n Fed

Una vez preparada la encuesta, dimos la aplicación a un grupo de 8 usuarios de diferentes edades: 4 hombres y 4 mujeres, con edades de 22 (2 hombres y 2 mujeres), 56, 55, 57 y 53 años.

De la muestra, hay encuestados que practican deporte a diario y encuestados que no; a su vez, hay encuestados que cuidan mucho su alimentación y encuestados que la descuidan un poco más. Hemos intentado que pese a ser una muestra pequeña, haya potenciales usuarios que provengan de diferentes situaciones.

Ahora ya con la encuesta terminada y la muestra de encuestados, pasamos la aplicación a los posibles usuarios y tras 5 días de uso les pasamos la encuesta y les dimos unas horas para contestarla, pues facilitamos el sondeo de manera digital.

De todos los encuestados, las respuestas medias (redondeadas al entero más cercano) en las encuestas son las siguientes:

- ¿Piensas que la aplicación es fácil de usar? 4.
- ¿Piensas que la aplicación ha repercutido/ puede repercutir de forma positiva para tí? 3.
- ¿Recomendarías esta aplicación a alguien que busca mejorar su alimentación/ejercicio? 4.
- ¿Te parece útil que una receta tenga sus macronutrientes? 4.
- En caso de que usaras esta aplicación regularmente ¿Pagarías la mensualidad? 4.
- En caso de ser alguien con una buena alimentación y/o que hace ejercicio ¿Te resulta útil la aplicación? 4.
- ¿Crees que podrías usar esta aplicación de forma regular? 4.
- ¿Te gustaría que las rutinas/recetas tuvieran vídeos instructivos para hacerlas? 3.
- ¿Consideras que 4,99€ es un precio justo por la mensualidad de la aplicación? 3.
- ¿Te gustaría poder contactar con profesionales relacionados con los ámbitos de la aplicación desde dentro de la propia aplicación? 4.

En cuanto a la pregunta de respuesta abierta, lo que más destaca en general entre los encuestados son propuestas de nuevas características o mejoras de las actuales. De todas las propuestas de este tipo, la más repetida es la adición de notificaciones a la aplicación, pues actualmente no hay ningún tipo de notificaciones.

También existen encuestados que proponen añadir más personalización a la aplicación, no en lo referido a temas de estilos, si no a poder cambiar el color principal naranja de la aplicación a otro a gusto del usuario.

Por otro lado, existe un encuestado que opina que le hubiera gustado poder tener más tiempo para probar la aplicación, pero que también estaría bien añadir más tipos de ejercicios e incluso algo similar a una monitorización de la rutina mientras esta se lleva a cabo.

Los últimos dos encuestados proponen que les gustaría poder enviar publicaciones directamente desde la aplicación, mediante algún tipo de chat privado o mensaje directo.

Todas las respuestas abiertas han sido evaluadas y tomadas en cuenta, así pues, valoraremos implementarlas a lo largo del tiempo de vida de la aplicación.

## 5. Tecnologías y herramientas

---

En este capítulo, describiremos de manera breve las herramientas y tecnologías clave que empleamos en el desarrollo del proyecto.

Al inicio del proyecto, exploramos diversas opciones para las tecnologías a utilizar en la aplicación, dado que existen múltiples alternativas para desarrollar aplicaciones móviles multiplataforma. Nuestra primera propuesta consistía en emplear *React Native* para el *Front-End* con el uso de HTML, CSS y JavaScript y Java puro para la API del *Back-End*.

*React Native* es un framework de código abierto ampliamente utilizado para desarrollar aplicaciones multiplataforma, es decir, abarcan tanto Android, iOS, macOS, Web como Windows, aprovechando las características nativas de las plataformas respectivas. Es notablemente popular y es utilizado por aplicaciones conocidas como Instagram, Facebook, Discord y Pinterest, entre otras.

Si bien esta primera opción nos resultó atractiva debido a su capacidad para aprovechar las características nativas de los sistemas operativos, la mantuvimos como una alternativa viable, pero continuábamos explorando posibilidades. Nos interesaba la idea de explorar un lenguaje nuevo que no hubiéramos utilizado previamente y distanciarnos un poco de las soluciones estándar.

Durante nuestra búsqueda de otras tecnologías, nos encontramos con *Xamarin*, la propuesta de Microsoft para el desarrollo multiplataforma. *Xamarin* compartía muchas similitudes con *React Native* en términos de su enfoque en el uso de las características nativas.

Sin embargo, nuestra búsqueda nos llevó a descubrir .NET MAUI, que era la evolución de *Xamarin.Forms* de Microsoft para el desarrollo multiplataforma. Este framework estaba recién nacido, ya que salió al público apenas un mes antes de que lo encontráramos. El hecho de ser una tecnología novedosa no nos desanimó, ya que cumplía con todos los criterios que buscábamos en una tecnología para nuestro proyecto. Finalmente, basamos nuestra aplicación en .NET MAUI y C#, utilizando Visual Studio 2022 como nuestro entorno de desarrollo integrado (IDE).

Nos dimos cuenta .NET MAUI no nos permitía trabajar con una base de datos directamente y que para ello necesitaríamos crear una API REST. Así que decidimos que para el *Back-End* usaríamos ASP.NET Core para montar el API dado que al usar C# no necesitaríamos aprender un nuevo lenguaje y además podemos usar la misma solución en Visual Studio para ambos proyectos, es decir, para el *Front-End* y para el *Back-End*.

### 5.1. Visual Studio 2022





## 5.2. ASP.NET Core

ASP.NET Core es básicamente la versión de código abierto de ASP.NET, el cual es un marco multiplataforma de alto rendimiento cuya finalidad es compilar aplicaciones que están habilitadas para usarse en la nube y que están conectadas a internet.

Elegimos usar esta tecnología para crear la *Web API HTTP RESTful*, dado que Visual Studio nos permite hacerlo de manera muy sencilla, ya que este IDE nos da una plantilla de una *Web API* ya creada, por lo que partimos de una base estructural.

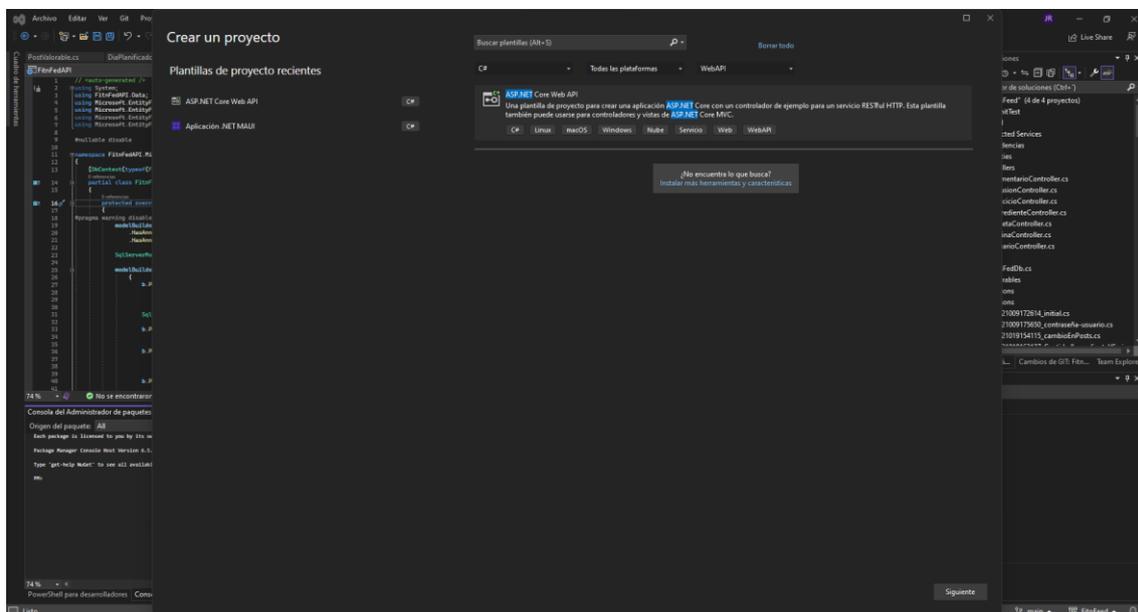


Figura 16: Opción para crear una Web API con ASP.NET Core en Visual Studio 2022.

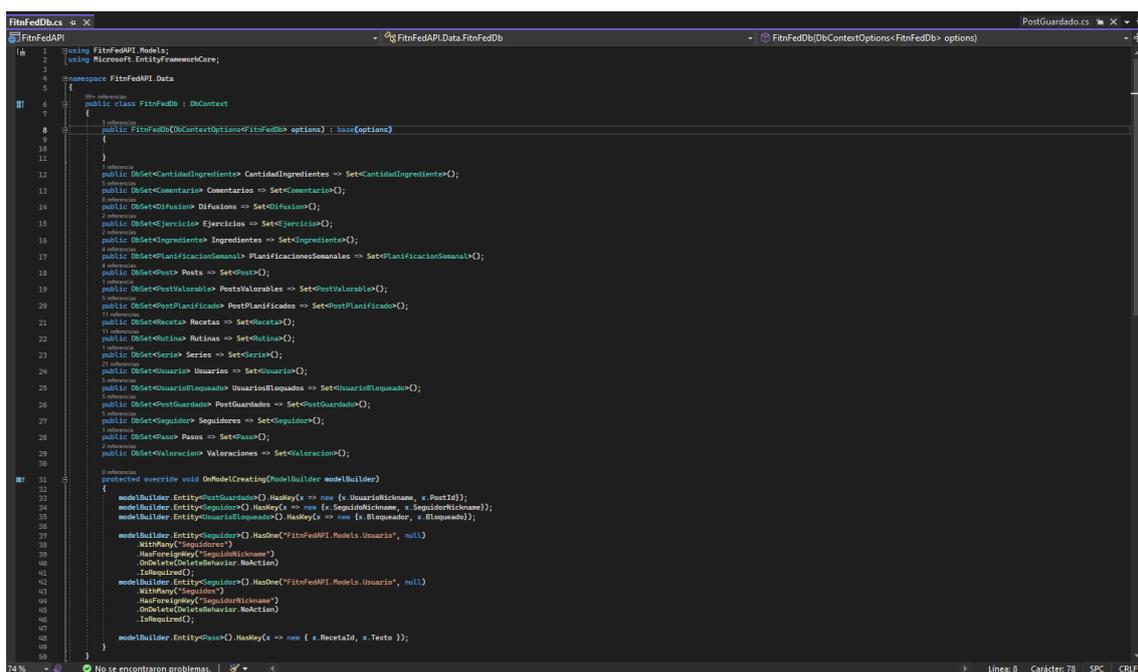
## 5.3. Entity Framework Core

Entity Framework Core es la versión de código abierto y multiplataforma de Entity Framework.

Este *framework* nos permite manejar de forma sencilla e intuitiva una base de datos relacional usando objetos .NET, dado que autogenera parte del código de acceso a datos que normalmente es necesario desarrollar.

La base de Entity Framework son los modelos y el contexto de la base de datos. Los modelos se corresponden con las clases .NET que representan los objetos que queremos guardar en la base de datos. Por otro lado, el contexto es una clase que representa la configuración de la base de datos. En el contexto se define la conexión a la base de datos, los modelos que deseamos guardar y configuración extra como definir claves primarias de varios valores y claves ajenas.

Además, la consulta a la base de datos se realiza usando *Language-Integrated Query (LINQ)*. Este lenguaje utiliza una sintaxis de consulta declarativa que nos permite realizar cualquier operación de consulta minimizando el código necesario a escribir respecto la sintaxis de consulta tradicional de las bases de datos SQL.



```
1 using FitnFedAPI.Models;
2 using Microsoft.EntityFrameworkCore;
3
4 namespace FitnFedAPI.Data
5 {
6     [DbContext]
7     public class FitnFedDb : DbContext
8     {
9         public FitnFedDb(DbContextOptions<FitnFedDb> options) : base(options)
10        {
11        }
12        public DbSet<CantidadIngrediente> CantidadIngredientes => Set<CantidadIngrediente>();
13        public DbSet<Comentario> Comentarios => Set<Comentario>();
14        public DbSet<Difusion> Difusiones => Set<Difusion>();
15        public DbSet<Ejercicio> Ejercicios => Set<Ejercicio>();
16        public DbSet<Ingrediente> Ingredientes => Set<Ingrediente>();
17        public DbSet<PlanificacionSemanales> PlanificacionesSemanales => Set<PlanificacionSemanales>();
18        public DbSet<Post> Posts => Set<Post>();
19        public DbSet<PostValarable> PostValarables => Set<PostValarable>();
20        public DbSet<PostPlanificado> PostPlanificados => Set<PostPlanificado>();
21        public DbSet<Receta> Recetas => Set<Receta>();
22        public DbSet<Rutina> Rutinas => Set<Rutina>();
23        public DbSet<Serie> Series => Set<Serie>();
24        public DbSet<Usuario> Usuarios => Set<Usuario>();
25        public DbSet<UsuariosBloqueados> UsuariosBloqueados => Set<UsuariosBloqueados>();
26        public DbSet<PostGuardado> PostGuardados => Set<PostGuardado>();
27        public DbSet<Seguidor> Seguidores => Set<Seguidor>();
28        public DbSet<Paso> Pasos => Set<Paso>();
29        public DbSet<Valaracion> Valaraciones => Set<Valaracion>();
30
31        protected override void OnModelCreating(ModelBuilder modelBuilder)
32        {
33            modelBuilder.Entity<PostGuardado>().HasKey(x => new { x.UsuarioId, x.PostId });
34            modelBuilder.Entity<Seguidor>().HasKey(x => new { x.SeguidorId, x.UsuarioId });
35            modelBuilder.Entity<UsuariosBloqueados>().HasKey(x => new { x.UsuarioId, x.BloqueadoId });
36            modelBuilder.Entity<Seguidor>().HasOne("FitnFedAPI.Models.Usuario", null)
37                .WithMany("Seguidores")
38                .HasForeignKey("SeguidorId")
39                .OnDelete(DeleteBehavior.NoAction)
40                .IsRequired();
41            modelBuilder.Entity<Seguidor>().HasOne("FitnFedAPI.Models.Usuario", null)
42                .WithMany("Seguidores")
43                .HasForeignKey("SeguidorId")
44                .OnDelete(DeleteBehavior.NoAction)
45                .IsRequired();
46            modelBuilder.Entity<Paso>().HasKey(x => new { x.RecetaId, x.Texto });
47
48        }
49    }
50 }
```

Figura 17: Clase FitnFedDb. Representa el contexto de la base de datos.



## 5.4. SQL Server

A la hora de crear la base de datos teníamos claro que queríamos que fuera una base de datos relacional y que fuera compatible con Entity Framework Core. Por ello, decidimos investigar diferentes sistemas de gestión de bases de datos relacionales como MySQL o PostgreSQL, pero no nos convencían ninguno de ellos.

Finalmente nos decidimos por SQL Server por varias razones. La primera razón es que es muy fácil integrar bases de datos SQL Server con Visual Studio y Entity Framework. La segunda es que existe una aplicación de Microsoft llamada Microsoft SQL Server Management Studio la cual es la mejor opción para realizar operaciones de mediante un entorno gráfico en la base de datos. Finalmente, la razón por la que nos hizo elegir definitivamente SQL Server es que teníamos la opción gratuita de tener esta base de datos en un servidor de Azure por lo que siempre iba a estar disponible su acceso y nos permite configurar diferentes parámetros de configuración y seguridad.

Id	Titulo	Imagen	Texto	Fecha	UsuarioNickna...	Discriminator	PostId	TipDeComida	Dificultad	Rutina_Dificult...	SumaRepeticio...	Valoracion	Votes
225	Amezc con tomate	NULL	Amezc y un poc...	2022-12-15 04:0...	Bobo	Receta	NULL	0	NULL	11.5	3.875	4	
226	no se	<Binary data>	jaja	2022-12-15 07:2...	Juan	Rutina	NULL	NULL	NULL	0	NULL	NULL	NULL
227	Prueba	<Binary data>	hgf	2022-12-15 07:2...	Juan	Receta	NULL	0	0	NULL	NULL	NULL	NULL
228	Amezc con jamón	NULL	Amezcito con ja...	2022-12-15 11:6...	Visitantes	Receta	NULL	0	2	NULL	NULL	NULL	NULL
229	Receta nuev...	NULL	sulfurohidru...	2022-12-15 11:2...	Visitantes	Receta	NULL	0	1	NULL	NULL	NULL	NULL
230	Edicom receta	NULL	Receta de prueba	2022-12-15 11:4...	Visitantes	Receta	NULL	0	1	NULL	NULL	NULL	NULL
231	Receta HP	NULL	Descripcion sem...	2022-12-15 12:0...	Visitantes	Receta	NULL	0	0	NULL	NULL	NULL	NULL
232	Rutina sofa	NULL	En el sofa se pu...	2022-12-15 12:0...	Visitantes	Rutina	NULL	NULL	1	NULL	NULL	NULL	NULL
233	La rutina poder...	NULL	De locos, unas l...	2022-12-15 12:1...	Visitantes	Rutina	NULL	NULL	2	NULL	NULL	NULL	NULL
234	No hombre	NULL	esta es una rec...	2022-12-15 12:3...	Visitantes	Rutina	NULL	NULL	0	4.40157187709...	4.40157187709...	1	
235	Pover electron...	NULL	La mejor receta?	2022-12-15 12:3...	Visitantes	Receta	NULL	0	0	10.44447123825...	3.481480453884...	3	
236	me cago en la ...	NULL	lo que sea	2022-12-23 16:1...	Juan	Comentario	228	NULL	NULL	NULL	NULL	NULL	NULL
237	me cago en la ...	NULL	me cago en la ...	2023-06-19 09:5...	equinu	Diffusion	NULL	NULL	NULL	NULL	NULL	NULL	NULL
238	Feramps	NULL	amezc con pegam...	2023-09-04 16:4...	feramps	Receta	NULL	0	2	0.299897909164...	0.299897909164...	1	
239	NULL	NULL	hai cogido tal...	2023-09-04 16:4...	feramps	Comentario	238	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 18: Vista de la tabla Posts en Microsoft SQL Server Management Studio.

## 5.5. Microsoft Azure

Microsoft Azure es una plataforma de Microsoft que nos ofrece una gran variedad de servicios y soluciones para nuestro proyecto y nos permite resolver diferentes problemas que teníamos.

En nuestro caso, Azure nos permite tener en su nube tanto la base de datos SQL Server como la Web API de forma gratuita por limitada durante un año. Gracias a ello, podemos acceder a ambos recursos en cualquier momento con total disponibilidad.

Otro motivo por el que usamos Azure es porque nos permite usar su servicio Azure DevOps. Este servicio ofrece una gran cantidad de funcionalidades útiles para llevar a cabo el desarrollo de nuestro código.

Así mismo, Azure DevOps ofrece un repositorio con gestión de versiones, el cual es indispensable para el desarrollo de cualquier proyecto.

Otras funcionalidades interesantes de Azure DevOps son las relacionadas con la planificación del proyecto. Permite crear diferentes unidades de trabajo (UTs), problemas encontrados y épicas, agrupaciones de UTs, además de poder planificar los *Sprints* del proyecto.

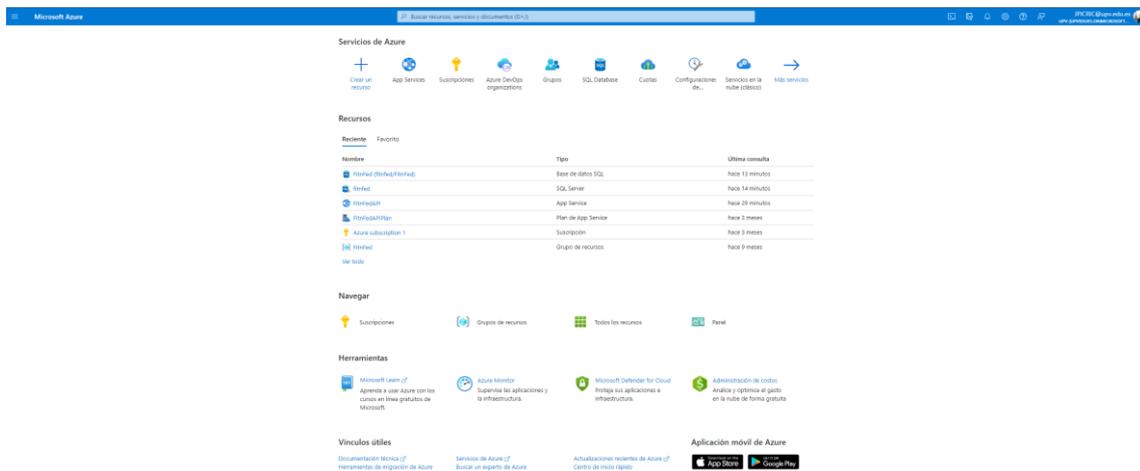


Figura 19: Vista de la interfaz de inicio del portal de Azure.

## 5.6. Swagger

Swagger es un conjunto de herramientas de código abierto que nos permite documentar, diseñar, construir y utilizar servicios web RESTful y que incluye automatización de la documentación, generación de casos de prueba y generación de código. Por todo ello, es imprescindible para nosotros esta herramienta.

En nuestro caso, al utilizar la plantilla de ASP.NET Core ya viene configurado y al iniciar nuestra web API podemos ver la documentación de esta en Swagger.

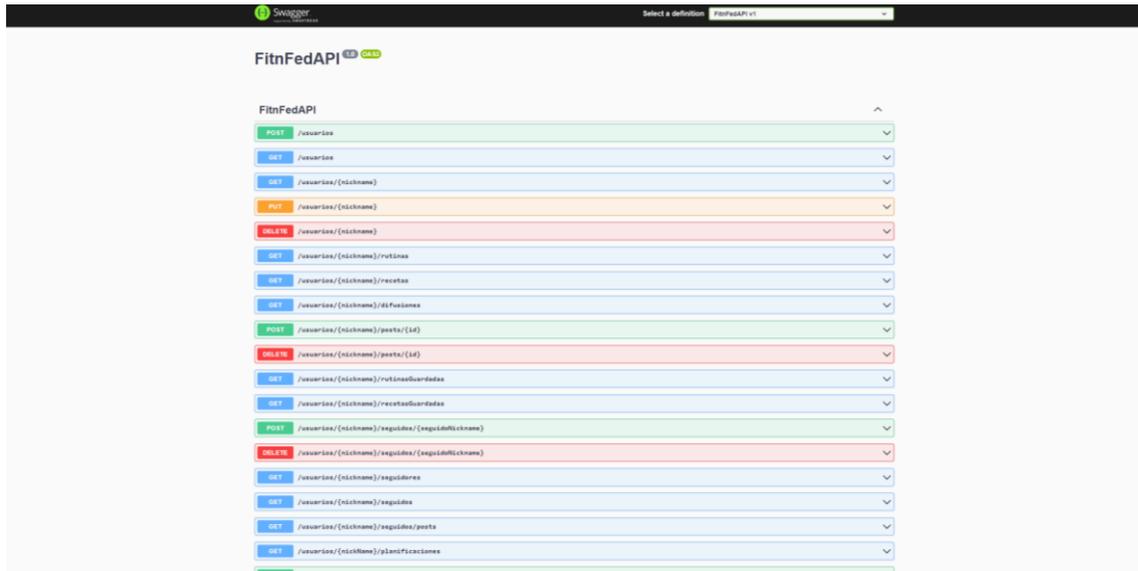


Figura 20: Swagger de la API de Fit n Fed

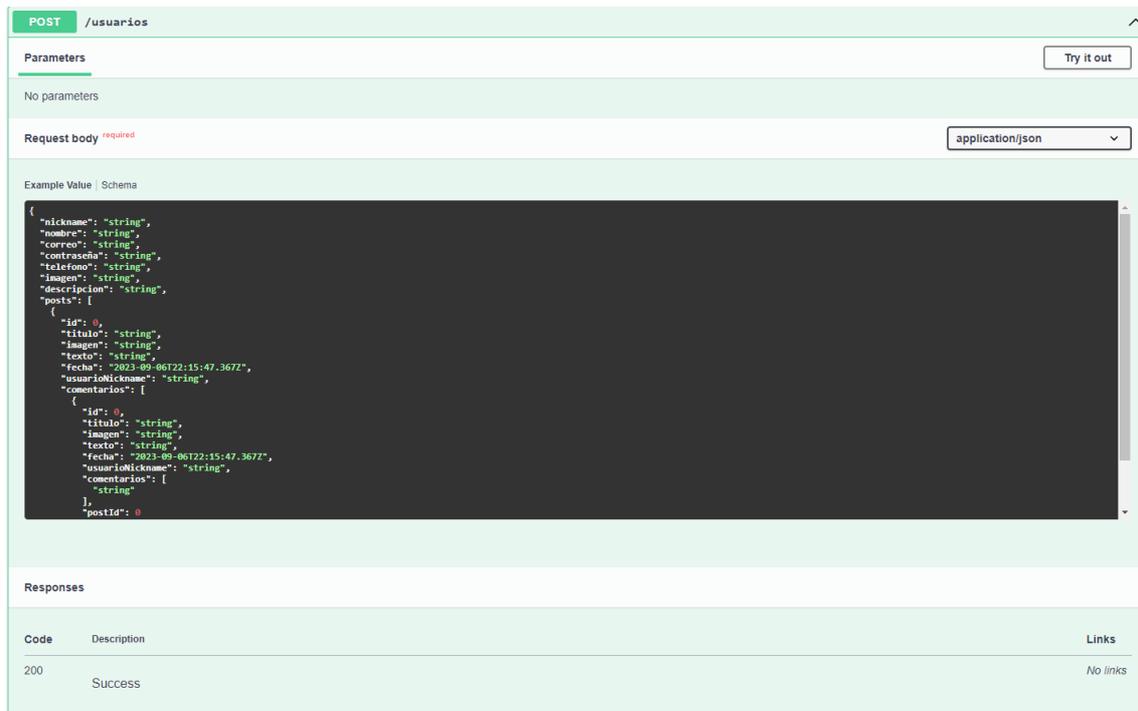


Figura 21: Vista desde Swagger del endpoint para introducir un nuevo usuario.

# 6. Desarrollo de la aplicación

---

En este capítulo explicaremos de manera breve cómo hemos realizado el desarrollo de la aplicación, en este caso del *Back-End*.

Al organizar el desarrollo en tres *sprints* diferentes la mejor manera de explicarlo es en orden cronológico, describiendo que se hizo en cada uno de ellos.

Pero antes de explicar el desarrollo realizado para este TFG debemos definir como era el estado de la API antes de empezar.

## 6.1. Estado inicial

Primero hay que explicar la estructura de la API de *Fit n Fed*. Esta consiste en una estructura de 3 capas:

- **Capa de acceso a datos:** Esta capa es la más interna y está gestionada en su mayoría por Entity Framework Core. En esta capa se definen los modelos y el contexto de la base de datos.
- **Capa de lógica de negocio:** En esta capa se define toda la funcionalidad lógica de Fit n Fed. Se han creado diferentes clases llamadas XXXXController donde XXXX sería un objeto o funcionalidad de la aplicación, por ejemplo, para controlar la lógica relacionada con las rutinas existe RutinasController.
- **Capa de interfaz de la API:** En esta capa es donde se encuentran definidos los *endpoints* de nuestra API.

Por otro lado, debemos de explicar el modelo de datos del cual partíamos:

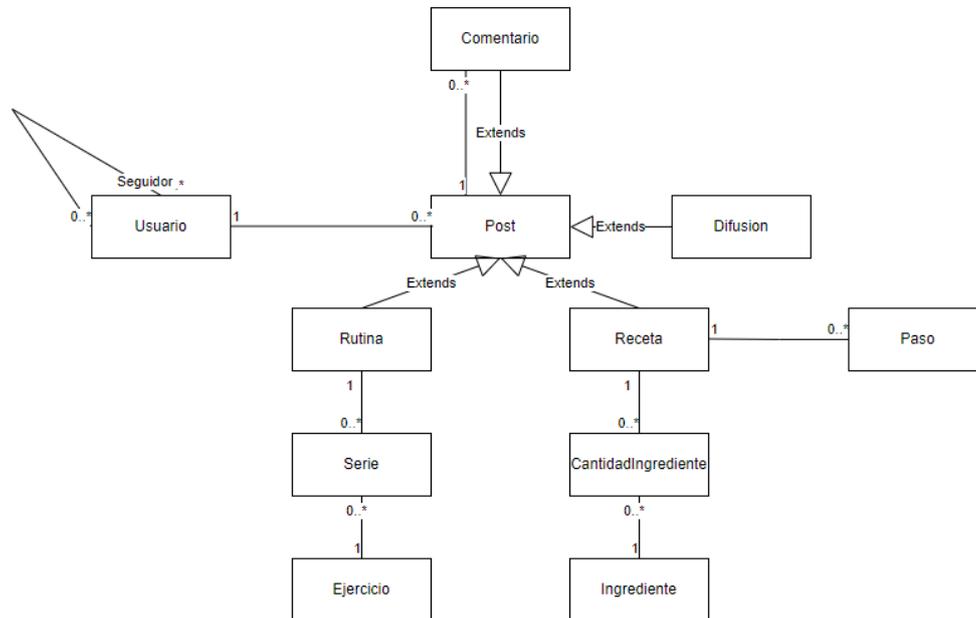


Figura 22: Modelo de datos inicial.

Como podemos observar en la Figura 22, el objeto principal es el usuario, el cual puede tener varios *posts*. Estos *posts* pueden ser difusiones, comentarios, rutinas o recetas.

- **Difusiones:** Son *posts* con únicamente texto y título.
- **Comentarios:** Son difusiones pero que pueden estar asociadas a otros *posts*.
- **Rutinas:** Son rutinas de entrenamiento las cuales pueden tener varias series de ejercicios.
- **Recetas:** Son recetas de cocina las cuales están formadas por diferentes pasos y diferentes cantidades de ingredientes.

Además, un usuario puede seguir a otros usuarios.

## 6.2. Primer Sprint

En el primer sprint nos centramos en la valoración de las rutinas y recetas y en empezar a desarrollar la planificación semanal, dado que eran los objetivos funcionales más importantes.

### 6.2.1. Valoración de publicaciones

Para añadir la valoración a las recetas y rutinas se valoraron varias opciones, pero al final se decidió por una opción bastante lógica. Crear una nueva clase llamada `PostValorable` que extiende a `Post`, la cual tendrá valoración media, la suma de todas las valoraciones y el número de valoraciones, y de la cual extenderán las clases `Rutina` y `Receta`.

```
namespace FitnFedAPI.Models
{
    4 referencias
    public abstract class PostValorable : Post
    {
        2 referencias
        public double Valoracion { get; set; }

        3 referencias
        public int Valoraciones { get; set; }

        5 referencias
        public double SumaValoraciones { get; set; }
    }
}
```

Figura 23: Clase `PostValorable`.

Parecía que con esto era suficiente pero no tuvimos en cuenta que era necesario guardar la información de que persona ha valorado cierto `post` para que si vuelve a valorar el mismo `post` no se suma esta valoración, si no que sustituya a la anterior. Para ello se crea un nuevo modelo llamado `Valoracion`, el cual sirve para guardar en base de datos la relación mencionada anteriormente.

```
using System.ComponentModel.DataAnnotations;

namespace FitnFedAPI.Models
{
    3 referencias
    public class Valoracion
    {
        [Key]
        0 referencias
        public int Id { get; set; }

        2 referencias
        public int IdPost { get; set; }

        2 referencias
        public string UserNickname { get; set; } = string.Empty;

        3 referencias
        public double Valor { get; set; }
    }
}
```

Figura 24: Clase `Valoracion`.

Para poder usar esta nueva funcionalidad solo fue necesario crear un único *endpoint* el cual nos permite valorar un *post* a partir de su id, la valoración y el *nickname* del usuario. En caso de que el usuario ya haya valorado el post, se sustituye su valoración actual y en caso de que no se crea un objeto Valoración nuevo. Después siempre se recalculan los parámetros del PostValorable.

The screenshot displays a REST client interface for the endpoint `POST /posts/{idPost}/valoracion`. The 'Parameters' section contains three required fields: `idPost` (integer, path), `valor` (number, query), and `nickName` (string, query). The 'Responses' section shows a 200 Success response with a JSON body:

```
{
  "id": 0,
  "titulo": "string",
  "imagen": "string",
  "texto": "string",
  "fecha": "2023-09-07T00:14:36.781Z",
  "usuarioNickname": "string",
  "comentarios": [
    {
      "id": 0,
      "titulo": "string",
      "imagen": "string",
      "texto": "string",
      "fecha": "2023-09-07T00:14:36.781Z",
      "usuarioNickname": "string",
      "comentarios": [
        "string"
      ]
    }
  ],
  "postId": 0
},
{
  "valoracion": 0,
  "valoraciones": 0,
  "sumaValoraciones": 0
}
```

Figura 25: Endpoint para valorar un post.

No fue necesario crear ninguno más dado que una vez valorado no se permite borrar la valoración y no tiene sentido crear un *endpoint* el cual devuelva un objeto Valoración, dado que en *Front-End* no existe este objeto dado que no tiene utilidad.

## 6.2.2. Planificación semanal

El objetivo de la planificación semanal es que un usuario pueda tener varias planificaciones semanales, la cuales pueden contener diferentes *posts* planificados entre los siete días de la semana. Para poder resolverlo se crearon dos nuevas clases: *PlanificaciónSemanal* y *PostPlanificado*.

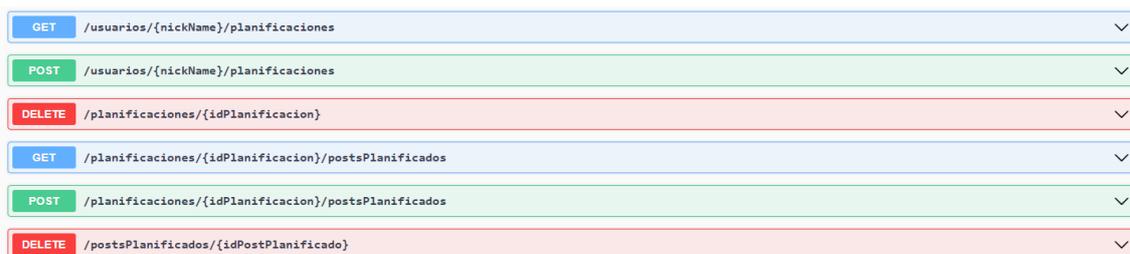
Primero se pensó en que *PostPlanificado* fuera un nuevo tipo de *Post* al igual que *PostValorable* pero esto llevaba a muchas complicaciones y no nos permitía manejar correctamente la información. Así que se decidió que *PostPlanificado* fuera una clase independiente y que solo tiene los parámetros necesarios: su *Id*, la *Id* del *post* planificado, la *id* de la planificación semanal con la que está relacionada, el día de la semana y la hora.

A su vez, *PlanificacionSemanal* debía contener su *Id*, el nombre de la planificación y el *nickname* del usuario del cual pertenece dicha planificación.

Justo en este punto nos dimos cuenta de que había un problema, y es que necesitábamos que la clase tuviera una lista por cada día de la semana para poder pasarle esta información correctamente al *Front-End* pero al añadir las listas a *PlanificacionSemanal* Entity Framework las detectaba como nuevas relaciones con *PostPlanificado* y este no sabía cómo gestionarlo.

Para solucionarlo, se encontró que existía una anotación que nos permitía decirle a Entity Framework que no queríamos que tuviera en cuenta las listas a la hora de crear la base de datos Esta anotación es [*NotMapped*].

Una vez creados los objetos necesarios se procedió a desarrollar los *endpoints* necesarios.



GET	/usuarios/{nickName}/planificaciones
POST	/usuarios/{nickName}/planificaciones
DELETE	/planificaciones/{idPlanificacion}
GET	/planificaciones/{idPlanificacion}/postsPlanificados
POST	/planificaciones/{idPlanificacion}/postsPlanificados
DELETE	/postsPlanificados/{idPostPlanificado}

Figura 26: Endpoints relacionados con la planificación semanal.

En este caso se crearon seis nuevos *endpoints*, tres para los *PostPlanificados* y tres para la *PlanificacionSemanal*. Los métodos *GET* sirven para obtener un *PostPlanificado* o una *PlanificacionSemanal* según sea el caso, los métodos *POST* para crear o actualizar dichos objetos y los métodos *DELETE* para eliminarlos.

## 6.3. Segundo Sprint

En este segundo *sprint* planeamos implementar otras dos nuevas características: los ajustes y los macronutrientes asociados a una receta.

### 6.3.1. Macronutrientes

Al principio, la idea era buscar una API externa la cual nos proporcionara la información que necesitábamos, pero todas las APIs eran de pago o gratuitas de forma limitada, así que a largo plazo nos tocaría pagar por ello. Después de investigar durante varios días se planteó la opción de añadir nosotros de forma manual o automatizada los datos necesarios en nuestra base de datos.

Finalmente, se decidió introducir de forma manual la cantidad de proteínas, grasas y carbohidratos que tiene cada alimento en nuestra base de datos. Al estar en una fase de desarrollo esta opción es viable dado que no necesitamos tener muchos ingredientes en *Fit n Fed* actualmente, pero en un futuro la mejor opción sería añadirlos a la base de datos de forma automatizada.

Por ello, dejamos de usar los ingredientes como un simple texto y pasamos a tener una clase más completa la cual tiene la información de los macronutrientes por cada 100 gramos.

Además, para poder enviar de forma sencilla la información al *Front-End* era necesario crear una nueva clase Macro, la cual contiene el nombre del macronutriente, la cantidad en 100 gramos y la cantidad total en la receta. Esta información se calcula utilizando los datos de los ingredientes que previamente hemos añadido, haciendo una suma del cálculo individual de cada ingrediente por la cantidad de este en la receta.

```

1 referencia
private static List<Macro> GetMacros(Receta receta, FitnFedDb db) {
    List<Macro> macros = new List<Macro>();
    Macro proteinas = new Macro { Name = "Proteinas" };
    Macro grasas = new Macro { Name = "Grasas" };
    Macro carbohidratos = new Macro { Name = "Carbohidratos" };
    Macro kcal = new Macro { Name = "Kcal" };

    foreach (var cantidadIngrediente in receta.CantidadIngredientes)
    {
        var ingrediente = cantidadIngrediente.Ingrediente;

        proteinas.Cantidad100g += ingrediente.Proteinas;
        proteinas.CantidadTotal += ingrediente.Proteinas * cantidadIngrediente.Cantidad / 100;
        grasas.Cantidad100g += ingrediente.Grasas;
        grasas.CantidadTotal += ingrediente.Grasas * cantidadIngrediente.Cantidad / 100;
        carbohidratos.Cantidad100g += ingrediente.Carbohidratos;
        carbohidratos.CantidadTotal += ingrediente.Carbohidratos * cantidadIngrediente.Cantidad / 100;
        kcal.Cantidad100g += ingrediente.Kcal;
        kcal.CantidadTotal += ingrediente.Kcal * cantidadIngrediente.Cantidad / 100;
    }

    return macros;
}

```

Figura 27: Método que calcula los macronutrientes de una receta.

Otra modificación que hubo que hacer fue en la clase receta, pues ahora debíamos añadir una lista de Macros.

En este caso no hizo falta crear ningún *endpoint* nuevo dado que solo hemos añadido información a la receta.

### 6.3.2. Ajustes

En cuanto a los ajustes de la aplicación se decidió empezar por bloquear usuarios y eliminar la cuenta.

Para eliminar la cuenta parecía que lo único que había que hacer es eliminar al Usuario de la base de datos, pero además de ello era necesario borrar cualquier otro objeto que tenga relación directa con el usuario que quiere eliminar su cuenta. Para ello serviría con configurar la base de datos con un borrado en cascada en las relaciones necesarias, pero no se consiguió hacer esto dado que Entity Framework nos avisaba de que podría haber problemas al hacerlo y no nos lo permitía hacer en las relaciones necesarias, así que se decidió programar un método que borra todo lo relacionado con un usuario.

```

1 referencia
public static async Task DeleteUsuarios(string nickname, FitnFedDb db)
{
    var usuario = await db.Usuarios.FindAsync(nickname);

    if (usuario is null) { throw new NotFoundException("No existe el usuario " + nickname); }

    db.Usuarios.Remove(usuario);

    var rutinas = await RutinaController.GetRutinasByUsuarioNickname(usuario.Nickname, db);
    foreach (var rutina in rutinas)
    {
        await RutinaController.DeleteRutina(rutina.Id, db);
    }

    db.Usuarios.Remove(usuario);
    var recetas = await RecetaController.GetRecetasByUsuarioNickname(usuario.Nickname, db);
    foreach (var receta in recetas)
    {
        await RecetaController.DeleteReceta(receta.Id, db);
    }
}

```

Figura 28: Parte del método que borra la cuenta de un usuario.

En este caso se creó un *endpoint DELETE* para eliminar al usuario a partir de su *nickname*.

En cuanto a la funcionalidad de bloquear usuarios iba a ser sencillo pero costoso. Primero deberíamos crear un modelo el cual nos sirva para introducir una nueva tabla en la base de datos que relacione a dos usuarios los cuales uno ha sido bloqueado por otro. A este nuevo modelo de datos lo hemos llamado *UsuarioBloqueado*.

Una vez hecho esto necesitamos definir tres *endpoints*. El primero es un *POST* que sirve para crear un *UsuarioBloqueado* y así bloquear a un usuario. Un método *GET* que devuelve los usuarios bloqueados de un usuario y un método *DELETE* que elimina el *UsuarioBloqueado* especificado desbloqueando así a dicho usuario.

POST	/usuarios/{nickName}/bloquear
GET	/usuarios/{nickName}/bloqueados
DELETE	/usuarios/{nickName}/bloqueados/{bloqueado}

Figura 29: Endpoints relacionados con bloquear y desbloquear usuarios.

Finalmente, debíamos usar esta información en todos los sitios donde haga falta, para que el usuario no pueda ver nada relacionado con los usuarios que tenga bloqueados.

```

15 referencias
public static async Task<List<Usuario>> GetUsuariosBloqueados(string nickName, FitnFedDb db)
{
    var usuariosBloqueados = await db.UsuariosBloquados.Where(ub => ub.Bloqueador == nickName).ToListAsync();
    var usuarios = new List<Usuario>();
    foreach (var usuarioBloqueado in usuariosBloqueados)
    {
        var usuario = await db.Usuarios.FindAsync(usuarioBloqueado.Bloqueado);

        if (usuario != null)
        {
            usuarios.Add(usuario);
        }
    }
    return usuarios;
}

```

Figura 30: Método para obtener los usuarios bloqueados

```

2 referencias
private static async Task<List<Receta>> SeleccionarRecetas(string nickName, List<Receta> recetas, FitnFedDb db)
{
    var usuarios = await UsuarioController.GetUsuariosBloqueados(nickName, db);

    foreach (var usuario in usuarios)
    {
        recetas.RemoveAll(r => r.UsuarioNickname == usuario.Nickname);
    }

    if (recetas.Count() <= 20)
        return OrdenarRecetas(recetas);
    else
        return OrdenarRecetas(recetas).Take(20).ToList();
}

```

Figura 31: Método para filtrar las recetas.

## 6.4. Tercer Sprint

El tercer *sprint* es el último del desarrollo del proyecto, y por lo tanto el último que se documentará en la memoria. Todo lo pendiente después del sprint quedará para trabajo futuro, que veremos más adelante.

Este último *sprint* contiene la implementación de las mejoras de la planificación semanal a la par que la de la implementación de nuevos ajustes.

### 6.4.1. Nuevos ajustes

En este sprint buscábamos ampliar el apartado de ajustes. Concretamente ahora el usuario puede decidir el tema de la aplicación y la forma en la que se ordenan los *posts* a la hora de usar el buscador de la aplicación, pudiendo ordenar por los mejor valorados por los más recientes.

Para ello se decidió crear una nueva clase llamada Ajustes. Esta clase estará destinada a contener toda la información sobre los ajustes de un usuario. Los atributos de esta clase serán su Id, el *nickname* del usuario al cual pertenecen, el tema de la aplicación preferido, el parámetro por el cual ordenar a la hora de usar el buscador y una lista con los usuarios bloqueados.

En cuanto al tema y al parámetro de ordenación se han creado dos enumerables nuevos, uno para cada ajuste.

```
namespace FitnFedAPI.Models
{
    0 referencias
    public class Ajustes
    {
        [Key]
        0 referencias
        public int Id { get; set; }
        0 referencias
        public string NickName { get; set; } = string.Empty;
        0 referencias
        public Tema Tema { get; set; }
        0 referencias
        public TipoOrdenacion TipoOrdenacion { get; set; }
        [NotMapped]
        0 referencias
        public List<Usuario> UsuariosBloqueados { get; set; } = new List<Usuario>();
    }
}
```

Figura 32: Nueva clase Ajustes.

Hay que tener en cuenta que todo usuario debe tener ajustes, así que hemos tenido que modificar el método de crear usuario para que cree automáticamente sus ajustes.

Una vez hecho esto, debemos de cambiar los métodos de ordenación de recetas y rutinas para que se ordenen conforme el usuario desea.

```

2 referencias
private static List<Rutina> OrdenarRutinas(TipoOrdenacion tipoOrdenacion, List<Rutina> rutinas)
{
    switch (tipoOrdenacion)
    {
        case TipoOrdenacion.Tiempo:
            rutinas.OrderByDescending(rutina => rutina.Fecha).ToList();
            break;
        case TipoOrdenacion.Valoracion:
            rutinas.OrderByDescending(rutina => rutina.Valoracion).ToList();
            break;
    }
    return rutinas.OrderByDescending(rutina => rutina.Fecha).ToList();
}

```

Figura 33: Método para ordenar las rutinas una vez encontradas.

Finalmente, se creó un único *endpoint* de tipo PUT el cual sirve para actualizar los ajustes del usuario.

## 6.4.2. Mejora de la planificación

Para la planificación se decidió que estaría bien obtener un pequeño resumen que muestre la cantidad de rutinas y recetas que tiene cada día de la planificación.

La forma de resolverlo fue crear una nueva clase llamada *DiaPlanificado*, el cual contendrá una lista de *PostPlanificados* y dos enteros que representarán la cantidad de recetas y rutinas que se haya planificado para ese día.

Así mismo, también debemos modificar las clases *PostPlanificado* y *PlanificacionSemanal*. En *PostPlanificado* debemos de añadir el tipo de post, si es rutina o receta, y en *PlanificacionSemanal* debemos cambiar las listas de *PostPlanificados* por varios *DiaPlanificados*.

Otra de las cosas que debemos cambiar es el método para añadir la información de la planificación semanal para que vaya sumando las recetas y rutinas según se van añadiendo los *posts* planificados.

```

1 referencia
public static async Task<List<PlanificacionSemanal>> GetPlanificacionesSemanales(string nickName, FitnFedDb db)
{
    var planificaciones = await db.PlanificacionesSemanales.Where(plani => plani.UserNickname == nickName).ToListAsync();

    foreach (var plani in planificaciones)
    {
        var posts = await GetPostsPlanificados(plani.Id, db);
        foreach (var post in posts)
        {
            switch (post.DayOfWeek)
            {
                case DayOfWeek.Monday:
                    plani.Monday.Posts.Add(post);
                    anyadirALResumen(post.TipoPost, plani.Monday);
                    break;
                case DayOfWeek.Tuesday:
                    plani.Tuesday.Posts.Add(post);
                    anyadirALResumen(post.TipoPost, plani.Tuesday);
                    break;
                case DayOfWeek.Wednesday:
                    plani.Wednesday.Posts.Add(post);
                    anyadirALResumen(post.TipoPost, plani.Wednesday);
                    break;
                case DayOfWeek.Thursday:
                    plani.Thursday.Posts.Add(post);
                    anyadirALResumen(post.TipoPost, plani.Thursday);
                    break;
            }
        }
    }
}

```

Figura 34: Método para obtener las planificaciones de un usuario.

```

7 referencias
private static void anyadirALResumen(TipoPost tipoPost, DiaPlanificado dia) {
    if (TipoPost.Rutina.Equals(tipoPost))
    {
        dia.Rutinas++;
    }
    else if (TipoPost.Receta.Equals(tipoPost))
    {
        dia.Recetas++;
    }
}

```

Figura 35: Método para añadir la información al resumen del día.

## 6.5. Estado final

Antes de terminar con la explicación del desarrollo de la API debemos comparar el estado final con el estado inicial antes de realizar el TFG y para ello vamos a exponer el modelo de datos actual de la aplicación y ver su evolución después de todo el desarrollo. Veremos como los usuarios y los *posts* siguen siendo el eje principal de la aplicación y como la planificación semanal tiene un gran protagonismo en *Fit n Fed*.

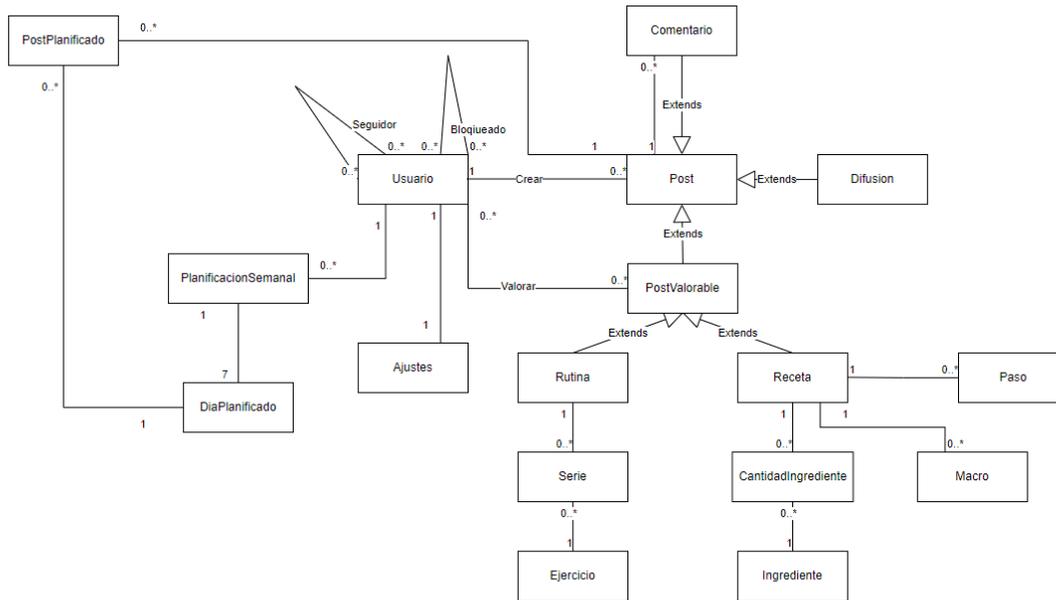


Figura 36: Modelo de datos actual

## 7. Conclusiones

El desarrollo de *Fit n Fed* ha sido una gran experiencia, dado que ha sido una oportunidad de aprender en profundidad como desarrollar una web API, desde los fundamentos de un servicio HTTP RESTful hasta la gestión de esta para el despliegue. Además, el desarrollar una aplicación de esta envergadura es un reto muy gratificante de superar.

*Fit n Fed* ha cumplido todas nuestras expectativas, ya que cumple todos los objetivos que se plantearon al principio. Incita a llevar un estilo de vida saludable y ayuda a cumplir los propósitos de la gente. Así mismo, se han implementado la mayoría de las funcionalidades planteadas, aplazando las más difíciles y la menos importantes para un futuro.

En cuanto al *feedback* de los potenciales clientes, siempre ha sido positivo. Además, los resultados de las encuestas indican que vamos por buen camino. Sin embargo, el proyecto acaba de empezar, por lo que todavía falta mejorar algunas funcionalidades y añadir nuevas.

Personalmente, desarrollar esta aplicación es el sueño de unos estudiantes que primero de carrera, cuando se nos planteó en Formación y Orientación de Empresas que creáramos nuestra propia empresa y nuestro propio proyecto, no podían imaginarse que acabaría siendo real, pues durante todos estos años nunca hemos abandonado la idea y hoy en día está más cerca que nunca de ser real.

Por último, esto no habría sido posible sin el apoyo de nuestros compañeros a lo largo de la carrera, desde FOE hasta PSW y PIN. Gracias a ellos esto pudo comenzar y haremos todo lo posible para que *Fit n Fed* pueda ver la luz.

## 8. Trabajo futuro

Desde una perspectiva empresarial, el futuro de *Fit n Fed* es muy prometedor y con un gran potencial de crecimiento. Nuestra estrategia se basa en aumentar la visibilidad de la aplicación en todos los canales disponibles para expandir nuestra base de usuarios y, en particular, atraer inversores interesados en impulsar el proyecto a un nivel más amplio.

Para lograr esto, planeamos crear perfiles en redes sociales, con un enfoque principal en TikTok, ya que creemos que es una plataforma efectiva para dar a conocer nuestra aplicación. Conforme aumente el número de usuarios, iremos introduciendo nuevas funcionalidades e incitaremos a los usuarios a que recomienden la aplicación, lo que nos ayudará a llegar a un público aún más amplio.

Confiamos en que, con el tiempo, los anuncios dentro de la aplicación generarán ingresos gradualmente, al igual que algunos usuarios optarán por suscribirse al modo premium para disfrutar de mejoras y eliminar los anuncios.

Nuestra visión es que *Fit n Fed* crezca de manera orgánica y sostenible, con una base de usuarios que brinde comentarios valiosos y esté satisfecha con la aplicación. Este crecimiento nos permitirá considerar la posibilidad de reducir la cantidad de anuncios en la aplicación en el futuro.

Además, todavía existen nuevas funcionalidades que planeamos implementar en un futuro, ya que no hemos podido añadir todas las características planteadas en el capítulo cuatro. Aspiramos a expandir *Fit n Fed* a nivel internacional, ofreciendo la aplicación en múltiples idiomas, como inglés, francés o italiano, con el objetivo de ayudar a las personas a alcanzar sus metas de salud y mejorar sus hábitos de manera gradual y saludable.



## 9. Bibliografía

- INE. *Instituto Nacional de Estadística*. (2021). Obtenido de <https://www.ine.es/>
- Información general de ASP.NET Core*. (2023). Obtenido de <https://learn.microsoft.com/es-es/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>
- Información general de Entity Framework Core*. (2023). Obtenido de <https://learn.microsoft.com/es-es/ef/core/>
- OMS. (2022). *Global Status on physical activity*. Obtenido de <https://apps.who.int/iris/bitstream/handle/10665/363560/9789240060449-eng.pdf?sequence=1&isAllowed=y>
- Swagger (Software) - Wikipedia, la enciclopedia libre*. (2022). Obtenido de [https://es.wikipedia.org/wiki/Swagger\\_\(software\)](https://es.wikipedia.org/wiki/Swagger_(software))

## 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				X
ODS 2. Hambre cero				X
ODS 3. Salud y bienestar	X			
ODS 4. Educación de calidad				X
ODS 5. Igualdad de género				X
ODS 6. Agua limpia y saneamiento				X
ODS 7. Energía asequible y no contaminante				X
ODS 8. Trabajo decente y crecimiento económico				X
ODS 9. Industria, innovación e infraestructuras				X
ODS 10. Reducción de las desigualdades				X
ODS 11. Ciudades y comunidades sostenibles			X	

ODS 12. Producción y consumo responsables			X	
ODS 13. Acción por el clima				X
ODS 14. Vida submarina				X
ODS 15. Vida de ecosistemas terrestres				X
ODS 16. Paz, justicia e instituciones sólidas				X
ODS 17. Alianzas para lograr objetivos				X

Los ODS u Objetivos de Desarrollo Sostenible son un conjunto de 17 metas a corto plazo planteadas por la UNESCO para la Agenda 2030. Estos objetivos detallan aspectos a mejorar hasta dicha fecha. Entre ellos, destaca uno por encima del resto a la hora de relacionarlo con el proyecto desarrollado, el ODS número 3, salud y bienestar.

Este objetivo trata la situación mundial frente a la pandemia del COVID-19, hablando de como la crisis epidémica global ha afectado negativamente a la salud de las personas y a su bienestar.

Esto precisamente es lo que intentamos paliar desde Fit n Fed, pues queremos ayudar a las personas a mejorar su salud mediante nuestra aplicación. De hecho nuestra motivación y objetivos principales tratan de este tema, pues queremos que Fit n Fed sirva a sus usuarios para luchar contra las consecuencias de la pandemia del COVID-19, mediante las recetas, rutinas y comunicación con otros usuarios.