

GRADO EN Grado en Ingeniería Informática de Gestión y Sistemas de Información

# TRABAJO FIN DE GRADO

## VeganMania: App para Controlar tu Alimentación



**Estudiante:** Elorza Gabilondo, David

---

**Director:** Bengoetxea Kortazar, Kepa Xabier

**Subdirector:** Perona, Balda, Iñigo

---

**Curso:** 2023-2024

**Fecha:** 28-06-2024

# Resumen Laburpena Abstract

Este proyecto presenta una aplicación de salud y alimentación que facilita el seguimiento de tu dieta y hábitos alimenticios. Con una extensa base de datos de recetas permite registrar fácilmente tus comidas y obtener un desglose detallado de las calorías, vitaminas y minerales que consumes. Especialmente está enfocado en personas veganas y/o vegetarianas que necesitan cambiar de hábitos alimenticios y es más complicado completar una dieta sana y completa. Para lograr este objetivo, la web cuenta con una calculadora nutricional personalizada que permite a los usuarios ingresar sus datos personales y objetivos de salud para hacer el seguimiento de la alimentación, tanto a nivel de macronutrientes como micronutrientes. Además, se proporciona un blog informativo con artículos actualizados sobre nutrición, ejercicios físicos y estilos de vida saludables. También se incluye una sección dedicada a la búsqueda de recetas, con alternativas fáciles de preparar utilizando ingredientes frescos, nutritivos y de temporada. Por último, podemos ver los productos de temporada nacionales para cada mes, tratando de favorecer el consumo local.

**Palabras Clave:** Alimentación vegana, nutrición personalizada, calculadora nutricional, recetas saludables, consumo local, aplicación web.

Proiektu honek zure dietaren eta elikadura-ohituren jarraipena egitea errazten duen osasun- eta elikadura-aplikazio bat aurkezten du. Errezeten datu-base zabal baten bidez, erraz erregistratzen dira zure janariak, eta xehekatu egiten dira kontsumitzen dituzun kaloriak, bitaminak eta mineralak. Bereziki, elikadura-ohiturak aldatu behar dituzten pertsona begano eta/edo begetarianoetara bideratuta dago, eta zailagoa da dieta osasuntsua eta osoa osatzea. Helburu hori lortzeko, webguneak nutrizio-kalkulagailu pertsonalizatua du, erabiltzaileei beren datu pertsonalak eta osasun-helburuak sartzeko aukera ematen diena, elikaduraren jarraipena egiteko, makronutrienteei zein mikronutrienteei dagokienez. Gainera, informazio blog bat eskaintzen da, nutrizioari, ariketa fisikoei eta bizimodu osasungarriei buruzko artikulak eguneratuekin. Horrez gain, errezetak bilatzeari buruzko atal bat dago, osagai freskoak, elikagarriak eta sasoiak erabiliz erraz prestatzeko aukerekin. Azkenik, hila-bete bakoitzerako sasoiako produktu nazionalak ikus ditzakegu, tokiko kontsumoa errazten saiatuz.

**Gako-hitzak:** Elikadura beganoa, nutrizio pertsonalizatua, nutrizio-kalkulagailua, errezeta osasungarriak, tokiko kontsumoa, web-aplikazioa.

This project presents a health and food application that makes it easy to track your diet and eating habits. With an extensive database of recipes it allows you to easily record your meals and get a detailed breakdown of the calories, vitamins and minerals you consume. It is especially focused on vegans and / or vegetarians who need to change their eating habits and it is more complicated to complete a healthy and complete diet. To achieve this goal, the website has a personalized nutritional calculator that allows users to enter their personal data and health goals to track food, both at the macronutrient and micronutrient level. In addition, an informative blog is provided with up-to-date articles on nutrition, physical exercise and healthy lifestyles. Also included is a section dedicated to finding recipes, with easy-to-prepare alternatives using fresh, nutritious and seasonal ingredients. Finally, we can see the national seasonal products for each month, trying to favor local consumption.

**Keywords:** Vegan food, personalized nutrition, nutritional calculator, healthy recipes, local consumption, web application.

# Índice

<b>Resumen Laburpena Abstract</b>	<b>1</b>
<b>Lista de figuras</b>	<b>6</b>
<b>Lista de tablas</b>	<b>8</b>
<b>Lista de acrónimos</b>	<b>10</b>
<b>Glosario</b>	<b>11</b>
<b>1. Introducción</b>	<b>12</b>
1.1. Descripción del proyecto . . . . .	12
1.2. Razones de la elección del TFG . . . . .	14
<b>2. Enfoque Inicial</b>	<b>16</b>
2.1. Conexión con los Objetivos de Desarrollo Sostenible y los principios y Valores Democráticos . . . . .	16
2.2. Objetivos . . . . .	17
2.3. Herramientas . . . . .	19
2.4. Alcance . . . . .	20
2.5. Planificación Temporal . . . . .	29
2.6. Análisis de Riesgos . . . . .	32
2.7. Descripción del presupuesto . . . . .	34
<b>3. Estado del Arte</b>	<b>40</b>
3.1. Proceso de Obtención de Datos . . . . .	40
3.2. Aplicaciones similares . . . . .	44
<b>4. Captura de Requisitos</b>	<b>45</b>

4.1. Documento de Requisitos . . . . .	45
4.2. Casos de Uso . . . . .	47
4.3. Modelo de Dominio . . . . .	49
<b>5. Análisis y diseño</b>	<b>52</b>
5.1. Planteamiento del Desarrollo de Trabajo . . . . .	52
5.2. Diagrama relacional de la Base de Datos . . . . .	52
5.3. Diagrama de Clases . . . . .	56
5.4. Arquitectura . . . . .	56
5.5. Diseño de Interfaces . . . . .	58
<b>6. Desarrollo</b>	<b>73</b>
6.1. Instalación . . . . .	73
6.2. Implementación . . . . .	76
6.3. Despliegue Web . . . . .	82
<b>7. Pruebas</b>	<b>83</b>
7.1. Pruebas de Flujo de Trabajo . . . . .	83
7.2. Pruebas de funcionalidad . . . . .	85
7.3. Pruebas con Usuarios Reales . . . . .	89
7.4. Pruebas de Seguridad . . . . .	89
<b>8. Conclusiones y Marco Futuro</b>	<b>92</b>
8.1. Objetivos . . . . .	92
8.2. Reflexión Personal . . . . .	93
<b>9. Trabajo Futuro</b>	<b>94</b>
9.1. Base de Datos . . . . .	94
9.2. Tecnologías . . . . .	94
9.3. App Móvil . . . . .	95
9.4. Soporte Multilingüe . . . . .	95
9.5. Recomendaciones Personalizadas . . . . .	95
9.6. Login mediante Google/Github/Facebook... . . . .	95
9.7. Blog Interactivo . . . . .	95



# Lista de figuras

1.	Número de casos de obesidad registrados en España de 2011 a 2022 [1] . . .	13
2.	Número de hombres que padecen sobrepeso por regiones del mundo [4] .	14
3.	EDT . . . . .	21
4.	Diagrama de Gantt . . . . .	30
5.	Diagrama de Casos de Uso . . . . .	47
6.	Diagrama de Casos de Uso . . . . .	48
7.	Diagrama Entidad-Relación VeganMania . . . . .	53
8.	Arquitectura Web VeganMania . . . . .	57
9.	Interfaz de Registro . . . . .	58
10.	Mail de Bienvenida una Vez Registrado y Verificado . . . . .	59
11.	Interfaz de Inicio de Sesión . . . . .	60
12.	Interfaz Solicitud Cambio de Contraseña . . . . .	60
13.	Mail para Restablecer Contraseña . . . . .	61
14.	Interfaz para Cambiar de Contraseña . . . . .	61
15.	Mail Verificar Cuenta . . . . .	62
16.	Interfaz que Completa la Verificación . . . . .	62
17.	Interfaz Sobre Nosotros . . . . .	63
18.	Interfaz Modificar Datos . . . . .	64
19.	Interfaz Blog . . . . .	65
20.	Interfaz Artículo del Blog . . . . .	65
21.	Interfaz de Recetas . . . . .	66
22.	Interfaz de Artículos de Temporada . . . . .	66
23.	Interfaz Añadir Comida . . . . .	67
24.	Interfaz Añadir Alimento . . . . .	68
25.	Interfaz Mis Comidas . . . . .	68

26.	Interfaz detalles de la comida . . . . .	69
27.	Interfaz Detalles Alimento . . . . .	69
28.	Interfaz Tu Seguimiento . . . . .	70
29.	Consultar objetivos 1 . . . . .	71
30.	Consultar objetivos 2 . . . . .	71
31.	Interfaz Tutorial . . . . .	72
32.	Lógica de Funcionamiento de la Aplicación . . . . .	79
33.	Resultado push Workflow . . . . .	85

# Lista de tablas

1.	Seguimiento del Proyecto . . . . .	22
2.	Definir Objetivos (Realización del DOP) . . . . .	22
3.	Definir Tareas (Realización del DOP) . . . . .	22
4.	Planificar Tareas (Realización del DOP) . . . . .	22
5.	Estimar Tiempos de las Tareas (Realización del DOP) . . . . .	23
6.	Desarrollar EDT (Realización del DOP) . . . . .	23
7.	Investigación trabajos previos . . . . .	23
8.	Captura de requisitos . . . . .	23
9.	Selección de herramientas . . . . .	24
10.	Identificación de riesgos . . . . .	24
11.	Búsqueda de Datos y/o APIs . . . . .	24
12.	Planificación temporal . . . . .	24
13.	Diseño de estructuras de datos . . . . .	25
14.	Diseño de interfaces . . . . .	25
15.	Diseño de Despliegue Web . . . . .	25
16.	Diseño de Lógica Web . . . . .	26
17.	Implementación de la lógica web . . . . .	26
18.	Implementación del Despliegue web . . . . .	26
19.	Implementación de las Interfaces Web . . . . .	27
20.	Reuniones de Seguimiento . . . . .	27
21.	Pruebas de Flujo de Trabajo . . . . .	27
22.	Pruebas de Funcionalidad . . . . .	28
23.	Pruebas de Usuario . . . . .	28
24.	Pruebas de Seguridad . . . . .	28
25.	Corrección de Errores . . . . .	28

26.	Estructuración de la memoria . . . . .	29
27.	Redacción de la memoria . . . . .	29
28.	Creación de la defensa . . . . .	29
29.	Resumen de tareas . . . . .	31
30.	Tabla de definición de riesgos . . . . .	32
31.	Riesgo: Análisis de nuevas funcionalidades . . . . .	32
32.	Riesgo: Falta de conocimientos . . . . .	32
33.	Riesgo: Servicios de terceros sin soporte . . . . .	33
34.	Riesgo: Fallos de seguridad en librerías de terceros . . . . .	33
35.	Riesgo: Enfermedad . . . . .	33
36.	Riesgo: No cumplir la planificación . . . . .	34
37.	Riesgo: Cambio de requisitos o planteamiento del proyecto . . . . .	34
38.	Pruebas para un usuario sin identificar . . . . .	86
39.	Pruebas para un usuario identificado PT.1 . . . . .	87
40.	Pruebas para un usuario identificado PT.2 . . . . .	88

# Lista de acrónimos

**TFG** Trabajo Fin de Grado

**XML** eXtensible Markup Language (Lenguaje de Marcado Extensible)

**CSV** Comma-Separated Values

**IMC** Índice de Masa Corporal

**BD** Base de Datos

**SGBD** Sistema de Gestión de Bases de Datos

**MV** Máquina Virtual

**SSL** Secure Sockets Layer

**IP** Internet Protocol

**API** Application Programming Interface

**SPA** Single Page Application

**ER** Entidad-Relación

**UX** User-eXperience

**UI** User-Interface

**DOP** Diagramas de Operaciones del Proceso

**URL** Uniform Resource Locators

**ODS** Objetivos de Desarrollo Sostenible

# Glosario

**Dataset** Conjunto de datos utilizado para entrenar y evaluar modelos de inteligencia artificial o algoritmos.

**Bug** Se usa para nombrar a los errores que se producen en un programa informático.

**Freemium** Es un modelo de negocio basado en la creación y prestación de un producto o servicio gratuito, pero que se convierte en pagado para tener algunas funcionalidades extra.

**Vulnerable/Vulnerabilidad** En informática, una vulnerabilidad es una debilidad existente en un sistema que puede ser utilizada por una persona malintencionada para comprometer su seguridad.

**Proxy** Es un intermediario entre un dispositivo cliente (como una computadora o un teléfono) y otro servidor en la red.

**Proxy Inverso** Es un tipo de proxy que se sitúa frente a uno o varios servidores web y dirige las solicitudes de los clientes a esos servidores. A diferencia de un proxy tradicional, que actúa en nombre de los clientes, el proxy inverso actúa en nombre de los servidores.

# 1. Introducción

En este capítulo se proporcionará una descripción del proyecto y las razones de su elección.

## 1.1. Descripción del proyecto

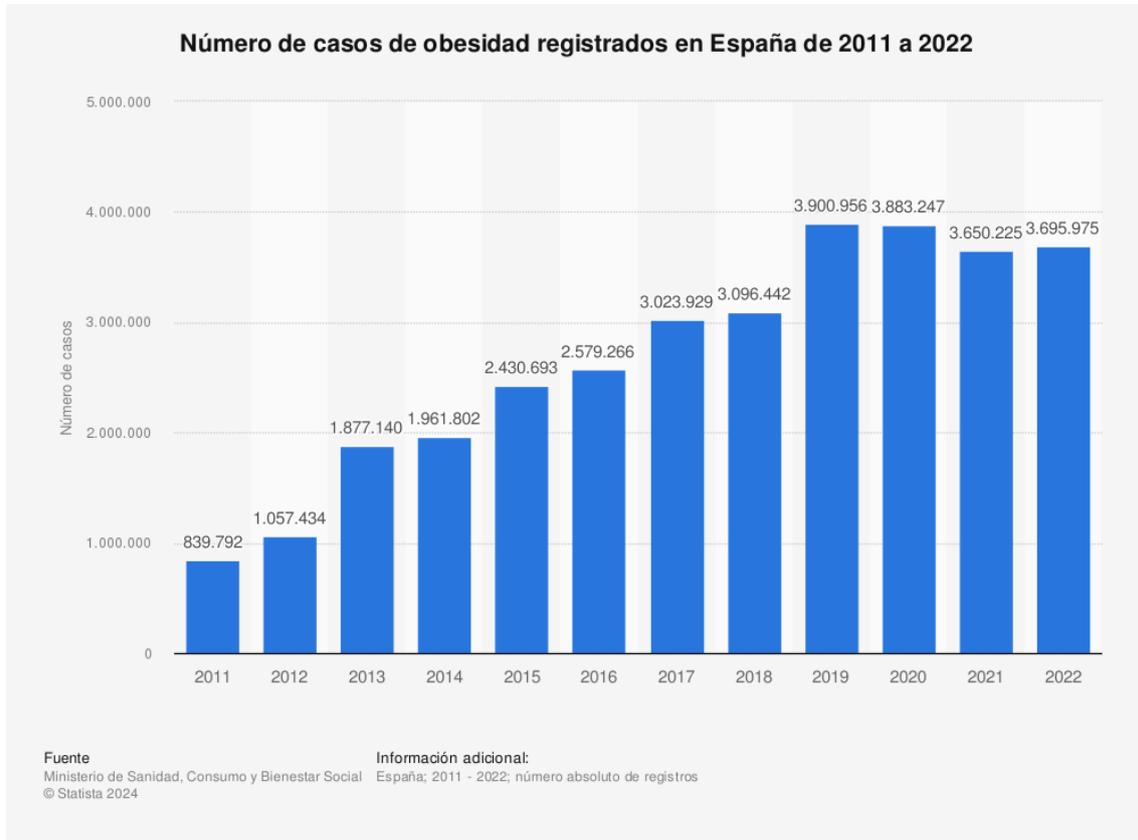
En un mundo cada vez más consciente de los desafíos ambientales y sociales a los que nos enfrentamos, el veganismo ha emergido como un estilo de vida que no solo se preocupa por la salud individual y los animales, sino que también está intrínsecamente relacionado con el impacto medioambiental. Esta dieta, al tener un menor impacto ambiental que la convencional, especialmente cuando se basa en productos de proximidad, mantiene una estrecha relación con los Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas.

El hacerse vegano o vegetariano es un desafío, ya que hay que cambiar las proteínas animales por otras vegetales, y hay que tomar un suplemento de vitamina B12. Además, las proteínas completas no se encuentran tan fácilmente. De igual modo, nos encontramos con que los productos sustitutivos de origen animal suelen tener precios elevados. Este problema se agrava en familias con bajos ingresos, donde la elección de alimentos suele inclinarse hacia opciones más económicas y menos saludables. Por ello, hemos desarrollado una página web que ofrece una solución integral para ayudar a los usuarios a seguir una dieta balanceada y cumplir con sus objetivos calóricos y nutricionales de manera efectiva con productos fácilmente disponibles a su alcance. La plataforma está diseñada para ser fácil de usar y accesible para personas con distintos niveles de conocimiento sobre nutrición.

Este Trabajo de Fin de Grado (TFG) se centra en la creación de una aplicación accesible y fácil de usar que realiza un seguimiento de los nutrientes obtenidos durante la semana, para poder mejorar la dieta llegando a tomar todos los nutrientes necesarios según la edad, sexo, nivel de actividad, entre otros factores, para lograr los objetivos según la persona. La aplicación está enfocada principalmente en personas que siguen una alimentación vegana/vegetariana, debido a la dificultad de mantener dicha alimentación por los precios elevados y la poca disponibilidad de elección. Las recomendaciones diarias de nutrientes han sido tomadas de las guías de la Organización Mundial de la Salud (OMS) y otras instituciones reconocidas en el campo de la nutrición.

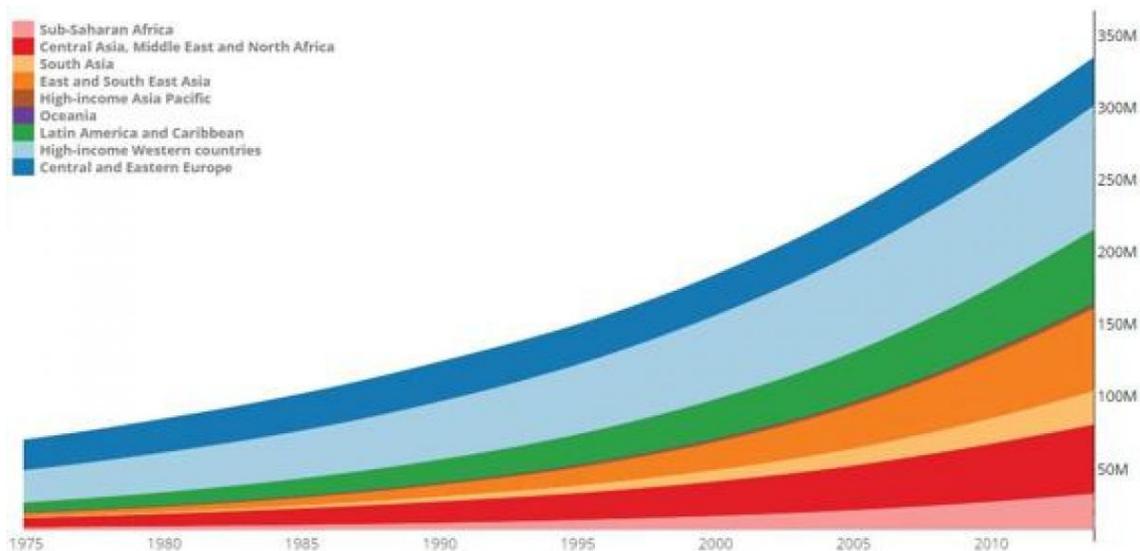
No obstante, también busca que cualquier persona pueda hacer uso de ella para mejorar su alimentación. Es un método también para luchar contra la alta cantidad de productos ultra-procesados y de baja calidad en lo que respecta a nutrientes. En los últimos años, la prevalencia del sobrepeso ha aumentado considerablemente en España, convirtiéndose en un problema de salud pública de gran envergadura. Según los datos

analizados por Statista (provenientes del Ministerio de Sanidad, Consumo y Bienestar Social [2]), el número de casos de sobrepeso ha mostrado una tendencia creciente desde 2011 hasta 2022. Como podemos observar en la figura 1, a pesar de haber un descenso respecto a 2019-2020 (marcada también por el confinamiento), ha habido un aumento notorio en relación con 2011.



**Figura 1:** Número de casos de obesidad registrados en España de 2011 a 2022 [1]

Además, otros estudios como el realizado por 'The Lancet'[3], muestran de igual modo la tendencia alcista a nivel mundial. Donde en cuarenta años la cifra de personas que padecen sobrepeso **se ha multiplicado por 500 al pasar de 105 millones en 1975 a 641 en la actualidad**. De estas, el 58% son mujeres y el 41% restante, hombres. De hecho, en 1975 la cantidad de personas con bajo peso duplicaba a los obesos. Entendiendo por obesidad a un IMC superior a 30 kg/m<sup>2</sup>. Como se puede apreciar en la figura 2.



**Figura 2:** Número de hombres que padecen sobrepeso por regiones del mundo [4]

Teniendo en cuenta las conclusiones de estos estudios y los riesgos que el sobrepeso y la obesidad entrañan para la salud, se resalta la urgencia de abordar este problema mediante intervenciones efectivas y accesibles que promuevan una alimentación saludable y equilibrada.

El software cuenta con una interfaz web que la hace fácilmente accesible y su utilización no requiere un conocimiento avanzado en informática. Además, permite a los usuarios el seguimiento de sus objetivos nutricionales con gran detalle, lo que resulta diferencial, ya que a pesar de que existen aplicaciones para lograr llegar a los objetivos, están más enfocadas al seguimiento calórico, y la gente vegana/vegetariana necesita otros detalles específicos como las vitaminas o minerales.

El diseño de la página es sencillo e intuitivo, y presenta información nutricional sobre los alimentos, donde se detallan hasta 31 nutrientes. Incluye una herramienta de seguimiento en la sección "Tus Comidas", donde los usuarios pueden registrar sus desayunos, comidas y cenas, calculando así su progreso nutricional diario, semanal y mensual. Además, para mejorar el conocimiento de los usuarios se dispone de una sección blog, donde habrá artículos que ayuden a potenciar el conocimiento general y donde los usuarios puedan interactuar, un buscador de recetas que pretende dejar atrás el uso de comida precocinada y facilitar el acceso a recetas de calidad y saludables. Por último, se podrá consultar los productos en temporada y fuera de temporada nacionales.

La herramienta creada en este TFG, está pensada para cualquier persona que busque tener un control de su alimentación, sin importar el tipo, desde una dieta omnívora, pescetariana, mediterránea, etc. hasta vegana. Asimismo, esta página web les facilita llegar a nutrientes específicos, permitiendo la obtención de una educación alimentaria y el uso de las herramientas disponibles.

## 1.2. Razones de la elección del TFG

A la hora de escoger un TFG u otro, era indispensable para mí que fuese útil y que pudiese ser una herramienta atractiva y de ayuda para la gente. Buscado una oportunidad

para desarrollar algo propio, donde tuviese que enfrentarme a problemas y desarrollarme a mí mismo a la par que el proyecto.

Por otro lado, lo que también me llamó la atención a la hora de escoger este proyecto, fue el desarrollo web. A la hora de la elección, estaba muy poco familiarizado en este área y quería tener un proyecto del estilo sobre todo para una vez acabada la carrera, tener más claro qué áreas son las que más me llaman la atención. Ya que de forma parecida, en mis prácticas he estado trabajando con el análisis de datos y he tratado por mi cuenta de practicar un poco en el área de la ciberseguridad.

## 2. Enfoque Inicial

En esta sección, se definirá el planteamiento inicial del proyecto, estableciendo sus objetivos y su conexión con los Objetivos de Desarrollo Sostenible (ODS). Además, se especificará la planificación necesaria para alcanzar dichos objetivos de manera eficiente y efectiva.

### 2.1. Conexión con los Objetivos de Desarrollo Sostenible y los principios y Valores Democráticos

Esta aplicación, busca de algún modo alinearse con los objetivos de desarrollo sostenible y los principios y valores democráticos. A continuación, detallamos cómo nuestra plataforma contribuye a estos objetivos [10]:

#### 2.1.1. Desarrollo Sostenible

En primer lugar, se comentarán los Objetivos de Desarrollo Sostenible con los que se alinea el proyecto.

##### 2.1.1.1. ODS 2: Hambre Cero

El ODS 2 busca eliminar el hambre en el mundo y garantizar el acceso a alimentos seguros y nutritivos para todos.

Mediante una dieta basada en vegetales como la que propone el veganismo, se puede contribuir a este objetivo. Principalmente porque utiliza menos recursos y produce menos emisiones de gases de efecto invernadero en la producción de carne animal. De igual modo, al evitar el consumo de productos de origen animal, se liberan recursos como tierra y agua que se utilizan de manera más eficiente para cultivar alimentos vegetales, lo que podría ayudar a reducir la hambruna en áreas con escasez de recursos.

##### 2.1.1.2. ODS 3: Salud y Bienestar

Al promover una alimentación basada en plantas y realizar un seguimiento para llegar a los nutrientes objetivos según tus condiciones nuestra aplicación apoya la salud y el bienestar. Una dieta bien planificada puede prevenir diversas enfermedades crónicas, como las cardíacas, diabetes o ciertos tipos de cáncer, contribuyendo a una vida más saludable y longeva.

### **2.1.1.3. ODS 4: Educación de Calidad**

De igual modo, mediante la sección de blog donde se ofrecen artículos que tratan de impulsar y mejorar el conocimiento general, el usuario puede recibir educación acerca de la alimentación y sobre cómo lograr los objetivos nutricionales.

### **2.1.1.4. ODS 12: Producción y Consumo Responsables**

Por otro lado, nuestra aplicación destaca recetas y productos de temporada nacionales. Fomentando el consumo de alimentos locales y reduciendo la huella de carbono asociada al transporte de alimentos. De esta forma se apoya tanto a los agricultores locales como se promueven prácticas de consumo y producción sostenibles.

### **2.1.1.5. ODS 13: Acción por el Clima**

Al incentivar las dietas basadas en plantas, contribuimos a la reducción de las emisiones de gases de efecto invernadero, ya que la producción de alimentos de origen vegetal generalmente tiene un menor impacto ambiental con la producción de carne.

### **2.1.1.6. ODS 15: Vida de Ecosistemas Terrestres**

La agricultura animal también es una de las principales causas de deforestación y degradación del suelo. Evitar el consumo de carne y productos lácteos contribuye a la conservación de los ecosistemas terrestres y la biodiversidad reduciendo la presión sobre los recursos naturales y los hábitats.

### **2.1.1.7. ODS 16: Paz, Justicia e Instituciones Sólidas**

Si bien la relación entre el veganismo y el ODS 16 no es directa, el veganismo promueve la justicia en el sentido de que busca reducir el sufrimiento animal y abogar por una alimentación más ética. Además, puede fomentar la conciencia sobre la importancia de la toma de decisiones éticas en nuestra vida cotidiana.

## **2.1.2. Principio y Valores Democráticos**

Por otro lado, nuestra plataforma está diseñada para que tenga un enfoque inclusivo, donde independientemente de los objetivos o sus hábitos alimenticios, sea accesible para todos los usuarios. Se promueve la educación y el acceso a la información a través de un blog con noticias y artículos educativos.

## **2.2. Objetivos**

El objetivo principal de este trabajo es el desarrollo de una aplicación web que permita llevar a cabo el seguimiento nutricional de cualquier persona pero enfocado en particular a los veganos/vegetarianos ya que tienen mayor dificultad para alcanzar todos sus objetivos nutricionales como proteínas, minerales y vitaminas ya que descartamos

los productos de origen animal. Se busca que la gente aprenda y obtenga conocimiento para aplicar en su día a día. Además, se busca que se puedan consultar los productos de temporada para que de esta forma fomentar el comercio y consumo de artículos locales. Para ello, se quiere ofrecer un diseño intuitivo, limpio y user friendly.

Para cumplir los requisitos del proyecto será necesario cumplir los siguientes objetivos:

- Desarrollar una aplicación WEB que:
  - Permita el seguimiento exhaustivo de la alimentación nutricional, ofreciendo recetas que utilicen artículos de temporada y con gráficas para ver el progreso
  - Que permita mejorar el conocimiento promedio del usuario.

La aplicación deberá contar con características como:

- **Home:** Se podrá ver un resumen de la página, quienes somos y motivación. Además, contiene los botones para iniciar sesión y registrarse.
- **Inicio/Registro:** Para que cada usuario pueda tener su información y no se mezcle con la de otros usuarios hace falta que haya Auth.
- **Inicio:** Una vez registrado se podrán ver los progresos agrupados por tipo de nutriente: Mineral, Grasas, Vitaminas... Además, se podrá ver el progreso diario, semanal o mensual o consultar incluso días anteriores.
- **Mis Comidas:** Una sección donde se pueden ver todas las comidas introducidas hasta la fecha. Además, para cada comida se puede ver los Alimentos que la conforman y añadir o borrar Comidas y Alimentos.
- **Recetas:** En esta sección se pueden consultar recetas en caso de introducir un ingrediente o receta, pudiendo mirar los ingredientes necesarios, los tipos de alérgenos y un link donde te lleva a la receta original con un vídeo.
- **Info Alimentos:** Para poder consultar la información nutricional de determinados alimentos se puede introducir la cantidad de algo que hayas ingerido para que te salga todos los nutrientes que esa cantidad contiene y así poder añadirlo a tu progreso.
- **Artículos de Temporada:** Interfaz donde se permiten ver los productos que están de temporada, fuera de temporada y que están o entrando o apunto de salir. Se mostrará la información mes a mes y de productos cuya cosecha sea española, con una descripción o consejo para cada producto.
- **Blog:** Para poder dar duración alimenticia a los usuarios, se dispone de un blog donde hay varios artículos relacionados con recetas, vida sana, noticias sobre dietética y nutrición donde los usuarios podrán interactuar entre ellos.

Durante el desarrollo de la aplicación, se prestará especial atención a la seguridad de los datos. También se buscará una interfaz intuitiva y fácil de usar, considerando la diversidad de perfiles y niveles de experiencia de los usuarios. Tratando además, de hacer la web escalable para poder implementar nuevos módulos y funcionalidades en el futuro.

## 2.3. Herramientas

En esta sección se especificarán los programas usados en el desarrollo de este trabajo. Es una parte ciertamente importante ya que a nivel empresarial el uso de alguna licencia puede suponer un coste bastante grande, lo que se debería tener en cuenta. No obstante, en este caso todo lo que se ha utilizado ha sido en su versión gratuita o la versión premium gracias a las ventajas de ser estudiante. En particular, mediante el estatus de estudiante se han conseguido ventajas en las herramientas DigitalOcean y Name.com.

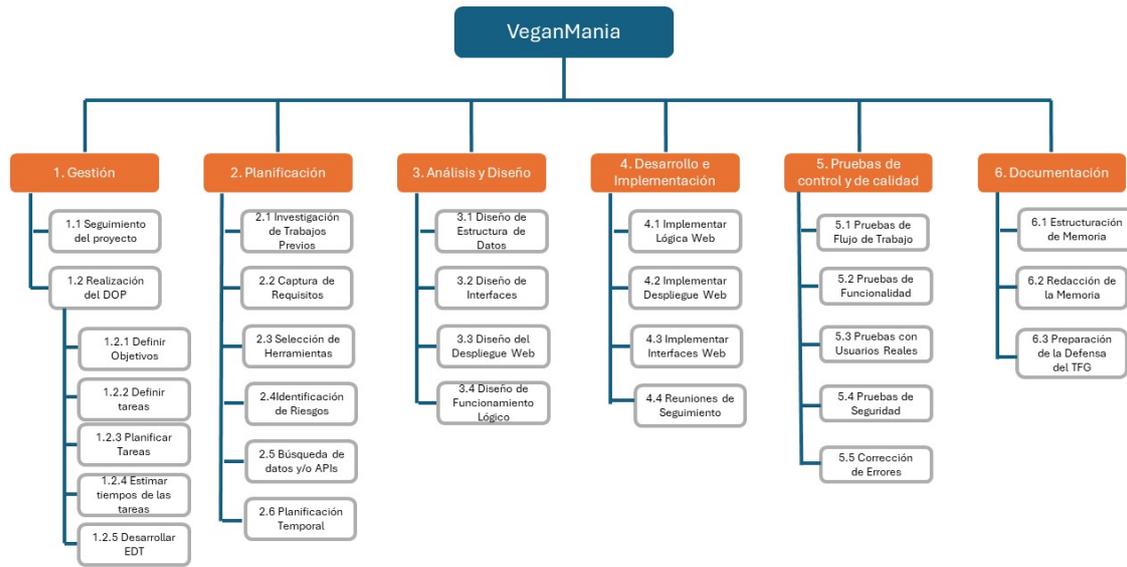
- **OWASP ZAP:** OWASP ZAP es un escáner de seguridad web de código abierto. Se ha utilizado como una aplicación de seguridad y como una herramienta para pruebas de penetración.
- **Visual Studio Code:** Entorno de desarrollo integrado (IDE) utilizado para escribir y editar código. Proporciona diversas funcionalidades y extensiones que facilitan la programación. En particular, las extensiones indispensables para este proyecto han sido: Live Server, Docker, Git y Postman.
- **Python:** El lenguaje de programación elegido para desarrollar todo el proyecto. Todas las librerías utilizadas en Python son gratuitas y se especifican en la sección de Módulos.
- **Git/Github:** Herramienta y repositorio utilizado para llevar un control de versiones de tal manera que todo el progreso se guarde pudiendo acceder a versiones anteriores en el caso de haber algún problema. Además, como se ha mencionado, es gracias a Github Student el tener herramientas premium. Por otro lado, el link al repositorio de <https://github.com/kepaxabier/flask-mysql-dockercompose>[Github].
- **DigitalOcean:** DigitalOcean es un proveedor en la nube confiable y económico. Donde puedes tener varios droplets con máquinas virtuales que en este caso se han utilizado para desplegar la aplicación de manera continua. En este caso, se han conseguido 200€ de crédito para obtener una máquina virtual con una IP en la que desplegar de forma continua la aplicación.
- **Docker Desktop:** Docker Desktop incluye herramientas para desarrolladores, Docker App, Kubernetes y sincronización de versiones. Permite crear imágenes y plantillas eligiendo idiomas y herramientas. Las principales ventajas son la velocidad, la seguridad y la flexibilidad. Se ha utilizado para hacer la aplicación usando Docker de manera que se pueda desplegar la web en cualquier sitio con una línea de código. Por otro lado, se han subido a DockerHub las imágenes creadas por si cualquier persona quisiera utilizarlas sin crear un Dockerfile, el link es el siguiente <https://hub.docker.com/repository/docker/davidelorza/veganmania/general>[DockerHub].
- **Name:** Name es una empresa de alojamiento web y registrador de nombres de dominio. Se ha utilizado para registrar un dominio gracias al paquete estudiante de Github. Esto sirve para que en vez de acceder a la web mediante la IP de la máquina virtual de DigitalOcean, (p.ej: 195.122.122.1) se pueda acceder por un nombre que se pueda recordar y accesible, [veganmania.live](http://veganmania.live).
- **LucidChart:** LucidChart es un software de tipo SAAS para la elaboración de diagramas. Se ha usado para realizar varios diagramas de la documentación. Permite diseñar, visualizar y compartir los diagramas de manera efectiva. Con muchas plantillas para ayudar en el proceso.

- **Diagrams.net:** LucidChart es un software de tipo SAAS para la elaboración de diagramas. Junto a LucidChart se ha usado para bastantes de los diagramas de la documentación. Permite diseñar, visualizar y compartir los diagramas de manera eficiente.
- **Outlook:** Servicio de correo electrónico que ha sido fundamental para la comunicación con los tutores del TFG.
- **Teams:** Servicio de Microsoft para crear un equipo y canales para reunir a personas y trabajar en espacios enfocados con conversaciones y archivos. Además, incluye un calendario para que se conecte con las personas antes, durante y después de una reunión para que sea fácil encontrar la preparación y el seguimiento del TFG.
- **Gmail:** Servicio de correo electrónico utilizado para crear una cuenta para la aplicación. De manera que los mensajes de bienvenida y recuperar contraseña se envían mediante esta cuenta.
- **Overleaf:** Editor de LaTeX en línea utilizado para redactar el TFG. Proporciona una interfaz amigable y funcionalidades específicas para la escritura académica en LaTeX.
- **Windows 11:** Sistema operativo de Microsoft. Esta versión del sistema operativo de Microsoft se caracteriza por su interfaz amigable y sus funciones avanzadas. Se optó por Windows 11 ya que incluye todas las herramientas para la creación de la aplicación web y para la elaboración del TFG.
- **Ubuntu 22.04 LTS:** Sistema operativo de código abierto. Ubuntu es una distribución de Linux ampliamente utilizada y apreciada por su estabilidad, seguridad y soporte de largo plazo. Se decidió utilizar Ubuntu ya que había que comprobar que la aplicación fuese compatible para otros sistemas de manera que cualquier usuario pudiera hacer uso de ella.

## 2.4. Alcance

Para facilitar la visualización y la organización del proyecto se ha desglosado el proyecto en tareas y se han organizado de forma jerárquica, creando así el EDT (Estructura de Descomposición del Trabajo).

El proyecto se ha dividido en 6 paquetes de trabajo con sus consiguientes tareas. Esto se puede ver en la figura 3.



**Figura 3:** EDT

### 2.4.1. Gestión

Este paquete engloba las tareas más administrativas y de definición del proyecto.

<b>Seguimiento del Proyecto</b>	
Descripción	Se ha supervisado y monitorizado el avance del proyecto en relación con las actividades planificadas.
Horas estimadas	10 horas
Horas reales	10 horas

**Tabla 1:** Seguimiento del Proyecto

<b>Definir Objetivos (Realización del DOP (Diagramas de Operaciones del Proceso))</b>	
Descripción	Establecer objetivos realistas y significativos para el proyecto, y realizar revisiones periódicas para garantizar una finalización satisfactoria del mismo.
Horas estimadas	7 horas
Horas reales	7 horas

**Tabla 2:** Definir Objetivos (Realización del DOP)

<b>Definir Tareas (Realización del DOP ((Diagramas de Operaciones del Proceso))</b>	
Descripción	Se definirán las tareas a realizar y se hará una estimación del tiempo necesario para completarlas, así cómo revisar estas tareas a lo largo del proyecto para asegurar el correcto desarrollo del mismo
Horas estimadas	5 horas
Horas reales	7 horas

**Tabla 3:** Definir Tareas (Realización del DOP)

<b>Planificar Tareas (Realización del DOP)</b>	
Descripción	Estimar el orden y la importancia de las tareas para tener el proyecto más organizado
Horas estimadas	1 hora
Horas reales	1 hora

**Tabla 4:** Planificar Tareas (Realización del DOP)

<b>Estimar Tiempos de las Tareas (Realización del DOP)</b>	
Descripción	Estimar el tiempo necesario para cada tarea.
Horas estimadas	1 hora
Horas reales	1 hora

**Tabla 5:** Estimar Tiempos de las Tareas (Realización del DOP)

<b>Desarrollar EDT (Realización del DOP)</b>	
Descripción	Realizar el EDT.
Horas estimadas	1 hora
Horas reales	2 horas

**Tabla 6:** Desarrollar EDT (Realización del DOP)

## 2.4.2. Planificación

En la planificación, en cambio, se comienza a preparar el proyecto y en el desarrollo de esta fase se consiguió acotar mejor los objetivos de este. También se ha investigado qué otros proyectos se han realizado en el ámbito y cuál es su enfoque. Además, se han elegido herramientas acordes con los objetivos y los datos con los que vamos a trabajar.

<b>Investigación trabajos previos</b>	
Descripción	Se llevó a cabo una investigación exhaustiva sobre los trabajos relacionados con una web de nutrición.
Horas estimadas	5 horas
Horas reales	5 horas

**Tabla 7:** Investigación trabajos previos

<b>Captura de requisitos</b>	
Descripción	Establecer los requerimientos tanto funcionales como no funcionales para la aplicación. Adaptar estos requerimientos a lo largo del proyecto para asegurar un desarrollo correcto.
Horas estimadas	10 horas
Horas reales	10 horas

**Tabla 8:** Captura de requisitos

<b>Selección de herramientas</b>	
Descripción	Evaluar y elegir las herramientas más adecuadas para el desarrollo integral del proyecto. Estas herramientas deberán cubrir el diseño y desarrollo de una página web completa. Donde habrá que poder recoger los datos de alimentos y recetas para mostrarlos e interactuar con ellos.
Horas estimadas	10 horas
Horas reales	3 horas

**Tabla 9:** Selección de herramientas

<b>Identificación de riesgos</b>	
Descripción	Identificar los riesgos que pueden afectar de forma directa o indirecta al desarrollo correcto del proyecto.
Horas estimadas	13 horas
Horas reales	15 horas

**Tabla 10:** Identificación de riesgos

<b>Búsqueda de Datos y/o APIs</b>	
Descripción	En esta parte se pretende definir de donde se van a obtener los datos para las recetas, la información nutricional de cada alimento y las necesidades para cada persona en función de su actividad, altura, edad y género.
Horas estimadas	3 horas
Horas reales	5 horas

**Tabla 11:** Búsqueda de Datos y/o APIs

<b>Planificación temporal</b>	
Descripción	En esta parte se pretende establecer y planificar las tareas y etapas del proyecto.
Horas estimadas	8 horas
Horas reales	10 horas

**Tabla 12:** Planificación temporal

### 2.4.3. Diseño

El apartado de diseño es muy importante, ya que en él se especifican aspectos fundamentales como la estructura de datos a utilizar, el diseño de la página web, así como el diseño de como se va a desplegar la web y como se va a comunicar entre los distintos módulos.

<b>Diseño de estructuras de datos</b>	
Descripción	Diseñar una estructura de datos eficiente que garantice un acceso fácil y optimizado a los datos, minimizando el uso de recursos innecesarios. La estructura debe ser capaz de agrupar de manera coherente todos los datos recopilados de múltiples fuentes. Además, se deben crear estructuras de datos secundarias que contengan únicamente la información necesaria para funciones específicas con el fin de maximizar la eficiencia y minimizar la redundancia de datos.
Horas estimadas	2 horas
Horas reales	6 horas

**Tabla 13:** Diseño de estructuras de datos

<b>Diseño de interfaces</b>	
Descripción	Diseñar una maqueta de una página web que facilite su posterior programación. La página web debe tener una interfaz de usuario intuitiva, accesible para usuarios de todas las edades, incluyendo aquellos menos familiarizados con la tecnología. Además, se deben incorporar secciones técnicas para atender las necesidades de usuarios con conocimientos más avanzados. Es importante considerar una estética sencilla que no sobrecargue la página con demasiada información a la vez, con el objetivo de hacerla más atractiva y fácil de navegar. Por último, cabe destacar que el diseño de la página web puede evolucionar a lo largo del proyecto, adaptándose a los requisitos y objetivos cambiantes.
Horas estimadas	10 horas
Horas reales	10 horas

**Tabla 14:** Diseño de interfaces

<b>Diseño de Despliegue Web</b>	
Descripción	Para poder asegurar el funcionamiento de la web en un servidor al que cualquier persona pueda acceder o que simplemente con un archivo y un par de comandos cualquiera pueda hacer uso del programa se debe diseñar cómo se debe desplegar la web.
Horas estimadas	4 horas
Horas reales	5 horas

**Tabla 15:** Diseño de Despliegue Web

<b>Diseño de Lógica Web</b>	
Descripción	Se especifica como se conecta cada herramienta entre sí para un correcto funcionamiento. Además, se plantea el funcionamiento lógico para saber cómo acceder a cada sección de la web, quien puede hacerlo y las opciones que se tienen en cada sección de manera que se asegure un funcionamiento correcto y eficiente.
Horas estimadas	5 horas
Horas reales	10 horas

**Tabla 16:** Diseño de Lógica Web

#### 2.4.4. Desarrollo e Implementación

Durante la etapa de implementación, se lleva a cabo la puesta en práctica de las decisiones tomadas en el análisis y diseño previo. En esta fase del proyecto, se realizan las acciones necesarias para materializar los objetivos establecidos. Se emplean las herramientas y tecnologías elegidas y se pone en uso los datos obtenidos. Además, se aprovecha la investigación realizada sobre proyectos previos en el mismo ámbito para obtener ideas y enfoques relevantes. Mediante la implementación, se busca llevar a cabo de manera efectiva y eficiente el desarrollo del proyecto en línea con los objetivos definidos.

<b>Implementación de la lógica web</b>	
Descripción	Se implementará una lógica acorde con el funcionamiento requerido, donde la comunicación entre servicios sea eficaz y eficiente. Además, se debe poder acceder y manipular todos los datos que necesitemos sobre las recetas los blogs y la información nutricional de los alimentos junto a los avances de cada persona.
Horas estimadas	50 horas
Horas reales	55 horas

**Tabla 17:** Implementación de la lógica web

<b>Implementación del Despliegue web</b>	
Descripción	Para la fácil distribución de la app y que cualquiera con dos simples pasos pueda tener la aplicación en su ordenador hay que implementar un sistema de despliegue, que también es necesario para poder subir la aplicación a Internet donde cualquier persona pueda hacer uso de ella.
Horas estimadas	30 horas
Horas reales	40 horas

**Tabla 18:** Implementación del Despliegue web

<b>Implementación de las Interfaces Web</b>	
Descripción	Implementar las Interfaces Web de forma que se optimice el UX/UI (User-Experience/User-Interface), siendo interfaces amigables, intuitivas y que puedan ser usadas por cualquier tipo de usuario. Además, se deben mostrar las cosas de forma ordenada y bonita para que la información sea útil.
Horas estimadas	50 horas
Horas reales	60 horas

**Tabla 19:** Implementación de las Interfaces Web

<b>Reuniones de seguimiento</b>	
Descripción	Realizar reuniones de seguimiento para evaluar lo realizado con la planificación establecida.
Horas estimadas	20 horas
Horas reales	30 horas

**Tabla 20:** Reuniones de Seguimiento

#### 2.4.5. Plan de pruebas

Durante esta etapa, se llevará a cabo una revisión exhaustiva del código implementado y se verificará el funcionamiento adecuado de la aplicación. Se asegurará que todas las interfaces muestren lo que deben mostrar, sin que salgan errores por el camino independientemente de lo que el usuario haga. El plan de pruebas desempeñará un papel crucial en garantizar la calidad y fiabilidad del software desarrollado, permitiendo identificar posibles errores o fallos y asegurando un rendimiento óptimo de la herramienta web.

<b>Pruebas de Flujo de Trabajo</b>	
Descripción	En esta parte se ha creado un flujo de trabajo para que todos los cambios realizados y subidos a Github sean revisados para que si alguno contiene algún error no admita la modificación.
Horas estimadas	2 horas
Horas reales	2 horas

**Tabla 21:** Pruebas de Flujo de Trabajo

<b>Pruebas de Funcionalidad</b>	
Descripción	Se ha diseñado un plan de pruebas para comprobar todas las zonas críticas y en las que puede haber algún problema según lo que haga el usuario. De esta forma se pulirán errores y hallarán casos que no se han tenido en cuenta
Horas estimadas	15 horas
Horas reales	20 horas

**Tabla 22:** Pruebas de Funcionalidad

<b>Pruebas de Usuario</b>	
Descripción	Se le ha dado acceso anticipado a un conocido para que actúe como primer usuario y haga uso de la aplicación durante una semana para poder buscar más casos que puedan dar pie a un error del sistema.
Horas estimadas	5 horas
Horas reales	5 horas

**Tabla 23:** Pruebas de Usuario

<b>Pruebas de Seguridad</b>	
Descripción	Se ha realizado un plan básico para hacer la aplicación más segura de posibles ciberataques.
Horas estimadas	10 horas
Horas reales	15 horas

**Tabla 24:** Pruebas de Seguridad

<b>Corrección de Errores</b>	
Descripción	Todos los errores o áreas de mejora encontrados se han corregido y mejorado.
Horas estimadas	10 horas
Horas reales	25 horas

**Tabla 25:** Corrección de Errores

#### **2.4.6. Documentación**

La etapa de documentación es fundamental para presentar y comunicar el trabajo realizado a terceras personas. Esta fase garantiza que el conocimiento generado durante el proyecto quede registrado de manera clara y accesible, facilitando su comprensión y permitiendo que otros puedan replicar o utilizar los hallazgos en futuros trabajos o proyectos relacionados.

<b>Estructuración de la memoria</b>	
Descripción	Describir que estructura va a seguir la memoria y los puntos que se van a tratar
Horas estimadas	6 horas
Horas reales	5 horas

**Tabla 26:** Estructuración de la memoria

<b>Redacción de la memoria</b>	
Descripción	Durante la tarea de redacción de la memoria, se hará un registro del progreso del proyecto. Se expondrán todos los descubrimientos realizados en el campo de la detección de interacciones. También se mostrarán las decisiones de diseño y de implementación que hemos tomado. Por último, se expondrán los resultados obtenidos y se compararán con otros trabajos similares.
Horas estimadas	60 horas
Horas reales	50 horas

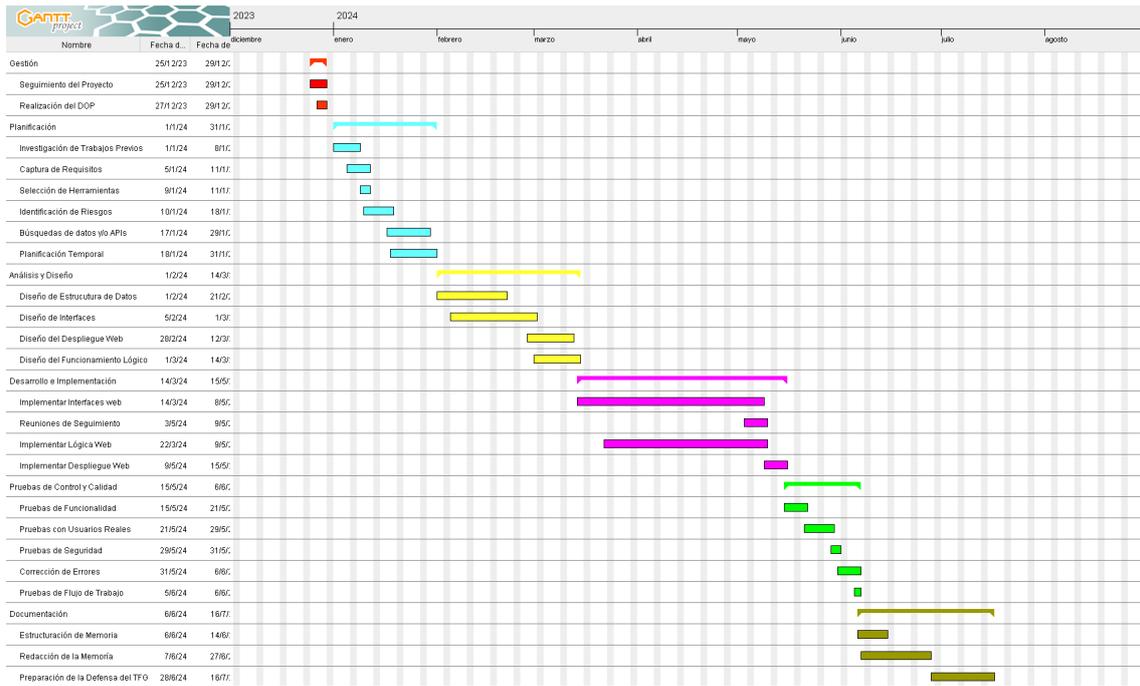
**Tabla 27:** Redacción de la memoria

<b>Elaboración de la defensa</b>	
Descripción	Crear una presentación clara, concisa y atractiva que comunique de manera efectiva los objetivos, metodología y resultados alcanzados en el trabajo realizado. Además de la presentación, la tarea también incluye la demostración de cómo funciona la web. La demostración tiene como objetivo mostrar cómo la herramienta funciona en tiempo real y cómo los usuarios pueden interactuar con ella para obtener resultados.
Horas estimadas	20 horas
Horas reales	20 horas

**Tabla 28:** Creación de la defensa

## 2.5. Planificación Temporal

Busco un TFG a inicios de diciembre de 2023 y los profesores Iñigo Perona y Kepa Bengoetxea me proponen VeganMania. Por ello, a mediado de diciembre comienza el proyecto y la planificación ha sido la siguiente:



**Figura 4:** Diagrama de Gantt

La siguiente tabla recoge todas las tareas identificadas en el proyecto, las agrupa por etapas recogiendo su duración estimada y calculando el desempeño para cada etapa, por último se calcula el tiempo total estimado del proyecto.

arge

<b>Etapa</b>	<b>Tarea</b>	<b>Estimación de horas</b>	<b>Total</b>
Gestión	Seguimiento del Proyecto Realización del DOP	10 horas 27 horas	37 horas
Planificación	Investigación de Trabajos Previos Captura de Requisitos Selección de Herramientas Identificación de riesgos Búsqueda de Datos y/o APIs Planificación temporal	5 horas 10 horas 3 horas 15 horas 5 horas 10 horas	48 horas
Análisis y Diseño	Diseño de Estructura de Datos Diseño de interfaces Diseño del Despliegue Web Diseño de Lógica Web	6 horas 10 horas 5 horas 10 horas	31 horas
Desarrollo e Implementación	Implementación de la lógica web Desarrollo del despliegue web Implementar Interfaces Web Reuniones de seguimiento	55 horas 10 horas 60 horas 30 horas	155 horas
Pruebas y control de calidad	Pruebas de Flujo de Trabajo Pruebas de Funcionalidad Corrección de Usuario Pruebas de Seguridad Corrección de Errores	2 horas 20 horas 5 horas 15 horas 25 horas	67 horas
Documentación	Estructuración de la memoria Redacción de la memoria Preparación de la defensa	5 horas 50 horas 20 horas	75 horas
<b>Total</b>			413 horas

**Tabla 29:** Resumen de tareas

### 2.5.1. Desviación Temporal

En particular, la planificación se ha desviado en las pruebas y control de calidad y en el desarrollo e implementación. En el caso de las pruebas y control de calidad, se encontraron algunas áreas en las que surgían problemas con determinadas entradas del usuario. Además, también se pudo comprobar que alguna de las interfaces necesitaba algo más de intuitividad por lo que también hubo que modificar un poco y añadir un tutorial para facilitar el uso.

Por otro lado, en el desarrollo de implementación se tardó un poco más porque a medida que se iba realizando el proyecto surgían nuevas ideas y mejoras que se fueron implementando. Entre ellas la interfaz para consultar los productos de temporada.

## 2.6. Análisis de Riesgos

En cualquier proyecto, es posible que surjan contratiempos que puedan afectar a nuestra planificación. Por ello, hay que identificar y analizar aquellos que puedan afectarnos y buscar planes de prevención y contingencia, así como la probabilidad de que se produzcan y el impacto que podrían tener. Se han establecido 4 categorías de riesgos dependiendo de la probabilidad de suceso:

Categoría	Probabilidad	Descripción
Baja	0 % y 25 %	El riesgo podría generarse en algún momento
Media	26 % y 50 %	El riesgo puede generarse más comúnmente
Alta	51 % y 75 %	El riesgo probablemente termine ocurriendo
Muy alta	76 % y 100 %	El riesgo ocurrirá en la mayoría de ocasiones

**Tabla 30:** Tabla de definición de riesgos

### 2.6.1. Lista de Riesgos

<b>Fallo del equipo</b>	
Descripción	Parte del proyecto se va a realizar con ordenador personal por lo que cualquier fallo de hardware o software puede retrasar el desarrollo del proyecto
Medidas de prevención	Mantener el equipo actualizado y realizar copias de seguridad con frecuencia
Plan de contingencia	Utilizar Github para llevar un control de versiones e incluso poder acceder desde diferentes dispositivos.
Categoría	Baja
Impacto	Dependiendo de la gravedad entre unas horas y un par de días

**Tabla 31:** Riesgo: Análisis de nuevas funcionalidades

<b>Falta de conocimientos</b>	
Descripción	No tener los conocimientos suficientes para completar alguna función de la aplicación
Medidas de prevención	Dedicar tiempo a un periodo de aprendizaje
Plan de contingencia	Utilizar la documentación y buscar en internet problemas similares
Categoría	Media
Impacto	Depende de la gravedad, menos de un día o una semana en los peores casos.

**Tabla 32:** Riesgo: Falta de conocimientos

<b>Algún servicio de terceros deje de dar soporte</b>	
Descripción	Algunos de los servicios utilizados en el proyecto deje de dar soporte o estén obsoletos
Medidas de prevención	Utilizar servicios que sean populares y con soporte.
Plan de contingencia	Buscar alternativas y dependiendo de la gravedad realizar una nueva planificación
Categoría	Baja
Impacto	Depende de la gravedad, al menos un día o una semana en los peores casos.

**Tabla 33:** Riesgo: Servicios de terceros sin soporte

<b>Fallos de seguridad en librerías de terceros</b>	
Descripción	Exista una vulnerabilidad en la aplicación debido a servicios de terceros
Medidas de prevención	Utilizar servicios que sean seguros e informarse de la seguridad que ofrecen.
Plan de contingencia	Buscar alternativas y dependiendo de la gravedad realizar una nueva planificación. Si es una brecha que no tiene solución cambiar de servicio y replantear la aplicación
Categoría	Baja
Impacto	Depende de la gravedad, al menos una semana o un mes en los peores casos.

**Tabla 34:** Riesgo: Fallos de seguridad en librerías de terceros

<b>Enfermedad</b>	
Descripción	Imprevistos que puedan surgir con la salud
Medidas de prevención	Ser consciente del estado físico de uno mismo
Plan de contingencia	Descansar y seguir las recomendaciones sanitarias
Categoría	Media
Impacto	Si la causa es una lesión, se podría considerar como un retraso de pocos días. Sin embargo, si se trata de una enfermedad, el retraso podría extenderse durante varios días o semanas

**Tabla 35:** Riesgo: Enfermedad

<b>No cumplir la planificación</b>	
Descripción	Excederse en los tiempos establecidos en el desarrollo de la planificación
Medidas de prevención	Tener en cuenta un margen de tiempo amplio en la planificación.
Plan de contingencia	Volver a planificar el trabajo y aplazar levemente algunas tareas
Categoría	Media
Impacto	muy grave

**Tabla 36:** Riesgo: No cumplir la planificación

<b>Cambio de requisitos o planteamiento del proyecto</b>	
Descripción	Los requisitos del proyecto cambian, y a su vez, se debe hacer un replanteamiento
Medidas de prevención	Tener buena comunicación con el equipo y realizar presentaciones que representen correctamente las funciones de la aplicación
Plan de contingencia	Replanteamiento del proyecto
Categoría	Probable
Impacto	Grave

**Tabla 37:** Riesgo: Cambio de requisitos o planteamiento del proyecto

## 2.7. Descripción del presupuesto

En esta sección se realiza una estimación de lo que ha costado el proyecto. Se ha intentado ser lo más exacto posible y se han calculado los costes en función de las horas empleadas. Gracias a estos cálculos podremos hacernos una idea aproximada de lo que costaría desarrollar un proyecto similar.

### 2.7.1. Costes de personal

En general, calcular los costes de personal es un tema bastante importante a tratar, ya que representan el gasto relacionado con los recursos humanos involucrados en el proyecto. Además, pueden suponer un gran coste. En este caso, ha sido principalmente el trabajo de un estudiante de Ingeniería Informática, que ha dedicado entorno a 360 horas, sin embargo, también ha habido tiempo por parte de los tutores que directamente pueden haber dedicado 20 horas cada uno.

Para calcular el coste de personal, vamos a usar el sueldo promedio de un programador Junior en España (10,77 €/h). Por otra parte, el salario medio en España para un docente de universidad es de 20€/h.

Para calcular el coste de personal usaremos la siguiente fórmula:

$$CosteTotal = \sum_{i=1}^n (Horas_i \cdot Sueldo/hora_i)$$

Donde:

- $Horas_i$  representa las horas invertidas en el trabajo  $i$ .
- $Sueldo/hora_i$  es el sueldo medio por hora del trabajo  $i$ .

En nuestro caso, tenemos el alumno y los dos tutores. Por lo que el resultado sería el siguiente:

$$\text{Coste total de personal} = 360h. \cdot 10,77\text{€/hora} + 2 \cdot (20h. \cdot 20\text{€/hora}) = 4667,2\text{€}$$

Como se puede observar es un coste a tener en cuenta en proyectos y puede suponer el mayor coste.

### 2.7.2. Costes de Hardware

En este ámbito tecnológico los costes de hardware son un aspecto a tener en cuenta ya que en general son la base para cualquier área tecnológica, como en este caso la programación. En este coste, se incluye la inversión económica inicial para disponer de los equipos (ordenadores, servidores, dispositivos de almacenamiento, etc) y el coste para mantenerlos. En este caso se ha hecho uso de los siguientes dispositivos.

**Ordenador portátil (Dell Inspiron 14500):** Se ha utilizado un ordenador portátil en todo momento utilizado para avanzar con el proyecto tanto en casa como fuera de casa. En este caso, en algún momento ha faltado un poco de potencia para poder ser multitarea.

- Procesador: Intel® Core i7-1165G7 de 11.ª generación
- Tarjeta Gráfica: Intel® Iris Xe
- RAM: 16 GB, 2 x 8 GB, DDR4, 3200 MHz de doble canal
- Memoria: SSD de 512 GB, M.2 2280, PCIe NVMe de 3.ª gen. x4

Total: 963 €

**Monitores:** Para poder aumentar la eficiencia de trabajo, haciendo mejor provecho del tiempo y trabajar de forma más grata se ha hecho uso de dos monitores que junto a un Dock se han conectado sendos monitores al portátil.

- Monitor 1: Lenovo ThinkVision S22E 21.5"LED FullHD 75Hz FreeSync
- Monitor 2: Lenovo ThinkVision S22E 21.5"LED FullHD 75Hz FreeSync
- Dock: Dell UD22 Estación de Acoplamiento Universal Thunderbolt

Total: 392 €

**Droplet DigitalOcean:** Para poder disponer de la web en cualquier momento se ha utilizado un droplet de DigitalOcean. Es decir, una máquina virtual que muestra en todo momento el contenido de la aplicación desplegada. Hay que mencionar que se disponía de crédito por el ya mencionado paquete de estudiantes de Github Student, pero el coste de ese uso ha sido el siguiente:

- Máquina Virtual: Ubuntu 22.04 LTS Regular, 1 GB CPU & 25 GB SSD disk VM:

Total: 6,12 €

Como se puede observar los costes de Hardware ascienden a 1361,12 €. En este caso ha sido solo el coste para una persona pero para una empresa grande donde hay cientos de trabajadores este coste asciende considerablemente por lo que es otro gran aspecto a tener en cuenta sobre todo en un área tecnológica como la nuestra.

### 2.7.3. Costes de software

El único software que se optó de pago fue Windows 11 para redactar el Documento del TFG debido a su familiaridad y comodidad. El coste de la licencia de Windows Home es de 119,99€, lo que proporciona acceso completo a todas las características y actualizaciones del sistema operativo.

Por otro lado, para tener registrado el dominio con un nombre que sea de nuestro gusto y de esta forma evitar que nuestra IP sea vulnerable se ha usado un dominio de name.com que se ha asociado al droplet de DigitalOcean. Debido a que por el paquete de estudiante sale gratis durante dos años no se tiene en cuenta, pero si se quisiera mantener en el futuro habría que pagar anualmente 7 euros.

**Coste Total Software:** 119,99 € = 119,99 €

### 2.7.4. Costes Indirectos

En esta sección, se van a describir los costes indirectos, que son aquellos que afectan al proceso productivo en general de uno o más productos, por lo que no se puede asignar directamente a un solo producto sin usar algún criterio de asignación. Por ejemplo, los gastos relacionados con el suministro de energía eléctrica o de conexión a Internet. En muchos casos hacer un buen cálculo de estos costes es diferencial porque a pesar de que sean difíciles de estimar son esenciales para garantizar un presupuesto completo y preciso que refleje todas las necesidades financieras del proyecto en su totalidad.

#### 2.7.4.1. Electricidad:

Para calcular el coste del consumo energético usaremos la siguiente fórmula:

$$CosteTotal = \sum_{i=1}^n (KWh_i \cdot horas_i) \cdot precioKWh$$

Donde:

- $KWh_i$  representa la cantidad de energía eléctrica consumida en la actividad  $i$  por hora.
- $precioKWh$  es el precio del kWh medio del año 2024 de Iberdrola con una potencia de 15KW. En este caso: 0.1743 [12]€/KWh
- $horas_i$  es el número de horas invertidas en la actividad  $i$ .

Por una parte, el consumo de energía de un portátil suele oscilar entre 50-100W/hora por lo que estableceremos 75W como el consumo de energía medio por hora. El portátil ha sido utilizado a lo largo de todo el proyecto, en torno a 360h.

Por otra parte, los monitores suelen estar en torno a los 25W hora, dado que hemos usado dos monitores, sumamos 50 W. Sumados a 12W de consumo medio de un Dock hacen 62 W. No obstante, hay que tener en cuenta que los monitores no se han utilizado en todo momento, ya que ha habido momentos que se ha avanzado tan solo con el portátil. Estimo que el uso de este conjunto ha sido de 300 horas de las 360 totales.

Además, hay que incluir los costes de iluminación y el consumo del router. En el caso de la iluminación, se ha utilizado una lámpara LED cuando no se disponía de luz solar, cuyo consumo se estima en 10 W/h. Por otro lado, el consumo medio por hora de un router suelen ser de 12 W. Vamos a considerar que la lámpara la hemos utilizado el 20 % de las horas, es decir 72, y que el router lo hemos utilizado todo el tiempo.

Por ello, teniendo en cuenta todos estos factores y teniendo en cuenta que esos W los tenemos que pasar a Kw, aplicamos la fórmula anteriormente mencionada para calcular el coste total de energía del proyecto.

$$EnergíaTotaldelPorttil(Wh) = 75 W \times 360 h = 27,000 Wh$$

$$EnergíaTotaldeMonitoresyDock(Wh) = 62 W \times 300 h = 18,600 Wh$$

$$EnergíaTotaldelaLmparaLED(Wh) = 10 W \times 72 h = 720 Wh$$

$$EnergíaTotaldelRouter(Wh) = 12 W \times 360 h = 4,320 Wh$$

$$EnergíaTotaldelPorttil(kWh) = \frac{27,000 Wh}{1000} = 27,0 kWh$$

$$EnergíaTotaldeMonitoresyDock(kWh) = \frac{18,600 Wh}{1000} = 18,6 kWh$$

$$EnergíaTotaldelaLmparaLED(kWh) = \frac{720 Wh}{1000} = 0,72 kWh$$

$$EnergíaTotaldelRouter(kWh) = \frac{4,320 Wh}{1000} = 4,32 kWh$$

$$CosteTotaldelPorttil = 27,0 kWh \times 0,1743 €/kWh = 4,7061 €$$

$$CosteTotaldeMonitoresyDock = 18,6 kWh \times 0,1743 €/kWh = 3,24398 €$$

$$CosteTotaldelaLmparaLED = 0,72 kWh \times 0,1743 €/kWh = 0,125496 €$$

$$CosteTotaldelRouter = 4,32 kWh \times 0,1743 €/kWh = 0,752976 €$$

$$CosteTotaldelProyecto = 4,7061 € + 3,24398 € + 0,125496 € + 0,752976 € = 8,828552 €$$

#### 2.7.4.2. Internet

La conexión a internet ha sido fundamental en el desarrollo del proyecto, y ha sido utilizada para diversas tareas como investigación, almacenamiento de datos, y acceso a la nube. Se optó por contratar un plan asimétrico a través de Euskaltel, con una velocidad de descarga de 1 Gbps y una velocidad de subida de 300 Mbps, garantizando así transferencias rápidas de datos. El coste total del plan ascendió a 34,95 €.

**Coste total Indirectos:** 34,95 € + 8,828552 € = 43,77 €

#### 2.7.5. Coste total

El coste total del proyecto se compone de los costes de personal, hardware, software y otros gastos indirectos. A continuación se muestra el desglose de cada uno:

- Costes de personal: 4667,2 €
- Costes de hardware: 1361,12 €
- Costes de software: 119,99 €
- Costes indirectos:
  - Electricidad: 8,83 €
  - Internet: 34,95 €

El coste total del proyecto se obtiene sumando todos los costes mencionados anteriormente:

$$\text{CosteTotaldelProyecto} = \text{CostesdePersonal} + \text{CostesdeHardware} + \text{CostesdeSoftware} + \text{CostesIndirectos}$$

$$\text{CosteTotaldelProyecto} = 4667,2\text{€} + 1361,12\text{€} + 119,99\text{€} + (8,83\text{€} + 34,95\text{€})$$

$$\text{CosteTotaldelProyecto} = 4667,2\text{€} + 1361,12\text{€} + 119,99\text{€} + 8,83\text{€} + 34,95\text{€}$$

$$\text{CosteTotaldelProyecto} = 6192,09\text{€}$$

De este resultado, cabe destacar que estos costes representan una inversión necesaria para la correcta ejecución del proyecto. Además, siempre que ha sido posible se ha buscado minimizar el coste teniendo en cuenta que cada importe desempeña un papel crucial en el logro de los objetivos y la entrega de resultados de calidad.

### **2.7.6. Beneficio de la Aplicación**

Esta aplicación pone de un modo solución a la gran cantidad de servicios con suscripción que existen. Nuestro objetivo es que el software sea libre y accesible, permitiendo a cualquier persona utilizar sus funcionalidades sin restricciones. Además, el usuario tiene la posibilidad de descargar y modificar el código, para hacer uso de manera local y aplicarlo a sus necesidades específicas.

Por otro lado, el mayor beneficio puede ser crear una comunidad. Si logramos atraer la atención de usuarios y desarrolladores, la aplicación puede ser de gran utilidad para un público amplio. De manera que aquellos interesados pueden optimizar, mejorar, corregir o implementar nuevas funciones a través de la colaboración en Github.

## 3. Estado del Arte

Por definición, la alimentación es el conjunto de acciones mediante las cuales se proporcionan alimentos al organismo. Es decir, abarca la selección de los alimentos, el cocinado y su ingestión. La alimentación depende de muchas variantes, desde las necesidades individuales y la disponibilidad de alimentos hasta aspectos psicológicos y modas. Además de ser una necesidad vital para la supervivencia, la alimentación también es bienestar, cultura y placer. Representa un factor clave en la vida de las personas y el hecho de alejarse de una cultura sana puede conducir a diversas enfermedades. En adición, día a día existe más comida precocinada de poca riqueza nutricional que hace que mucha gente pase al lado de los ultra-procesados, dejando de lado las bases para una alimentación saludable. Por ello, tratar de mejorar el conocimiento general y proporcionar herramientas para la ayuda de seguimiento nutricional es un tema que puede ser beneficioso.

A pesar de que existan una infinidad de aplicaciones para el control calórico, se busca dar un paso más intentando ofrecer alternativas y como ya se ha mencionado con objetivo de mejorar el conocimiento.

En esta sección, revisaremos el estado del arte en este sector alimentario, con especial atención a las opciones veganas y vegetarianas. Comenzaremos examinando las distintas bases de datos que se pueden emplear para las recetas, la información nutricional sobre cada alimento y los objetivos nutricionales para cada individuo. Posteriormente, haremos un análisis sobre los mejores métodos para abordar la tarea, dentro de esta parte está incluido dónde y cómo guardar la información para saber cómo acceder con ella fácil y rápidamente y obtener un poco de inspiración de otras webs parecidas para que sea lo más intuitiva posible. Con este análisis, se espera proporcionar una visión completa y actualizada de las estrategias de las mejores webs para ofrecer la mejor experiencia posible al usuario y que sea de gran utilidad.

### 3.1. Proceso de Obtención de Datos

En esta sección se van a comentar las bases de datos posibles identificadas para cumplir todos los objetivos. En específico, se necesita una base de datos o referencias para obtener los objetivos nutricionales de las personas en función de la etapa de la vida en la que se encuentren y su estado. Además, se necesita buscar alguna fuente de datos para saber la información nutricional de cada alimento. Por otro lado, necesitamos buscar una fuente que nos indique los productos de temporada por cada mes en España, para así priorizar el consumo local. Por último, para poder ofrecer recetas personalizadas de distintos tipos, y que se puedan filtrar para solo mostrar las que ofrezcan productos de temporada, necesitamos datos sobre recetas.

### 3.1.1. Datos de Objetivos Nutricionales

En primer lugar, para saber los objetivos nutricionales según la etapa de la vida se han identificado 3 fuentes de información:

- **Novartis Consumer Health:**<sup>1</sup> Novartis Consumer Health es una multinacional farmacéutica suiza, dedicada a productos de salud de consumo. Contiene la información de los objetivos de muchos nutrientes.
- **Autoridad Europea de Seguridad Alimentaria:**<sup>2</sup> La Autoridad Europea de Seguridad Alimentaria es una agencia de la Unión Europea, el cual es responsable de detectar y alertar todos aquellos riesgos y problemas que afecten a la seguridad alimentaria. Contiene de igual modo, la mayoría de los nutrientes que se buscan a ofrecer.
- **Comité Científico de la Agencia Española de Seguridad Alimentaria y Nutrición:**<sup>3</sup> El Comité Científico de la Agencia Española de Seguridad Alimentaria y Nutrición es el órgano encargado de proporcionar el asesoramiento científico necesario para que la Agencia lleve a cabo las actividades que le son encomendadas.

### 3.1.2. Datos sobre Productos de Temporada Nacionales

Por otro lado, en este apartado se mencionarán las fuentes encontradas para identificar que productos de cosecha local están de temporada.

- **Ministerio de Agricultura, Pesca y Alimentación:**<sup>4</sup> Ofrece el calendario para todas las frutas y verduras de temporada nacionales.

### 3.1.3. Datos sobre Alimentos y Recetas

En este apartado se recogen Alimentos y Recetas porque hay algunas bases de datos y APIs que recogen las dos funciones. Se van a identificar y comentar las más destacadas entre las identificadas. Se diferenciará entre obtención mediante API y descarga de Base de Datos.

#### 3.1.3.1. Descarga de Base de Datos

En primer lugar, una opción es descargar la base de datos y disponer de ella en todo momento. Para esta parte, hemos encontrado las siguientes fuentes de donde podríamos obtener la base de datos.

- **Base de Datos de Nutrientes del Departamento de Agricultura de los Estados Unidos (USDA)**<sup>5</sup> Esta base de datos proporciona información detallada sobre

---

<sup>1</sup>[https://seom.org/seomcms/images/stories/recursos/infopublico/publicaciones/soporteNutricional/pdf/anexo\\_05.pdf](https://seom.org/seomcms/images/stories/recursos/infopublico/publicaciones/soporteNutricional/pdf/anexo_05.pdf)

<sup>2</sup><https://www.efsa.europa.eu/es/topics/topic/dietary-reference-values>

<sup>3</sup>[https://www.aesan.gob.es/AECOSAN/docs/documentos/seguridad\\_alimentaria/evaluacion\\_riesgos/informes\\_comite/RECOMENDACIONES\\_DIETETICAS.pdf](https://www.aesan.gob.es/AECOSAN/docs/documentos/seguridad_alimentaria/evaluacion_riesgos/informes_comite/RECOMENDACIONES_DIETETICAS.pdf)

<sup>4</sup><https://www.alimentosdespana.es/es/campanas/historico-de-campana-y-programas/frutas/frutas-hortalizas-de-temporada/default.aspx>

<sup>5</sup><https://fdc.nal.usda.gov/download-datasets.html>

la composición de nutrientes de una amplia variedad de alimentos, incluyendo frutas, verduras, carnes, productos lácteos y más. Ofrecen entorno a 2M de productos, donde podemos obtener alguna marca conocida pero ninguna nacional. Por otra parte, aunque se puede descargar fácil y es accesible, no dan información sobre minerales y vitaminas, tan solo grasa y calorías.

- **Base de Datos de Nutrientes de la Unión Europea (ENDB)<sup>6</sup>**: La base de datos ENDB ofrece información sobre la composición de nutrientes de alimentos consumidos comúnmente en la Unión Europea. No obstante, no es posible acceder a la de varios países de Europa entre ellas la base de datos de España, BEDCA, ya que requiere de un código de autorización. La única base de datos accesible es la de Francia, CIQUAL, con 3000 instancias con toda la información nutricional necesaria.
- **Base de Datos de Nutrientes de Australia (AUSNUT)<sup>7</sup>**: Proporciona información sobre la composición de nutrientes de alimentos comunes en Australia. La base de datos contiene 5471 instancias con toda la información nutricional necesaria. No obstante, no se especifican marcas, por lo que en vez de salir una marca en concreto sale "commercial biscuit".
- **Base de Datos de Alimentos de la Administración de Alimentos y Medicamentos de los Estados Unidos (FDA)<sup>8</sup>**: Contiene información sobre la composición nutricional de alimentos procesados y productos alimenticios de los Estados Unidos. No obstante, la base de datos no es accesible.
- **Cronometer<sup>9</sup>**: Es una aplicación en línea que ofrece información detallada sobre la composición de nutrientes de alimentos y permite realizar un seguimiento de la ingesta dietética personal. Dispone de varias bases de datos pero ninguna española ni tampoco ninguna extensa, ya que la más extensa no está disponible.
- **Open Food Facts<sup>10</sup>**: Una base de datos colaborativa en línea que recopila información sobre la composición de alimentos de todo el mundo. Los datos son contribuidos por voluntarios. El conjunto de datos está disponible en distintos archivos como JSON, CSV, XML o MongoDB. Sin embargo, los datos no contienen información completa, es necesario usar la base de datos descargada y obtener información más detallada con la API.
- **MyFitnessPal<sup>11</sup>**: Aunque es conocida por su seguimiento de calorías y ejercicio, MyFitnessPal también proporciona información sobre la composición de nutrientes de alimentos y permite realizar un seguimiento de la ingesta dietética personal. La base de datos no está disponible.
- **MyFoodData<sup>12</sup>**: Web que sirve para hacer un seguimiento nutricional con artículos sobre la alimentación que se puede ver en el siguiente link link. La base de datos es bastante extensa con 14000 instancias y toda la información nutricional necesaria, no obstante, no hay productos europeos.
- **RecetasGratis<sup>13</sup>**: Más que en la nutrición esta base de datos se enfoca en ofrecer una amplia variedad de recetas con sus ingredientes. Es una base de datos con más

---

<sup>6</sup><https://www.eurofir.org/>

<sup>7</sup><https://www.foodstandards.gov.au/science/monitoringnutrients/ausnut/Pages/default.aspx>

<sup>8</sup><https://www.fda.gov/food/food-database-available-download>

<sup>9</sup><https://cronometer.com/>

<sup>10</sup><https://world.openfoodfacts.org/>

<sup>11</sup><https://www.myfitnesspal.com/es>

<sup>12</sup><https://www.myfooddata.com/>

<sup>13</sup><https://www.recetasgratis.net/>

de 20000 recetas, que contienen campos como categoría, tipo, dificultad, tiempo de elaboración y número de comensales que facilitan el uso y filtrado de las recetas.

### 3.1.3.2. Bases de Datos mediante API

Por otro lado, existen diversas APIs (Interfaces de Programación de Aplicaciones) que proporcionan acceso a información sobre alimentos y nutrientes a través de Internet. Estas APIs permiten a los desarrolladores de aplicaciones y sitios web acceder a datos nutricionales y de composición de alimentos para crear aplicaciones relacionadas con la nutrición, la salud y la dieta. Se han consultado las siguientes:

- **API de USDA Food Composition** <sup>14</sup>: Proporciona acceso a la base de datos de composición de alimentos del Departamento de Agricultura de los Estados Unidos (USDA), que contiene información nutricional detallada sobre una amplia variedad de alimentos. Ofrecen entorno a 2M de productos no europeos y ofrece 1000 solicitudes.
- **API de Edamam Nutrition Analysis** <sup>15</sup>: Ofrece acceso a una base de datos nutricional que incluye información sobre ingredientes y recetas. Permite calcular información nutricional y detalles de etiquetas para alimentos y recetas. Entre los planes, el gratuito permite analizar información básica sobre alimentos, un análisis nutricional más completo y recetas. Tiene un asistente inteligente y permite buscar 400 recetas mensuales con una buena documentación.
- **API de Nutritionix** <sup>16</sup>: Aplicación que proporciona datos nutricionales y de composición de alimentos, así como información sobre restaurantes y etiquetas de productos alimenticios, su link es el siguiente. No hay máximo de solicitudes ya que solo hay plan gratuito pero no existe una gran cantidad de alimentos, ni nutrientes.
- **API de Open Food Facts** <sup>17</sup>: Ofrece acceso a una base de datos colaborativa de información sobre alimentos de todo el mundo. Los datos son contribuidos por la comunidad y cubren una amplia variedad de productos. Por una parte permite 100 solicitudes y hay bastantes opciones para usarlo, y mucha documentación, el problema es que al ser colaborativa hay algunos productos duplicados y algún dato no parece muy fiable.
- **API de Spoonacular** <sup>18</sup>: Proporciona acceso a información nutricional, recetas, detalles de ingredientes y planificación de comidas. Ofrece información nutricional de comida (aunque no muy extensa), recetas y menús. Tiene una buena documentación y permite 60 solicitudes por minuto.
- **API de MyFitnessPal** <sup>19</sup>: Permite acceder a datos nutricionales y de ejercicio. No se puede acceder ya que hay que rellenar un formulario y según consideren el valor de tu aplicación la aceptan, aunque avisan de que aceptan muy pocas aplicaciones.
- **API de FatSecret** <sup>20</sup>: Permite acceso a información sobre alimentos, recetas, datos nutricionales y más. Hay una buena cantidad en sus alimentos y ofrece bastante información sobre nutrientes pero hay muy pocas recetas.

---

<sup>14</sup><https://fdc.nal.usda.gov/download-datasets.html>

<sup>15</sup><https://developer.edamam.com/edamam-nutrition-api>

<sup>16</sup><https://developer.nutritionix.com/>

<sup>17</sup><https://world.openfoodfacts.org/data>

<sup>18</sup><https://spoonacular.com/food-api>

<sup>19</sup><https://developer.underarmour.com/docs/myfitnesspal>

<sup>20</sup><https://platform.fatsecret.com/api/>

- **API de CalorieNinjas** <sup>21</sup>: Ofrece información sobre calorías y datos nutricionales de alimentos. Ofrece 10000 solicitudes mensuales pero no nos da información sobre minerales en los datos nutricionales.

## 3.2. Aplicaciones similares

Una forma para ver si merece la pena hacer una aplicación que cumpla todos los requerimientos, es observar la “competencia”. Observar si en el mercado existe alguna aplicación cómo la que tenemos pensada de hacer. En este caso, se han buscado las que mejores servicios dan y las más populares, y hemos encontrado las siguientes:

- **Cronometer**: Cronometer es un servicio que se puede usar mediante aplicación móvil u ordenador. Permite hacer un seguimiento exhaustivo de la alimentación con una gran cantidad de nutrientes y muy visual. Se pueden ver gráficas sobre los últimos días y añadir información sobre el ejercicio realizado y biometría personal. No obstante muchos de los servicios son de pago, como los de biometría o ver los datos nutricionales de los alimentos, solo los puedes ver una vez los hayas añadido a la dieta. <sup>22</sup>
- **FatSecret**: Esta web permite hacer un seguimiento nutricional básico aunque también ofrece alguna otra opción interesante. Entre ellas, se pueden consultar recetas, tiene una sección donde añadir y consultar las calorías consumidas en función del ejercicio realizado y sobre todo un tipo de feed donde los usuarios pueden compartir noticias. Sin embargo, no es muy intuitiva y a pesar de que es gratis no ofrece mucha información sobre los nutrientes, tan solo acerca de las calorías, grasas, azúcar y sal. <sup>23</sup>
- 

---

<sup>21</sup><https://calorieninjas.com/>

<sup>22</sup><https://cronometer.com/>

<sup>23</sup><https://www.fatsecret.es/>

# 4. Captura de Requisitos

En esta sección se identificarán los requisitos que debe cumplir la aplicación y quienes van a ser sus usuarios.

## 4.1. Documento de Requisitos

En este proceso del desarrollo del software se han identificado, y analizado las necesidades y expectativas de los usuarios y otras partes interesadas con respecto a la aplicación. Sirve para que el producto final cumpla con las especificaciones deseadas y proporcione valor a sus usuarios. Se han distinguido dos tipos de requisitos:

### 4.1.1. Requisitos Funcionales

Los requisitos funcionales son los que describen las funciones específicas que el sistema debe cumplir. Son el comportamiento de la aplicación, como responde a ciertas entradas y las condiciones bajo las cuales este comportamiento ocurre:

- **Registro de Usuarios**
  - Los usuarios deben poder registrarse en la plataforma proporcionando información básica como nombre, correo electrónico y contraseña.
  - Verificación del correo electrónico para asegurar la autenticidad de los usuarios.
- **Inicio de Sesión**
  - Los usuarios deben poder iniciar sesión utilizando su correo electrónico y contraseña.
  - Los usuarios pueden recuperar su contraseña en el caso de que la olviden.
- **Gestión de Recetas**
  - Los usuarios registrados deben poder consultar y ver recetas.
  - Filtrado de recetas por diversos criterios, incluyendo ingredientes de temporada.
  - Posibilidad de buscar recetas por nombre, ingredientes o tipo de comida
- **Gestión de Productos de Temporada**
  - El usuario puede consultar los productos de temporada nacionales.
- **Blog**
  - Los usuarios pueden acceder a la sección de blog donde podrán leer artículos relacionados con consejos veganos, vida saludable...

- **Seguimiento Nutricional**

- El usuario podrá añadir y borrar comidas que haya realizado a lo largo de los días y en distintos días.
- El usuario podrá ver las comidas que ha realizado y ver de qué se conforma cada comida.
- El usuario podrá consultar su seguimiento, donde se le mostrará el progreso nutricional diario, semanal o mensual respecto a sus objetivos. Por otro lado, también podrá revisar los objetivos nutricionales recomendados por cada etapa de la vida.

#### **4.1.2. Requisitos no Funcionales**

Por otro lado, los requisitos no funcionales son aquellos que describen las características de un sistema que no está directamente relacionadas con las funciones específicas que realiza, sino con la forma que estas funciones se llevan a cabo:

- **Seguridad**

- Autenticación y autorización: Sólo los usuarios autorizados pueden acceder a todas las funcionalidades.
- Confidencialidad: Protección de los datos contra accesos no autorizados.
- Integridad: Asegurar que los datos no puedan ser eliminados ni alterados sin autorización.
- Disponibilidad: Sistema siempre disponible y funcionando cuando se necesite.

- **Rendimiento**

- Tiempo de Respuesta.
- Capacidad de Procesamiento.
- Escalabilidad

- **Usabilidad**

- Interfaz de usuario: Deber ser una aplicación intuitiva y fácil de utilizar para que apetezca utilizarla.
- Accesible: Debe poder ser utilizada por cualquier tipo de persona sin excepción, independientemente de edad, género, condición física...
- Consistencia: El diseño debe ser uniforme.

- **Portabilidad**

- Compatibilidad con diferentes plataformas: Se debe poder ejecutar en diferentes sistemas operativos y navegadores.
- Compatibilidad con diferentes dispositivos: Se debe poder ejecutar y usar en ordenadores, móviles, tablets...
- 

- **Mantenibilidad**

- Modularidad: Diseñada de manera que sea fácil de modificar y mantener.

## 4.2. Casos de Uso

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. Además, la jerarquía de actores de los casos de uso se relaciona con la categorización de los usuarios que interactúan con el sistema en distintos niveles de responsabilidad o autoridad, junto con sus respectivas funciones o tareas asociadas.

Por su parte, mediante la jerarquía de actores podemos simplificar la información sabiendo también qué casos de uso puede heredar un actor de otro. Sin embargo, en este caso solo tenemos los actores *Usuario identificado* y *Usuario no Identificado* como se muestra en la imagen.



**Figura 5:** Diagrama de Casos de Uso

Por otro lado, una vez mostrada la jerarquía para nuestros casos de uso, se han identificado los siguientes casos de uso para los actores existentes:

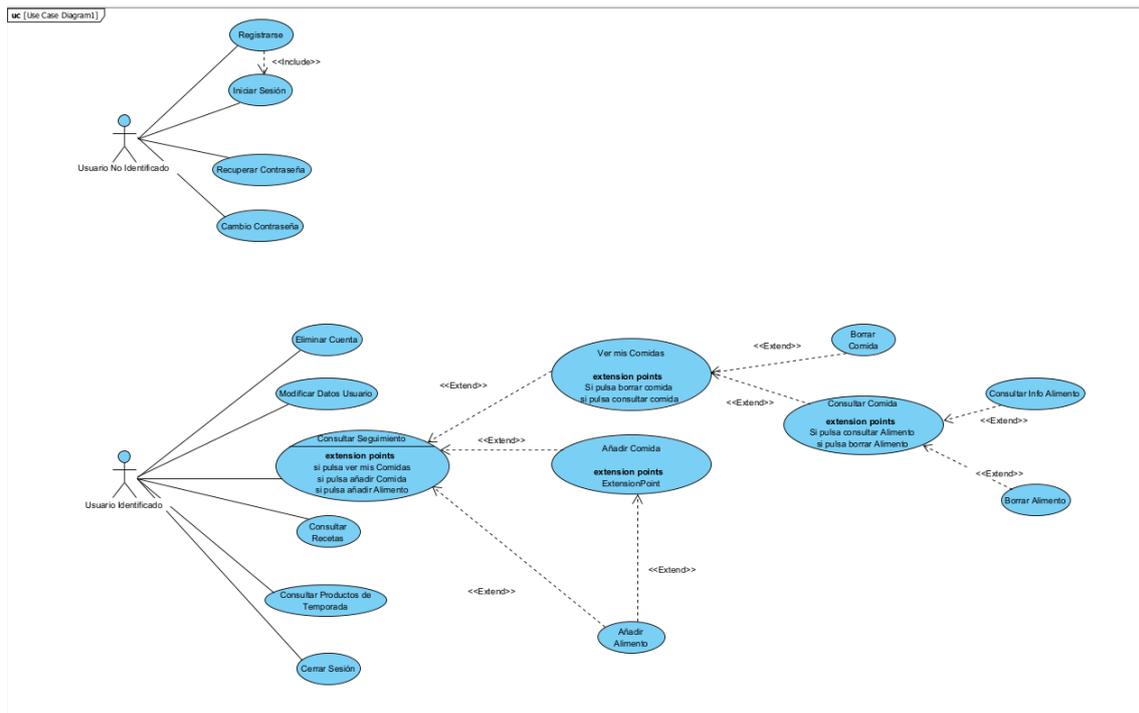


Figura 6: Diagrama de Casos de Uso

#### 4.2.1. Usuario no Identificado

Para el usuario sin identificar, es decir, el usuario que no tiene cuenta o tiene cuenta pero se ha cerrado la sesión se han encontrado los siguientes casos de uso:

- **Registrarse:** Un usuario ajeno al sistema se registra introduciendo sus datos personales como el correo, nombre o peso y altura. La cuenta se queda sin verificar hasta que el usuario verifica su cuenta mediante el correo.
- **Iniciar Sesión:** Un usuario cualquiera accede al sistema mediante sus credenciales. Para el correcto inicio de sesión las credenciales deben estar en el sistema, deben ser correctas y el usuario debe estar verificado.
- **Recuperar Contraseña:** Un usuario ajeno hace una solicitud de cambio de contraseña introduciendo el correo con el que registró su cuenta.
- **Cambiar Contraseña:** Un usuario dentro del sistema pero sin estar logueado cambia su contraseña mediante el mensaje que ha recibido en su cuenta de correo.

#### 4.2.2. Usuario Identificado

Para el usuario identificado, es decir el que ha introducido sus credenciales válidas y ha accedido a todas las funciones se han identificado los siguientes casos de uso.

- **Cerrar Sesión:** Se cierra la sesión de la cuenta asociada a la solicitud pasando a ser un usuario ajeno al sistema.
- **Eliminar Cuenta:** El usuario que hace la solicitud de borrado de cuenta elimina su cuenta de la web y pasa a ser un usuario sin acceso.
- **Modificar Datos Usuario:** El usuario modifica uno o varios datos

- **Consultar Recetas:** El usuario visualiza todas las recetas existentes del sistema.
- **Consultar Productos de Temporada:** El usuario visualiza todos los productos existentes en 3 grupos: en temporada, fuera de temporada y recién entrado/ a punto de salir.
- **Consultar Seguimiento:** Al usuario se le mostrará el avance sobre sus nutrientes consumidos respecto a sus objetivos. Además, tendrá la opción de añadir comida y añadir alimento, ver las comidas o las ingestas diarias recomendadas.
- **Consultar Objetivos:** El usuario podrá consultar las ingestas diarias recomendadas de los informes DRI (Dietary Reference Intakes). [13]
- **Añadir Comida:** El usuario añade a sus Comidas la comida registrada.
- **Añadir Alimento:** El usuario añade a su Comida el Alimento que ha introducido.
- **Borrar Comida:** El usuario Borra una determinada comida con todos sus respectivos Alimentos.
- **Borrar Alimento:** El usuario borra el Alimento de una Comida
- **Ver mis Comidas:** El usuario puede visualizar todas las comidas que ha realizado y tiene la opción de borrar o visualizar cada comida.
- **Consultar la información de un Alimento:** El usuario puede consultar todos los nutrientes aportados por ese Alimento.
- **Consultar Comida:** El usuario visualizar todos los Alimentos de una comida que ha introducido y tiene la opción de borrar o visualizar cada Alimento.

### 4.3. Modelo de Dominio

El diseño de una base de datos para una aplicación web es esencial para garantizar el rendimiento eficiente, escalabilidad, seguridad, facilidad de mantenimiento y flexibilidad a lo largo del tiempo. Es una parte importante a decidir antes de empezar el proyecto y tratar de definirlo bien desde el principio pensando en la forma más eficiente para nuestros casos de uso de manera que nuestras consultas y accesos a la base de datos sean lo más eficiente posibles. Buscando un diseño sólido que facilite el análisis y la coherencia de los datos para que contribuya a la funcionalidad y éxito a largo plazo de la aplicación.

#### 4.3.1. Entidades

Estas son las entidades en el modelo de dominio:

- **Cliente:** Esta entidad representa a los usuarios registrados. Los usuarios tienen atributos como el nombre, correo electrónico, contraseña (que se guarda hasheada), etc.
- **Alimento:** Esta entidad representa los alimentos que tenemos en la base de datos, con su id de alimento, nombre y la descripción.
- **Comida:** La entidad Comida representa las comidas realizadas y que se guardan en la base de datos con un nombre, descripción, el tipo de la comida, la fecha y su id.

- **Nutriente:** Esta entidad representa los nutrientes que disponemos y de los que podemos dar información al cliente, identificado por id y con atributos como nombre, descripción y la unidad con la que se mide ese nutriente.
- **Recetas:** Esta entidad representa todas las recetas que disponemos en la base de datos, también identificada por un id y con atributos como nombre, ingredientes, tiempo de preparación, categoría o imagen, entre otros.
- **ArticulosDeTemporada:** Para poder mostrar mes a mes los artículos que están de temporada, fuera de temporada y los que acaban de entrar o acaban de salir se guarda en la base de datos los productos con sus meses para cada estado, un nombre, una descripción y un id.
- **ResetPassword:** Esta entidad representa las solicitudes que han hecho los usuarios para cambiar de contraseña con sus tokens para que sea único, representado por una fecha de solicitud y una de expiración, usado por si ya se ha hecho el cambio, el token y un id.
- **HistóricoDeContraseñas:** Para poder guardar todas las contraseñas usadas por los usuarios se utiliza la entidad HistóricoDeContraseñas representada por un id, la contraseña y la fecha de cambio.
- **Etapas\_Vida:** Esta entidad representa todas las etapas por las que una persona pasa y las que hay que tener en cuenta principalmente para saber cuales son los nutrientes que debemos alcanzar. Para identificar una etapa de otra se usa un id, y guarda el año y mes de inicio, el año y mes de fin y una descripción de la etapa.

#### 4.3.2. Relaciones

Estas son las relaciones en el modelo de dominio:

- **solicita 1..\*:** Es una relación entre ResetPassword y Cliente. Representa todos los cambios de contraseña que un usuario ha solicitado.
- **tiene 1..\*:** Es una relación entre las entidades Cliente e HistoricoDeContraseñas. Representa todas las contraseñas que cada usuario ha tenido hasta el momento para que no se puedan repetir.
- **consume \*.\*:** Es una relación entre las entidades Cliente y Nutriente. Representa la cantidad que lleva consumida cada usuario de cada nutriente. En la relación tienen el atributo fecha\_consumo y cantidad para poder filtrar la cantidad por días, semanas o meses.
- **tiene\_objetivo \*.\*:** Es una relación entre las entidades Cliente y Nutriente. Representa la cantidad que tiene como objetivo cada usuario de cada nutriente. En la relación tienen el atributo cantidad.
- **realiza 1..\*:** Es una relación entre las entidades Cliente y Comida. Representa las comidas que cada usuario ha realizado.
- **incluye \*.\*:** Es una relación entre las entidades Comida y Alimento. Representa todos los Alimentos por los que está conformado una comida en particular. Se guarda en la relación la unidad y la cantidad.
- **tiene\_alergia \*.\*:** Es una relación entre las entidades Alimento y Cliente. Representa las alergias que cada cliente tiene a los Alimentos. Además, se conserva la gravedad y los síntomas para ese Alimento-Cliente.

- **contiene \*.\***: Es una relación entre las entidades Nutriente y Alimento. Representa la cantidad de nutrientes que tiene cada Alimento. Tiene un atributo cantidad donde se guarda la cantidad para 100 gr/ml.
- **necesita \*\*.\*\***: Es una relación entre las entidades Etapa\_Vida y Nutriente que sirve para almacenar los objetivos que tienen los Clientes en sus distintas Etapas de la Vida. Guarda el atributo cantidad.
- **esta\_en 1.\*\***: Es la relación que une las entidades Etapa\_Vida y Cliente. Sirve para saber las distintas etapas de la vida en las que se puede encontrar un cliente.

## 5. Análisis y diseño

En cualquier desarrollo de una aplicación web, una de las tareas más importantes es el análisis y diseño de la estructura. Dependiendo de nuestros objetivos, el tipo de página que queremos que sea, los datos que vayamos a servir e incluso los gustos, una estructura será más conveniente que otra.

### 5.1. Planteamiento del Desarrollo de Trabajo

Para obtener el mejor resultado posible en el proyecto se han seguido un conjunto de buenas prácticas. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar.

#### 5.1.1. Scrum

Indirectamente, se ha utilizado esta metodología de trabajo en el proyecto. Se define cómo un proceso en el que se realizan entregas parciales y regulares del trabajo. De esta forma, para cada reunión se comprueban los objetivos a realizar y se establecen nuevos en función de si se han realizado todos sin problema, falta alguno por realizar o no se ha conseguido llegar a los objetivos. De este modo, el tiempo de las entregas tiende a reducirse.

En el caso de este proyecto, en las primeras reuniones se definieron los objetivos generales y finales y objetivos para cada reunión. Y además, cada dos semanas se comprobaban los objetivos y se establecían nuevos tal y como se ha comentado.

#### 5.1.2. DevOps

De igual modo, se ha utilizado también el conjunto de prácticas DevOps. En primer lugar, mediante Github se llevó un control de versiones para poder actualizar el proyecto a medida que se iban añadiendo nuevas funciones. Esto permite poder volver a una versión anterior si surge algún problema durante los nuevos cambios. Por otro lado, mediante el Pipeline de Github Actions que se muestra en el apartado Pruebas, se permite automatizar el proceso de comprobación del código. Es decir, un script comprueba que no existe ningún error lógico de programación y que ésta se despliegue sin problemas.

### 5.2. Diagrama relacional de la Base de Datos

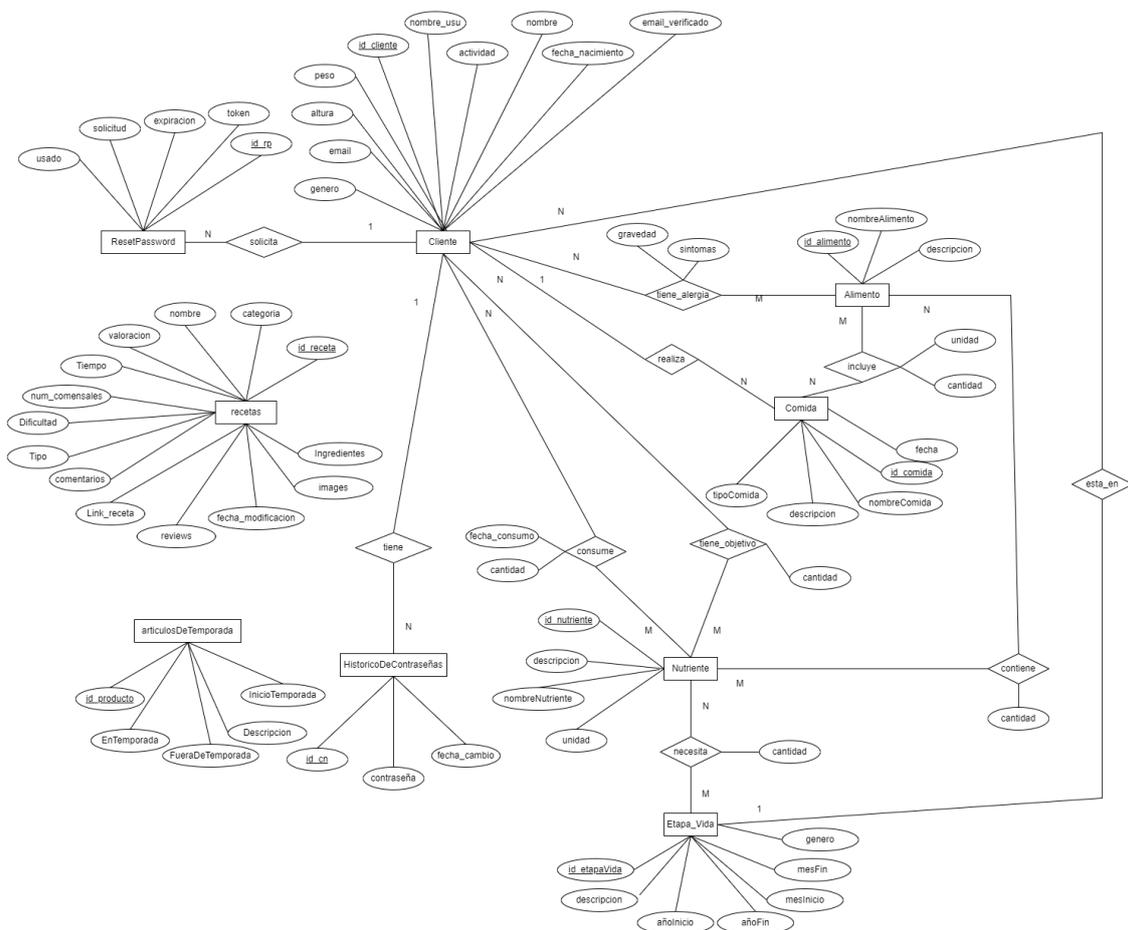
El diseño de una base de datos para una aplicación web es esencial para garantizar el rendimiento eficiente, escalabilidad, seguridad, facilidad de mantenimiento y

flexibilidad a lo largo del tiempo. Es una parte importante a decidir antes de empezar el proyecto y tratar de definirlo bien desde el principio pensando en la forma más eficiente para nuestros casos de uso de manera que nuestras consultas y accesos a la base de datos sean lo más eficiente posibles. Buscando un diseño sólido que facilite el análisis y la coherencia de los datos para que contribuya a la funcionalidad y éxito a largo plazo de la aplicación.

La siguiente tabla es el resultado de la *normalización* de una base de datos. La normalización de una base de datos es un proceso en el diseño de bases de datos relacionales que tiene como objetivo organizar la información de manera eficiente y eliminar la redundancia de datos.

Como resultado se ha obtenido una base de datos formada por 13 tablas. Algunas de las tablas existen para evitar la redundancia de datos, un problema que en etapas iniciales de un proyecto no suele ser ninguna amenaza, pero a medida que evoluciona la aplicación aumenta su gravedad. En la siguiente imagen se muestra el esquema de la base de datos.

Este es el diagrama modelo Entidad-Relación asociado al trabajo y que representa las entidades y relaciones recién comentadas:



**Figura 7:** Diagrama Entidad-Relación VeganMania

Por otro lado, para poder implementar la base de datos necesitamos relaciones (tablas). Para ello, mediante el algoritmo de los 7 pasos hemos transformado el diagrama ER en el Modelo Relacional. Es el que nos va a ayudar a implementar las tablas de la base de datos.

## 5.2.1. Descripción de las tablas

A continuación, se realizará una breve descripción de cada tabla utilizada.

### 5.2.1.1. Tabla: Cliente

Esta es una de las tablas más importantes y sobre la que se construye la arquitectura. Contiene los datos de los usuarios de la aplicación. Esta información sirve para identificarlos y para utilizar su información personal para una experiencia personalizada. Está identificado por un id **id\_cliente** y para iniciar sesión se utilizan principalmente los atributos **email** y **contraseña**, donde el email es único. Además, para que los datos sean personalizados para ese usuario, se guarda el **peso**, **altura**, **género** y **actividad** de la persona. Además, se guarda su **nombre** y **nombreUsuario**. Por último, para poder controlar si el usuario ha verificado su cuenta mediante email, se utiliza un booleano que por defecto es False: **email\_verificado**.

### 5.2.1.2. Tabla Alimento

La tabla Alimento contiene toda la información sobre los alimentos de la base de datos. Está identificada por un id **id\_alimento**. Por otro lado, tiene un **nombreAlimento** y una **descripción**.

### 5.2.1.3. Tabla Comida

La tabla Comida contiene la información sobre las comidas realizadas por los usuarios. Una comida está identificada por un **id\_comida** y además, tiene un **nombre**, **tipoComida**, **descripción** y una **fecha** que es la fecha en la que se realiza la comida. Por otro lado, tiene una clave foránea **id\_cliente** que representa el id del Cliente que modificó la contraseña

### 5.2.1.4. Nutriente

Esta tabla guarda la información sobre los 32 nutrientes de los que tenemos y damos información. Un nutriente tiene un id que lo identifica **id\_nutriente**, un nombre para poder diferenciarlo **nombre**, una **descripción** y finalmente se guarda también la unidad, porque depende de qué nutriente sea tendrá una unidad u otra, por ejemplo, para las calorías kcal y para las vitaminas nanogramos.

### 5.2.1.5. Recetas

En la tabla Recetas se guardan todas las recetas de las que se disponen en la base de datos. Como nosotros no tenemos toda la información acerca de las recetas y no nos pertenecen, mostramos lo básico y un link a la receta original donde hay más información.

Una receta se identifica por un id **id**. Entre sus otros atributos más importantes, está el **nombre** de la receta para poder diferenciarla, un atributo **ingredientes** donde se guardan todos los productos necesarios con su cantidad para poder realizar la receta, el **link** que nos redirige a la receta completa o la **categoría** que nos permite saber qué tipo de receta es: Vegetariana, de Carne, de Pescado, de Sopas...

#### 5.2.1.6. Tabla HistoricoDeContraseñas

Para guardar todas las contraseñas utilizadas por los usuarios se utiliza la tabla HistoricoDeContraseñas. La tabla se identifica con **id**, guarda la contraseña utilizada por ese usuario en **contraseña** junto a la **fecha** en la que se cambió. Por otro lado, tiene una clave foránea **id\_cliente** que representa el id del Cliente que modificó la contraseña.

#### 5.2.1.7. Tabla ResetPassword

De forma análoga, para poder resetear la contraseña se utiliza la tabla ResetPassword. Sirve para que cuando un usuario pulse para cambiar la contraseña, se le envía al mail un correo con el link para modificar la contraseña más un **token** que está asociado a ese cambio de contraseña. Además, se guarda la fecha de solicitud y de expiración en **solicitud** y **expiración**, para que pasadas 24 horas el token deje de ser válido. Además, para que esa solicitud de cambio de contraseña solo se pueda utilizar una vez, se guarda un booleano **usado**. Por otro lado, tiene una clave foránea **id\_cliente** que representa el id del Cliente que modificó la contraseña. Por último, se identifica una solicitud entre otra mediante su **id**.

#### 5.2.1.8. Tabla articulosDeTemporada

En la tabla articulosDeTemporada se guarda toda la información acerca de los productos de temporada, principalmente frutas y verduras. Cada artículo se identifica con un **id** en el atributo **enTemporada** se guardan todos los meses en los que esa fruta está en temporada, y de mismo modo para los atributos **inicioTemporada** y **fueraDeTemporada** para los meses en los que acaban de entrar o están a punto de salir y los meses que están fuera de temporada respectivamente. Finalmente, también tiene un **nombre** y una **descripción**

### 5.2.2. Tabla tiene\_objetivo

En esta tabla se guardan todos los objetivos que tienen los usuarios para cada Cliente. Un objetivo está identificado por el **id\_cliente** y el **id\_nutriente** que a su vez son claves foráneas con Cliente y Nutriente respectivamente como referencia. Además, se guarda la **cantidad** para saber el número que tiene como objetivo.

#### 5.2.2.1. Tabla consume

En esta tabla se guardan todos los nutrientes consumidos por cada cliente de nuestra web. Para ello, la tabla se identifica mediante el **id\_cliente**, el **id\_nutriente** y la **fecha**, ya que un cliente puede consumir el mismo nutriente en distintos días. El **id\_cliente** hace referencia a la tabla Cliente mediante que **id\_nutriente** hace referencia a la tabla Nutriente. Además, se guarda la **cantidad** que lleva consumida el cliente de ese nutriente para esa fecha.

#### 5.2.2.2. Tabla incluye

En la tabla incluye se conservan todos los alimentos que están incluidos para una determinada comida. Para ellos, la tabla se identifica por **id\_alimento** y

**id\_comida**. Donde ambos ids son claves foráneas que hacen referencia a las tablas Alimento y Comida respectivamente. Por último, se guarda en la tabla la **cantidad** ingerida de ese alimento junto a su **unidad**.

#### 5.2.2.3. Tabla tiene\_alergia

Para guardar las posibles alergias de los usuarios de la base de datos, se utiliza la tabla tiene\_alergia. Dicha tabla se identifica mediante el **id\_cliente** y el **id\_alimento**, siendo ambos atributos claves foráneas y primarias de la tabla, con referencia a las tablas Cliente y Alimento respectivamente. Además, se guarda la **gravedad** y los **síntomas** que ese alimento puede ocasionar al usuario.

#### 5.2.2.4. Tabla contiene

En la tabla contiene se guarda la cantidad de nutrientes que tiene determinado alimento. Para ello, se utiliza **id\_alimento** e **id\_nutriente** como claves principales y foráneas con referencia a las tablas Alimento y Nutriente respectivamente. Por último, se guarda la **cantidad** que indica la cantidad que tiene ese alimento del nutriente.

#### 5.2.2.5. Tabla Etapa\_Vida

En esta tabla se guarda toda la información sobre las distintas fases por las que pasa una persona en función de la edad y género. Está identificado por un **id\_etapa** y contiene la información para saber la fecha de inicio de la etapa mediante **añoInicio** y **mesInicio** y la fecha fin mediante **añoFin** y **mesFin**. Por último, cada etapa tiene una **descripcion**.

#### 5.2.2.6. Tabla necesita

La tabla necesita contiene la información para recopilar la cantidad de nutrientes que necesita el Cliente en función de la etapa de la vida en la que se encuentre. Para ello, tiene como clave primaria el **id\_etapaVida** e **id\_nutriente** de las tablas Etapa\_Vida y Nutriente respectivamente que a su vez son claves foráneas. Por otro lado, tiene un atributo **cantidad** que es la que indica el número recomendado del nutriente para esa etapa.

### 5.3. Diagrama de Clases

### 5.4. Arquitectura

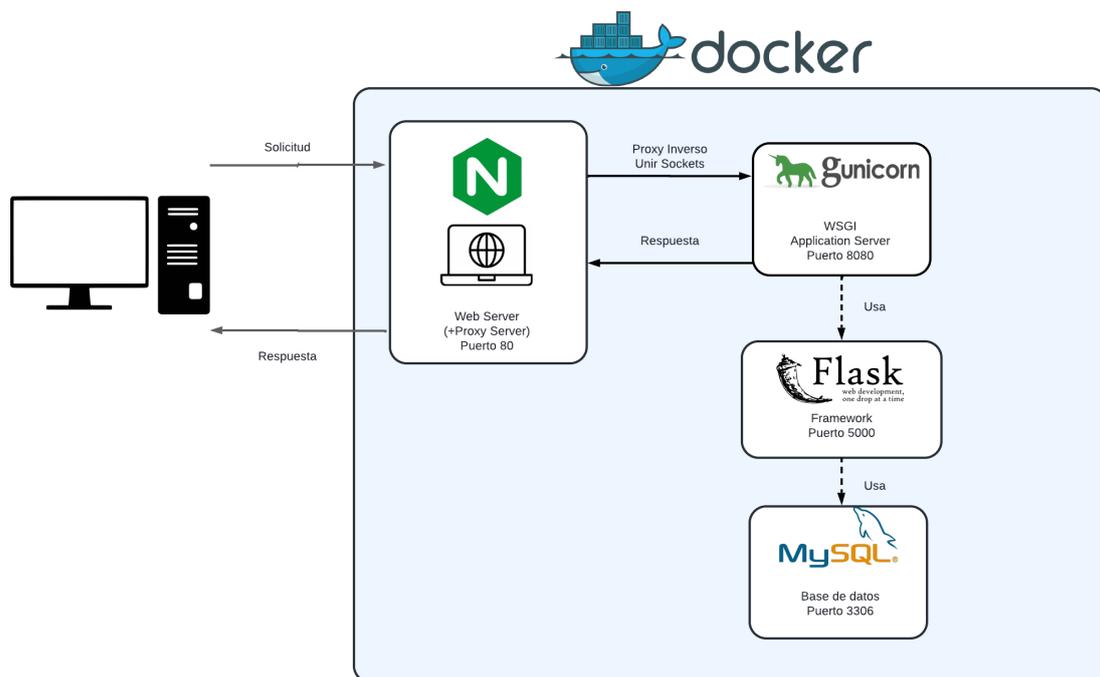
VeganMania es una aplicación desarrollada utilizando **Flask**, **MySQL** y **Nginx**. La aplicación está en un entorno Dockerizado y sigue una arquitectura comúnmente conocida como arquitectura de contenedores. En la mayoría de páginas webs que emplean usuarios con autenticación, se divide la aplicación en dos secciones. En primer lugar, el **back-end**. A veces denominado servidor, el back-end es la parte de la aplicación que administra la funcionalidad general de la aplicación. De manera que cuando el usuario interactúa con el front-end, la interacción envía una solicitud al back-end en formato HTTP. El back-end procesa la

solicitud y devuelve una respuesta. Pudiendo interactuar con micro-servicios, bases de datos o APIs de terceros (Edamam, gmail...). Por otro lado, el **front-end**, es la parte frontal de un sitio web o aplicación. Sirve para realizar la interfaz del sitio web, de forma que se trate que el usuario tenga una buena experiencia dentro del sitio o aplicación.

En este caso, nuestro back-end está compuesto por Flask y MySQL mientras que el front-end por Nginx y HTML+CSS+JS y Bootstrap que son los que le dan el aspecto a la web. No obstante, de intermediario entre Flask y Nginx para servir las interfaces por el puerto 80, está **Gunicorn**. Gunicorn es un servidor de aplicaciones WSGI que se encarga de manejar solicitudes HTTP y pasarlas a la aplicación Flask, y luego devolver las respuestas generadas por Flask a los siguientes.

En este sentido, se utiliza la combinación de Flask, Gunicorn y Nginx que sigue una arquitectura comúnmente utilizada para aplicaciones web Python. No obstante, para su correcto funcionamiento dentro de los usos posibles de Nginx, se usa como proxy inverso y servidor web. De tal forma que como se ha explicado, recibe las solicitudes de los clientes, le transmite a Flask las solicitudes y la respuesta de Flask la sirve.

Estos 3 servicios, integrados en **Docker**, que crea entornos virtuales para Nginx, MySQL y Flask, y establece los puertos de cada contenedor y una red para que se puedan comunicar entre ellos. Para ver esto de una forma visual se incluye la siguiente figura:



**Figura 8:** Arquitectura Web VeganMania

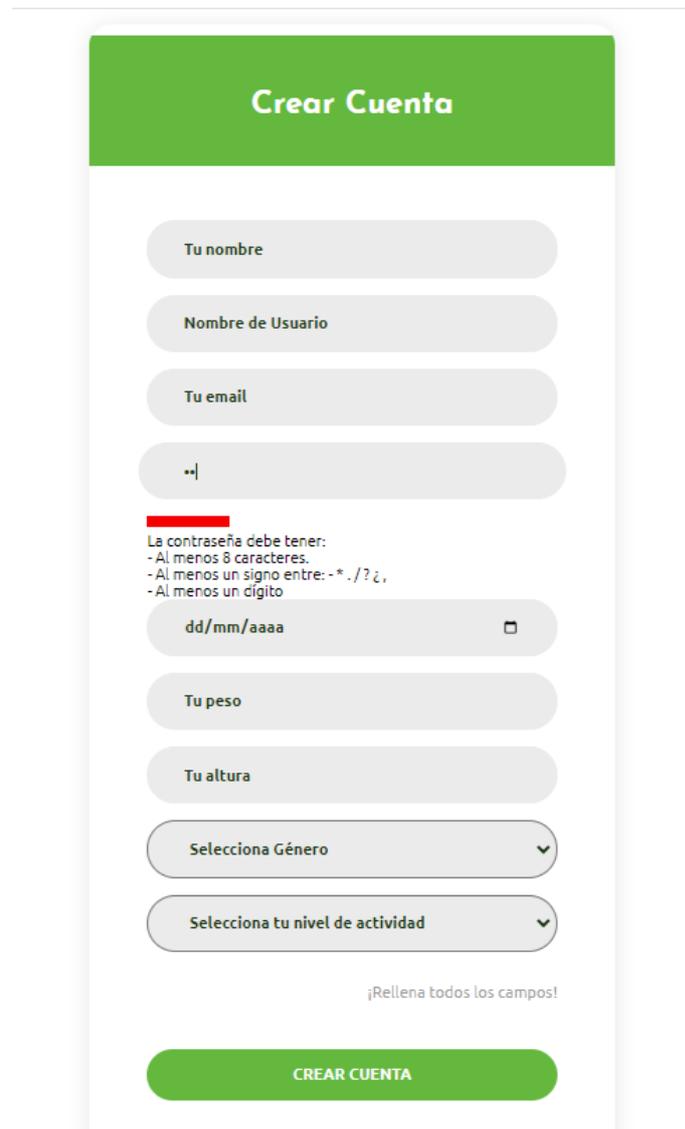
Por otro lado, cabe mencionar un último contenedor que es **Adminer**. No se ha comentado en la arquitectura porque de cara al usuario final no tiene repercusión. Ha sido de ayuda para poder comprobar que la información se registraba, modificaba y eliminaba correctamente de la base de datos, y poder visualizar la información de forma algo visual.

## 5.5. Diseño de Interfaces

Finalmente, se van a mostrar y explicar algunas de las interfaces diseñadas para darle apariencia al esqueleto diseñado en el *backend*. Se ha tratado de hacer la interfaz de manera que sea atractiva, intuitiva y fácil de usar. De igual modo, se ha planteado la web para que sea *responsive*, principalmente pensado para ordenadores o *tablets* pero también utilizable en móviles.

### 5.5.1. Registro

En la interfaz de registro, el usuario deberá introducir sus datos personales que son los que figurarán en su usuario, aunque posteriormente podrá cambiarlos. Por otro lado, si alguno de los campos no cumple con los requisitos, se deja algún campo en blanco, o el correo introducido existe en la base de datos, se imprimirá un mensaje de error con el motivo. Por otro lado, si los datos introducidos son correctos, se le enviará un correo de verificación de cuenta al usuario, donde si el usuario verifica su cuenta, se activará y se le enviará el correo de bienvenida.



The image shows a mobile registration form titled "Crear Cuenta". The form is contained within a white rounded rectangle with a green header bar. The fields are as follows:

- Tu nombre
- Nombre de Usuario
- Tu email
- |
- La contraseña debe tener:
  - Al menos 8 caracteres.
  - Al menos un signo entre: - \* . / ? ! ,
  - Al menos un dígito
- dd/mm/aaaa
- Tu peso
- Tu altura
- Selecciona Género
- Selecciona tu nivel de actividad

¡Rellena todos los campos!

CREAR CUENTA

Figura 9: Interfaz de Registro

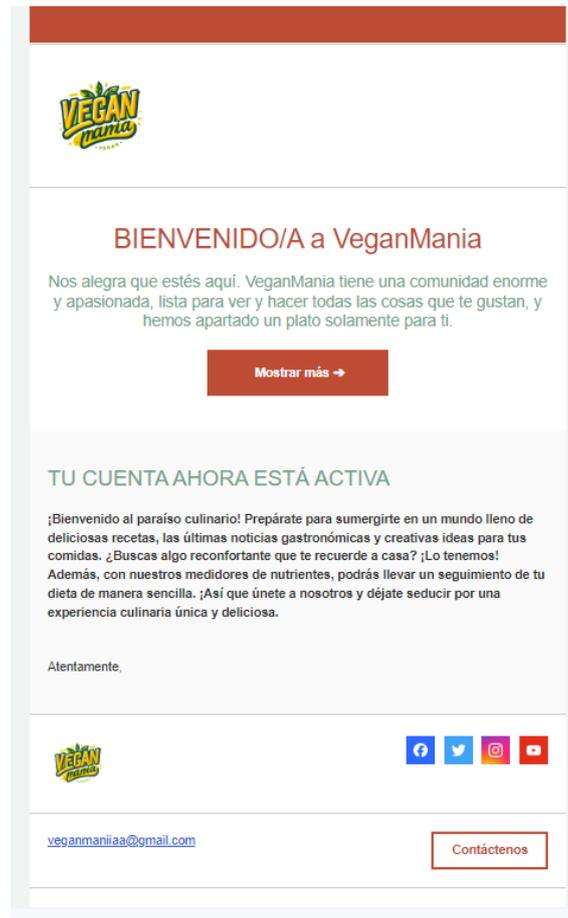
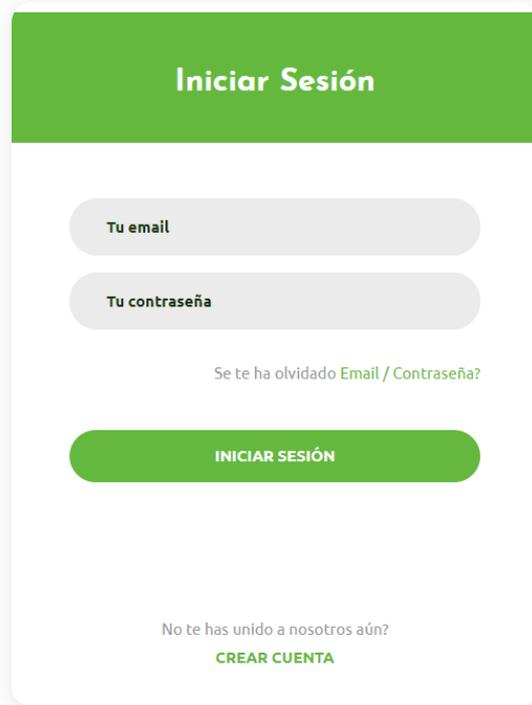


Figura 10: Mail de Bienvenida una Vez Registrado y Verificado

## 5.5.2. Inicio de Sesión

Por otro lado, en esta interfaz si el usuario trata de insertar alguno de los campos en blanco o sus credenciales son incorrectas, no se le permitirá al usuario acceder imprimiendo un mensaje con el motivo. Del mismo modo, si intenta acceder a una cuenta todavía no validada se le notificará que tiene pendiente la verificación. Si por el contrario está todo en orden, se iniciará la sesión del usuario y accederá a sus funciones.



Interfaz de Inicio de Sesión. El formulario tiene un encabezado verde con el título "Iniciar Sesión". Contiene dos campos de entrada de texto: "Tu email" y "Tu contraseña". Debajo de los campos, hay un enlace que dice "Se te ha olvidado Email / Contraseña?". Un botón verde con el texto "INICIAR SESIÓN" está centrado. En la parte inferior, hay un enlace que dice "No te has unido a nosotros aún?" y un botón verde con el texto "CREAR CUENTA".

**Figura 11:** Interfaz de Inicio de Sesión

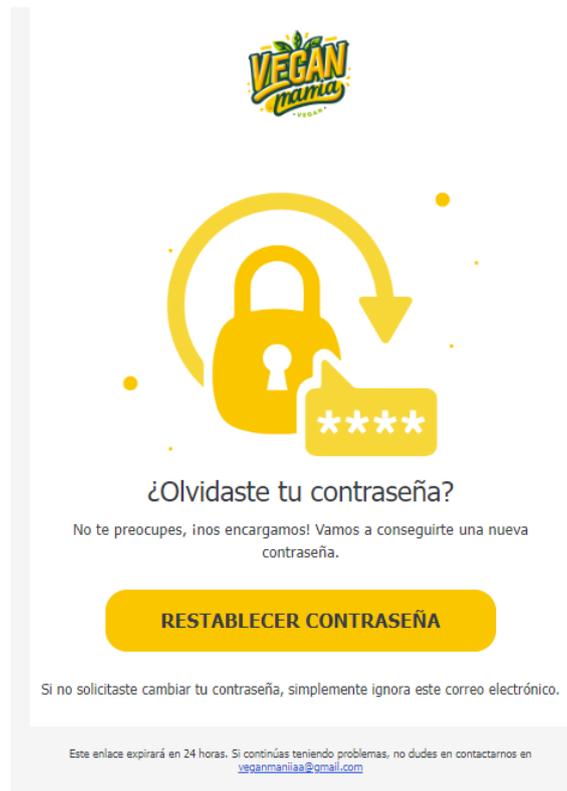
### 5.5.3. Recuperar Contraseña

Si el usuario se olvida de la contraseña o desea modificarla, podrá hacer una solicitud cada 24h para cambiar de contraseña. Donde la contraseña tendrá que cumplir los mismos requisitos que en el registro y no se podrá introducir ninguna que se haya usado antes. Si todo está en orden la contraseña se modificará correctamente.



Interfaz Solicitud Cambio de Contraseña. El formulario tiene un encabezado verde con el título "Recuperar Contraseña". Contiene un campo de entrada de texto: "Tu email". Debajo del campo, hay un botón verde con el texto "RECUPERAR CONTRASEÑA". En la parte inferior, hay un enlace que dice "No te has unido a nosotros aún?" y un botón verde con el texto "CREAR CUENTA".

**Figura 12:** Interfaz Solicitud Cambio de Contraseña



**Figura 13:** Mail para Restablecer Contraseña

The image shows a user interface for changing a password. The title is 'Cambia la Contraseña' in bold. Below the title is the label 'Nueva Contraseña' followed by a text input field. At the bottom of the form is a green button with the text 'Restablecer Contraseña'.

**Figura 14:** Interfaz para Cambiar de Contraseña

#### 5.5.4. Verificación de Cuenta

En esta parte se produce la verificación de la cuenta una vez se ha registrado. Recibes un correo al mail que has registrado y si pulsas el botón te dirigirá a la interfaz donde se avisará de que se ha verificado la cuenta.



**¡Hola David!**

Gracias por registrarte en VeganMania.  
Para completar el proceso de registro, por favor verifica tu dirección de correo electrónico haciendo clic en el siguiente enlace:

[Verificar correo electrónico](#)

**Figura 15:** Mail Verificar Cuenta

## Verificación Completa

Tu correo ha sido verificado correctamente. Ya puedes iniciar sesión.

[Cerrar y acceder](#)

**Figura 16:** Interfaz que Completa la Verificación

### 5.5.5. Sección Sobre Nosotros

En esta parte se hará un breve resumen de la motivación del proyecto, quiénes somos y cuáles son los objetivos.

---

#### ¿QUÉ ES VEGANMANIA?

Veganmania es una aplicación diseñada para ayudarte a cumplir tus objetivos nutricionales, especialmente aquellos que son más difíciles de alcanzar para veganos y vegetarianos. Al eliminar productos de origen animal, algunos nutrientes pueden ser más difíciles de obtener, pero con Veganmania, esto ya no será un problema.

#### OBJETIVO DE VEGANMANIA

El objetivo principal de Veganmania es facilitar una alimentación vegana y vegetariana equilibrada, asegurando que puedas obtener todos los nutrientes esenciales. Además, promovemos el uso de productos de temporada y locales, proporcionándote recetas y un blog informativo para mejorar tu conocimiento alimenticio.

#### IMPORTANCIA DE LOS NUTRIENTES

Mantener un equilibrio adecuado de nutrientes es crucial para una buena salud. Estar por debajo de los niveles necesarios puede llevar a deficiencias que afecten tu bienestar, mientras que excederse en algunos nutrientes también puede ser perjudicial. Veganmania te ayuda a estar justo en los niveles necesarios, proporcionando herramientas para monitorear tu ingesta y ajustar tu dieta en consecuencia.

Con Veganmania, puedes ver cómo llegas a tus objetivos nutricionales diarios, semanales y mensuales. La app te permite llevar un seguimiento detallado de 32 nutrientes esenciales, asegurándote de que tu dieta sea equilibrada y adecuada para tus necesidades.

#### FUNCIONALIDADES

Veganmania ofrece una variedad de funcionalidades diseñadas para apoyar tu estilo de vida vegano o vegetariano:

Recetas basadas en productos de temporada y locales.

Blog informativo con artículos sobre nutrición y bienestar.

App de seguimiento que monitoriza 32 nutrientes esenciales, ayudándote a mantener un equilibrio nutricional adecuado.

Consejos y trucos para obtener los nutrientes que suelen ser difíciles de conseguir en una dieta vegana o vegetariana.

Seguimiento de tus objetivos nutricionales diarios, semanales y mensuales.

#### METODOLOGÍA

Para asegurar que las recomendaciones nutricionales sean precisas y fiables, Veganmania se basa en datos recopilados de diversas fuentes científicas y consultas con nutricionistas especializados en dietas veganas y vegetarianas. Utilizamos una combinación de investigación académica y experiencia práctica para crear una base de datos robusta y actualizada.

---

**Figura 17:** Interfaz Sobre Nosotros

### 5.5.6. Sección Modificar Usuario

En esta interfaz podremos además de modificar nuestros datos eliminar nuestro usuario si deseamos borrarlos la cuenta ya sea por usar otro mail o porque ya no se quiere volver a usar. El único campo que no se puede modificar es el de la contraseña, para ello habría que resetearla como se hace en el apartado 5.3.3.

## Modificar Datos

Modifica los campos que quieres cambiar, no hace falta que pongas todos unicamente los que veas necesarios

Tu nombre actual es: David

**Tu nombre**

Tu fecha de nacimiento actual es: 1988-08-20

**dd/mm/aaaa** 📅

Tu peso actual es: 65.20

**Tu peso**

Tu altura actual es: 175.00

**Tu altura**

Tu género actual es: Hombre

**Selecciona Género** ▼

Tu nivel de actividad actual es: Sedentario

**Selecciona tu nivel de actividad** ▼

**MODIFICAR DATOS**

puede modificar el email con el que te registraste, si quieres puedes la cuenta y crearte una nueva

**Figura 18:** Interfaz Modificar Datos

### 5.5.7. Interfaz Blog

En esta sección los usuarios pueden consultar y buscar artículos de su interés. Es la forma que se plantea también para mejorar el conocimiento y ayudar a la gente a tener más nociones sobre su alimentación.

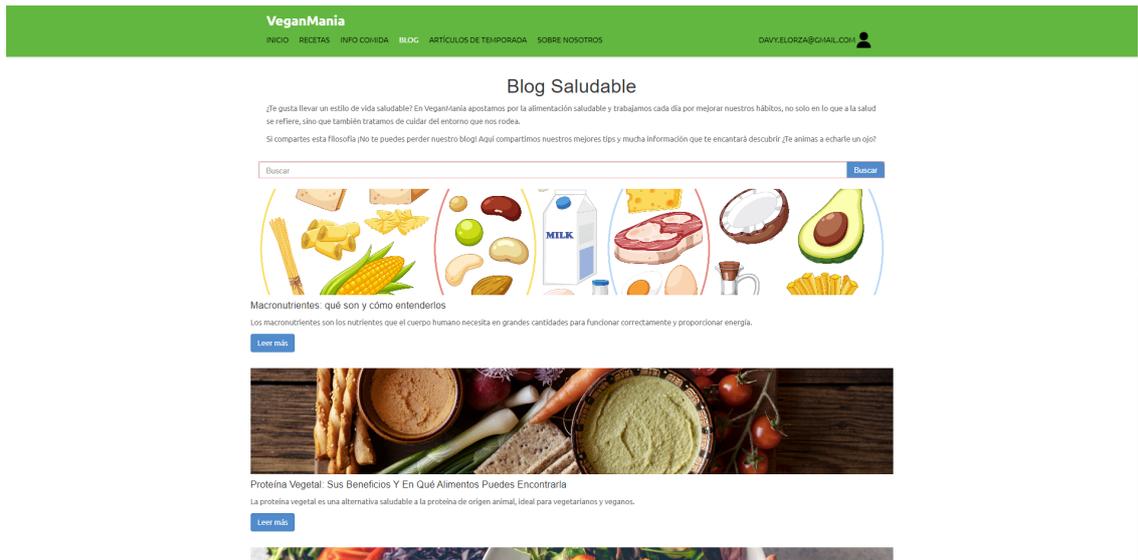


Figura 19: Interfaz Blog



Figura 20: Interfaz Artículo del Blog

### 5.5.8. Sección Recetas

En esta sección se pueden consultar todas las recetas. Si se abre por defecto se mostrarán todas aunque se puede filtrar por nombre, categoría, dificultad y tiempo de elaboración. Contiene paginación para poder navegar por las distintas recetas y si pulsas en alguna de las recetas se abrirá su información.



Figura 21: Interfaz de Recetas

### 5.5.9. Interfaz Artículos de Temporada

En esta interfaz se pueden consultar los productos de temporada nacionales agrupados artículos de temporada y fuera de temporada. Si pulsas en cualquier producto se abrirá una descripción y se podrán filtrar por nombre con el buscador.

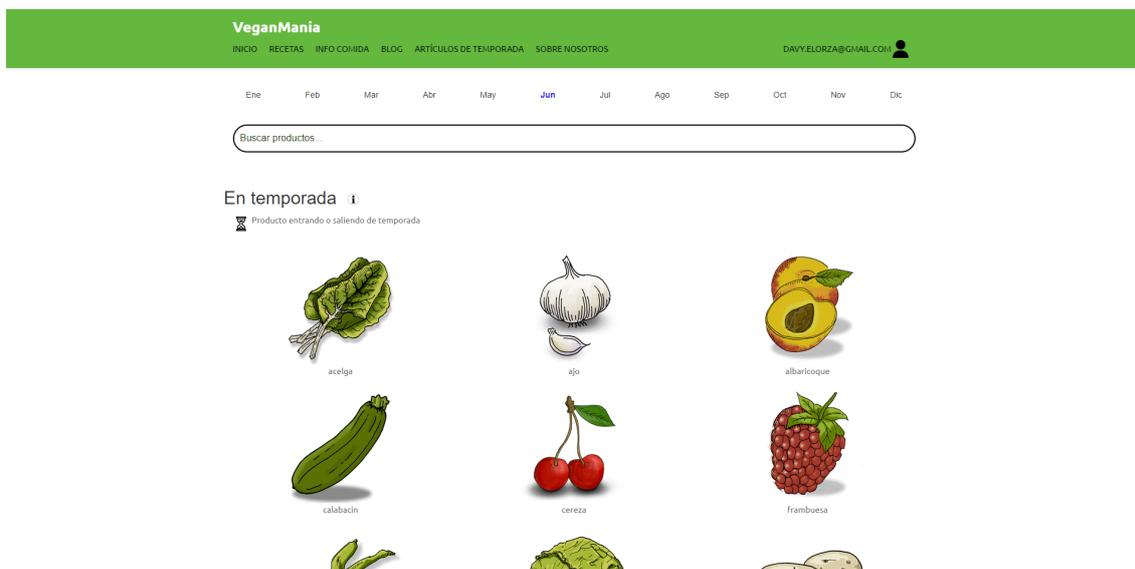


Figura 22: Interfaz de Artículos de Temporada

### 5.5.10. Interfaz Añadir Comida

Para añadir una nueva comida registraremos el nombre, tipo, descripción y fecha de la comida.

## Añadir Comida

Una comida está conformada por varios alimentos. Por ejemplo, en el desayuno podemos tener dos alimentos, un zumo de naranja y unas tostadas de jamón y luego en la comida podemos tener un primer plato que sean garbanzos, un segundo pollo y de postre yogur danone.

Una vez hayas añadido tu primera comida [añádele los alimentos](#).

**Nombre de la Comida:**

**Tipo de Comida:**

**Descripción:**

**Fecha:**

**Añadir**

**Figura 23:** Interfaz Añadir Comida

### 5.5.11. Interfaz Añadir Alimento

Para añadir alimentos a una comida específica, primero seleccionamos la comida principal y luego utilizamos la opción "añadir alimento". Aquí registramos cada alimento incluido en esa comida, proporcionando su nombre, descripción, tipo de comida y cantidad individualmente. Este proceso asegura que cada elemento esté correctamente documentado y asociado con la comida principal correspondiente.

## Añadir Alimento

Nombre del Alimento:

Descripción:

Comida:

Cantidad:

**Añadir**

Figura 24: Interfaz Añadir Alimento

### 5.5.12. Interfaz Mis Comidas

En esta interfaz se muestran todas las comidas que ha realizado el usuario agrupadas por el día de hoy, últimos 7 días y anteriores. Para cada comida están las opciones de VerComida y BorrarComida.

### Tus Comidas

Últimos 7 Días

Almuerzo 20-06-2024

Tipo: Almuerzo

Descripción: Fruta

Fecha: 2024-06-20

**Borrar Comida** **Ver Comida**

Comida 20-06-2024

Tipo: Comida

Descripción: Ensalada fresca y pechugas de pollo.

Fecha: 2024-06-20

**Borrar Comida** **Ver Comida**

Figura 25: Interfaz Mis Comidas

### 5.5.13. Interfaz Detalles de la Comida

En Mi Comida se podrá consultar los alimentos que componen la comida que se ha seleccionado para ver en la interfaz anterior. De forma parecida a las comidas, se puede ver el alimento o borrarlo.

## Detalles de la Comida

### Cena 20-06-2024

Descripción: Cena

Fecha: Crema de calabacín y salmón al horno.

#### Alimentos Incluidos

Crema  
Descripción: Crema de calabacín  
Cantidad: 175.00 gr  
[Eliminar Alimento](#) [Ver Alimento](#)

Pollo  
Descripción: Pechugas de pollo  
Cantidad: 160.00 gr  
[Eliminar Alimento](#) [Ver Alimento](#)

Figura 26: Interfaz detalles de la comida

#### 5.5.14. Interfaz Detalles Alimento

En esta interfaz se mostrará todos los nutrientes que contiene el alimento en función de la cantidad introducida.

#### Detalles del Alimento

Pollo  
Descripción: Pechugas de pollo

Nutrientes

- : Cantidad de energía proporcionada por el alimento 344.00  
Energía
- : Grasas totales presentes en el alimento 24.16  
Lípidos totales (grasas)
- : Grasas saturadas presentes en el alimento 6.90  
Ácidos grasos, saturados totales
- : Grasas trans presentes en el alimento 0.16  
Ácidos grasos, trans totales
- : Grasas monoinsaturadas presentes en el alimento 9.98  
Ácidos grasos monoinsaturados totales
- : Grasas poliinsaturadas presentes en el alimento 5.17  
Ácidos grasos poliinsaturados totales
- : Carbohidratos presentes en el alimento 0.00

Figura 27: Interfaz Detalles Alimento

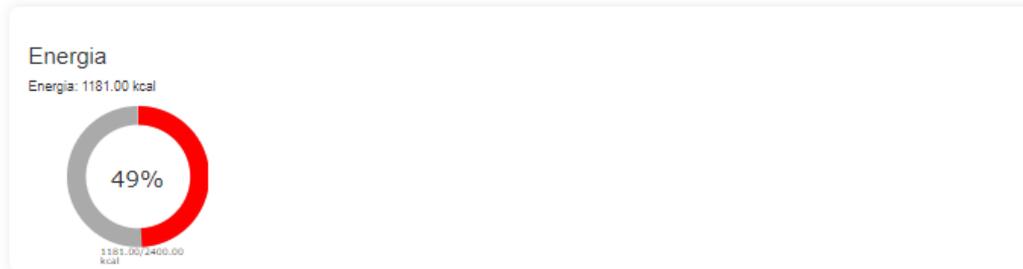
#### 5.5.15. Interfaz Inicio/Tu Seguimiento

Observamos los nutrientes consumidos. Los podemos ordenar en función de la fecha o grado de completitud.

## Tu Seguimiento

Añadir... Frecuencia Ordenar por porcentaje Ver Comidas Consultar Objetivos

### Energia



### Grasas

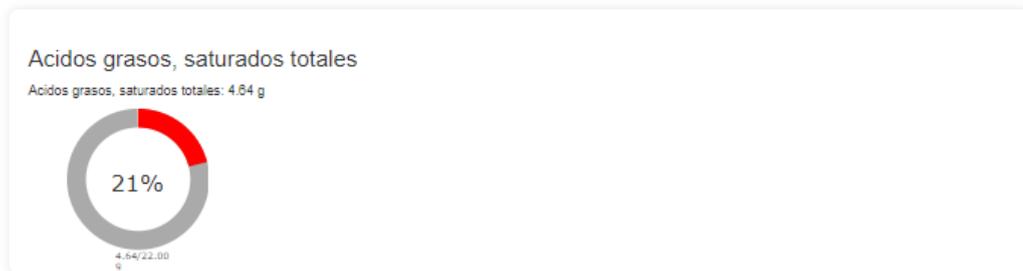
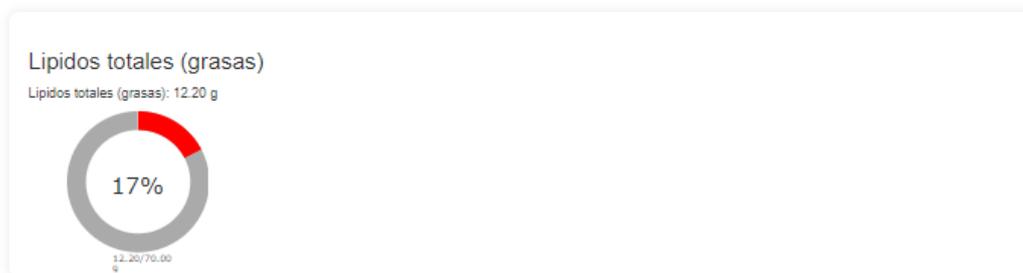


Figura 28: Interfaz Tu Seguimiento

### 5.5.16. Interfaz Consultar Objetivos

Al principio vemos una explicación sobre la importancia de llegar a los objetivos nutricionales, la cual incluye las fuentes de los datos. Seguido, podemos ver los objetivos nutricionales en función de la etapa de la vida en la que nos encontremos, teniendo la posibilidad de la filtración de varias etapas, incluyendo toda la etapa vital.

## Objetivos de Nutrientes por Etapa de Vida

La importancia de alcanzar los objetivos nutricionales y no excederlos radica en el equilibrio necesario para mantener la salud y el bienestar a lo largo de todas las etapas de la vida. Cada etapa de la vida tiene necesidades nutricionales específicas que deben ser satisfechas para apoyar el crecimiento, el desarrollo y el mantenimiento de las funciones corporales. Por ejemplo, los lactantes requieren nutrientes que favorezcan el rápido crecimiento y desarrollo del cerebro, mientras que los adultos mayores necesitan mantener la densidad ósea y la masa muscular.

No cumplir con los objetivos nutricionales puede llevar a deficiencias que afectan el funcionamiento del cuerpo, mientras que el exceso de nutrientes puede causar toxicidad y otros problemas de salud. Por ello, es crucial seguir las recomendaciones establecidas basadas en investigaciones científicas y proporcionadas por organizaciones como la Sociedad Española de Oncología Médica (SEOM), la Agencia Española de Seguridad Alimentaria y Nutrición (AESAN) y la Autoridad Europea de Seguridad Alimentaria (EFSA).

Las recomendaciones dietéticas ayudan a guiar la ingesta adecuada de macro y micronutrientes, asegurando que se obtengan los beneficios de una dieta equilibrada sin los riesgos asociados a la malnutrición o la sobrealimentación. Un balance adecuado de nutrientes no solo apoya el desarrollo físico, sino también el mental, emocional y la prevención de enfermedades crónicas. Por ejemplo, una ingesta suficiente de calcio y vitamina D es crucial para la salud ósea, mientras que el consumo adecuado de ácidos grasos omega-3 es esencial para la función cerebral y cardiovascular.

Las recomendaciones específicas para cada etapa de la vida tienen en cuenta factores como el género, la edad y las condiciones fisiológicas particulares (embarazo, lactancia, etc.). Esto asegura que cada individuo reciba el apoyo nutricional adecuado para sus necesidades únicas. Por ejemplo, durante el embarazo, se incrementa la necesidad de nutrientes como el ácido fólico y el hierro para apoyar el crecimiento y desarrollo del feto y la salud de la madre.

Para más detalles, puedes consultar las siguientes fuentes: [Soporte Nutricional \(SEOM\)](#), [Recomendaciones Dietéticas \(AESAN\)](#), [Dietary Reference Values \(EFSA\)](#).

**Figura 29:** Consultar objetivos 1

Género:  Rango de Edad:

Mujeres 31-50 años			
Nombre del Nutriente	Descripción	Unidad	Cantidad Necesaria
Energía	Cantidad de energía proporcionada por el alimento	kcal	2000.00
Lípidos totales (grasas)	Grasas totales presentes en el alimento	g	50.00
Ácidos grasos, saturados totales	Grasas saturadas presentes en el alimento	g	12.00
Ácidos grasos, trans totales	Grasas trans presentes en el alimento	g	0.00
Ácidos grasos monoinsaturados totales	Grasas monoinsaturadas presentes en el alimento	g	18.00
Ácidos grasos poliinsaturados totales	Grasas poliinsaturadas presentes en el alimento	g	12.00
Carbohidratos, por diferencia	Carbohidratos presentes en el alimento	g	130.00
Fibra dietética total	Fibra dietética presente en el alimento	g	28.00
Azúcares, total incluyendo NLEA	Azúcares totales presentes en el alimento	g	0.00
Proteína	Proteína presente en el alimento	g	48.00
Colesterol	Colesterol presente en el alimento	mg	0.00
Sodio Na	Sodio presente en el alimento	mg	1500.00
Calcio	Calcio presente en el alimento	mg	1000.00
Magnesio, Mg	Magnesio presente en el alimento	mg	320.00
Potasio, K	Potasio presente en el alimento	mg	2800.00
Hierro, Fe	Hierro presente en el alimento	mg	18.00
Zinc, Zn	Zinc presente en el alimento	mg	8.00
Fosforo, P	Fosforo presente en el alimento	mg	700.00
Vitamina A, RAE	Vitamina A presente en el alimento	mcg	700.00
Vitamina C, ácido ascórbico total	Vitamina C presente en el alimento	mg	75.00
Tiamina	Tiamina presente en el alimento	mg	1.10
Riboflavina	Riboflavina presente en el alimento	mg	1.10
Niacina	Niacina presente en el alimento	mg	14.00
Vitamina B-6	Vitamina B-6 presente en el alimento	mg	1.30
Folato, DFE	Folato presente en el alimento	mcg	400.00
Folato, comida	Folato de origen alimentario presente en el alimento	mcg	400.00

**Figura 30:** Consultar objetivos 2

### 5.5.17. Interfaz Tutorial

Este es el tutorial para que la gente pueda comprender mas fácilmente la funcionalidad de la web.

## Tutorial de la Aplicación

Aquí encontrarás información sobre cómo usar las funcionalidades básicas de la aplicación...



**Figura 31:** Interfaz Tutorial

## 6. Desarrollo

En esta sección se especificará como han implementado las diferentes partes de la aplicación. Desde la arquitectura de los contenedores hasta el funcionamiento interno y comunicación entre los contenedores de la aplicación.

### 6.1. Instalación

En primer lugar, se comentarán los pasos a seguir para y la instalación necesaria para poner en marcha la aplicación web. Se detallarán los requisitos del entorno de desarrollo y se proporcionarán instrucciones para instalar el software esencial.

#### 6.1.1. Preparación del Software Necesario

Como ya se ha mencionado se ha utilizado Docker para desplegar la web. De esta forma Docker empaqueta software en unidades estandarizadas (contenedores) que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. Ofrece escalabilidad y aseguran que la aplicación se ejecute de la misma forma en cualquier entorno.

Una forma fácil de entenderlo, es pensar que cada contenedor es una máquina virtual donde cada máquina ejecuta su parte y existe una comunicación entre ellas para desplegar la web. Por lo que en primer lugar hay que tener instalado Docker, en Windows mediante Docker Desktop [8] y en Linux siguiendo el siguiente tutorial [9].

##### 6.1.1.1. Configuración

Para configurar la aplicación hace falta en primer lugar crear el archivo `docker-compose.yml`. Es el encargado de crear una red entre los contenedores. En él habrá que especificar qué imagen se va utilizar para cada contenedor, configurando cada uno y la red conjunta. El archivo resultante es el siguiente:

```

version: "3"
services:
  app:
    build: ./app
    ports:
      - "5000:5000"
    depends_on:
      - db
    networks:
      - backend-network
      - frontend-network
    restart: always
    volumes:
      - ./app:/app

  db:
    build: ./db
    ports:
      - "3306:3306"
    volumes:
      #- ./db/init.sql:/docker-entrypoint-initdb.d/init.sql
      - ./data:/var/lib/mysql
    networks:
      - backend-network

  adminer:
    image: adminer
    container_name: adminer
    restart: always
    ports:
      - 8081:8080
    networks:
      - backend-network
      - frontend-network

  nginx:
    image: nginx:latest
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/nginx.conf:ro
    ports:
      - "80:80"
    depends_on:
      - app
    networks:
      - frontend-network

networks:
  backend-network:
  frontend-network:

volumes:
  data:

```

Como se puede observar, las imágenes de Flask/app y MySQL (db), se toman de un Dockerfile local, mientras que las de adminer y Nginx se obtienen de la Internet conectándose a DockerHub, que actúa como una especie de biblioteca para las imágenes Docker.

Normalmente, las imágenes de DockerHub ya suelen estar bien configuradas o hay varias versiones para cada tipo de proyecto, pero en este caso hemos necesitado un poco más de personalización para Flask y MySQL.

Los Dockerfile son archivos que construyen las imágenes con todas las dependencias deseadas. En el caso de Flask, el archivo de configuración ha sido el siguiente:

```
FROM python:3.6

# Instala gunicorn
RUN pip install gunicorn

EXPOSE 5000
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
COPY app.py /app

# Asegúrate de copiar el archivo de configuración de Gunicorn
COPY gunicorn_config.py /app

CMD gunicorn -c /app/gunicorn_config.py app:app
```

De este archivo hay que mencionar el comando:

```
RUN pip install -r requirements.txt
```

Dentro de requirements.txt habrá que incluir todas las librerías que se vayan a utilizar en python, principalmente Flask. En este caso, se han utilizado las siguientes y por ello se han incluido:

```
Flask
mysql-connector
requests
pandas
deep_translator
flask_mail
```

**Listing 6.1:** requirements.txt

Se mencionará brevemente para que se ha utilizado cada dependencia:

- **Flask:** La dependencia básica para construir la web.
- **mysql-connector:** Es una dependencia de Python que permite establecer conexiones con la base de datos según la configuración para de esta forma hacer consultas.
- **requests:** Una dependencia que ayuda a la conexión entre las plantillas de HTML y el código Python para comunicarse y compartir los tipos de solicitudes junto a la información enviada en el formulario.
- **pandas:** Mediante pandas se pueden crear fácilmente estructuras de datos dataframes, que permiten enviar el contenido a las plantillas, por ejemplo, los nutrientes o las recetas.
- **deep\_translator:** Como se ha comentado en el estado del arte, la API de Edamam contiene toda su información en inglés, por ello se utiliza este traductor para poder hacer el cambio inglés español y alrevés.
- **flask\_mail:** Permite mandar correos con python y se ha utilizado para las verificaciones, recuperaciones de cuenta y enviar el correo de bienvenida. Es necesario tener un *email*, se ha creado *veganmaniiaa@gmail.com* para la ocasión.

Por otro lado el Dockerfile usado para la configuración del docker-compose para la base de datos es la siguiente:

```
FROM mysql:5.7
ENV MYSQL_ROOT_PASSWORD rootasdeg2324
ENV MYSQL_DATABASE usuarios
ADD init.sql /docker-entrypoint-initdb.d
```

De igual modo, hay que hacer un inciso en `init.sql` que será el archivo donde la base de datos se inicializará.

Por otro lado, también hay que mencionar que para que la comunicación entre Nginx y Flask sea efectiva se han utilizado los siguientes ficheros de configuración.

```
events {
    worker_connections 4096; # Aumentar el número de
    conexiones por trabajador
}

http {
    server {
        listen 80;

        location / {
            proxy_pass http://app:5000;
            proxy_read_timeout 120s; # Ajustar el tiempo de
            espera en el proxy
        }
    }
}
```

**Listing 6.2:** nginx.conf

```
import multiprocessing

workers = multiprocessing.cpu_count() * 2 + 1 # Ajusta esto
según tu entorno
timeout = 120
bind = "0.0.0.0:5000"
```

**Listing 6.3:** gunicorn\_config.py

### 6.1.1.2. Despliegue de la Aplicación

Una vez cumplidos los requisitos de software se puede desplegar la aplicación en local. Será necesario lanzar el siguiente comando desde la terminal situándose en la ruta principal de la aplicación.

```
docker compose up
```

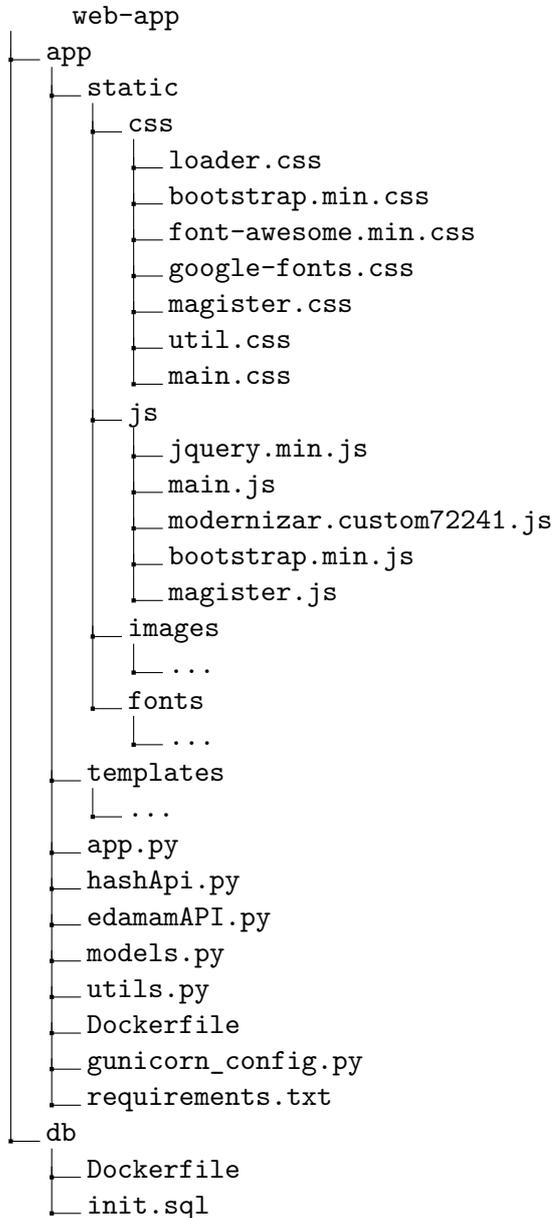
**Listing 6.4:** comando de Despliegue

## 6.2. Implementación

En esta sección se va a mencionar como se ha implementado la web, desde la estructura de las carpetas hasta como se ha realizado el código.

## 6.2.1. Estructura de las Carpetas

En esta sección, se va mostrar la estructura para la disposición de las carpetas y archivos, donde cada carpeta corresponde a un componente de la aplicación web. Esta organización se diseñó con el propósito de simplificar el desarrollo, el mantenimiento y la capacidad y escalabilidad del proyecto, para de esta forma trabajar de forma organizada y eficaz:



Dentro de la carpeta templates hay bastantes archivos que cada uno corresponde a una determinada parte de la web y se muestran a continuación:

```

    templates
├── mails
│   ├── reset_password.html
│   ├── verificarMail.html
│   └── welcomeMail.html
├── añadirAlimento.html
├── añadirComida.html
├── articulosDeTemporada.html
├── base.html
├── blog.html
├── buenHome.html
├── consultarObjetivos.html
├── index.html
├── infoComida.html
├── inicioUsu.html
├── loading.html
├── login.html
├── misComidas.html
├── modificarUsuario.html
├── productosDeTemporada.html
├── recetas.html
├── recuperarContrasena.html
├── registro.html
├── reset_password.html
├── sobreNosotros.html
├── tutorial.html
├── verAlimento.html
├── verComida.html
└── verificacion_correcta.html

```

- **/app/**: Carpeta raíz de la aplicación, en esta carpeta se guarda el archivo de configuración del contenedor y los ficheros que controlan la aplicación.
  - **/static/**: En esta carpeta se encuentran los archivos estáticos, como hojas de estilo *CSS*, archivos *JavaScript* y las imágenes utilizadas en la aplicación.
  - **/templates/**: Carpeta con los diseños de interfaces de usuario.
  - **app.py**: Es el archivo principal donde se encuentra la aplicación *Flask* y las rutas que manejan las solicitudes del usuario.
  - **models.py**: En este archivo se realizan las consultas requeridas por *app.py* a la base de datos.
  - **utils.py**: Contiene todos los métodos que no utilizan rutas para las interfaces, es decir, los métodos adicionales.
  - **requirements.txt**: Archivo que contiene todos los módulos de *Python* necesarios con la versión correspondiente para ejecutar la aplicación.
  - **gunicorn\_config.py**: Configuración WSGI Gunicorn para desplegar la web mediante este servidor HTTP.
  - **Dockerfile**: Fichero de configuración del contenedor en que se va a ejecutar la aplicación *Flask*.
- **/nginx/**: Carpeta con los ficheros de configuración para el proxy *nginx*.
  - **nginx.conf**: Fichero de configuración para *nginx*.

- **/db/**: Carpeta con los ficheros que inicializan la base de datos. Esta carpeta contiene 2 ficheros:
  - **init.sql**: Fichero de configuración de la base de datos con datos de testeo.
  - **Dockerfile**: Fichero de configuración del contenedor en que se va a ejecutar la aplicación MySQL.
- **docker-compose.yml**: Fichero de configuración de los contenedores *Docker*.

### 6.2.2. Estructura Lógica

Gracias a Flask y al motor de plantillas Jinja2 el seguimiento de la estructura MVC se hace más fácil. Se mencionará en el siguiente apartado de que funciones se encarga cada parte pero la estructura y el manejo de solicitudes es el siguiente:

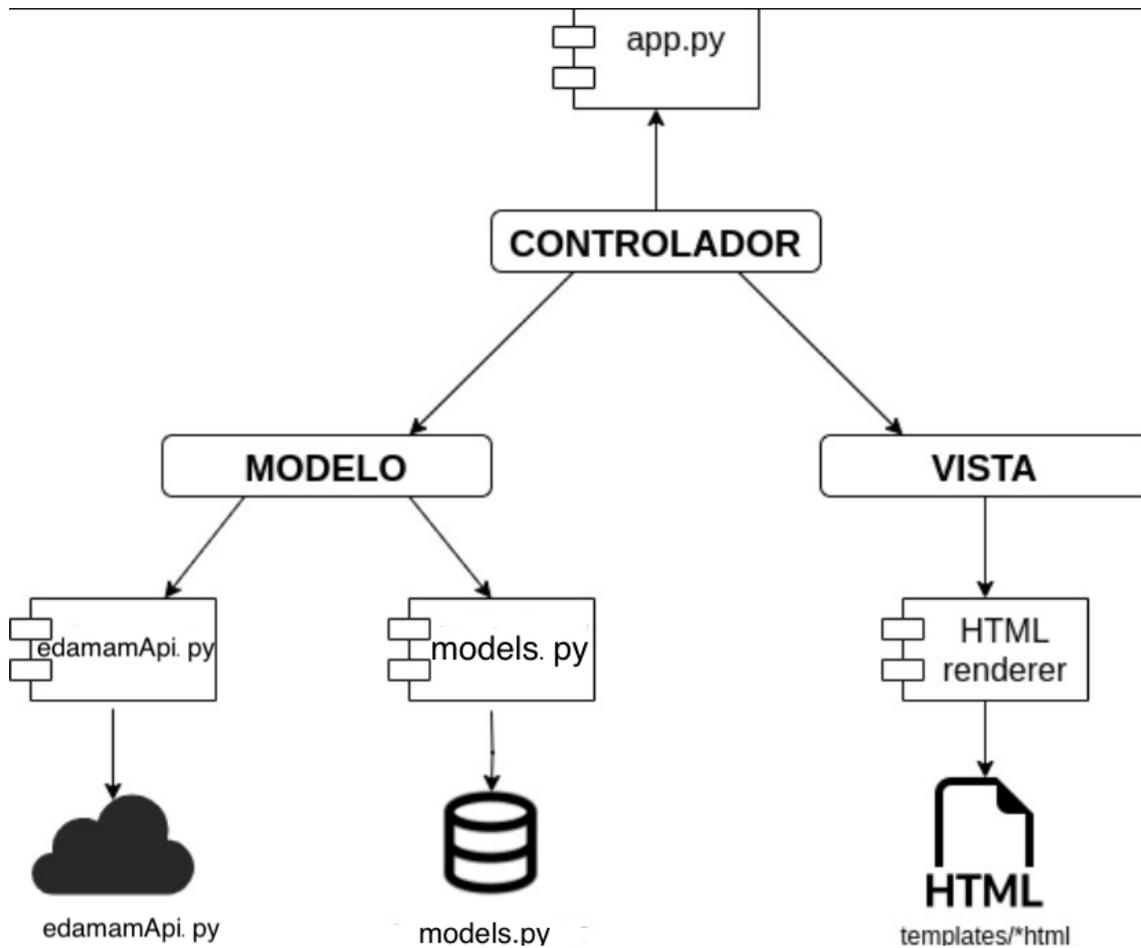


Figura 32: Lógica de Funcionamiento de la Aplicación

### 6.2.3. Código del Backend

En esta parte, se describe como funciona la lógica de la aplicación que hace que funcione correctamente y articula la web junto a algún ejemplo.

#### 6.2.3.1. Controlador

El controlador se encarga de gestionar la lógica de negocio y las interacciones del usuario, controla el flujo de la aplicación. Son las rutas y funciones dentro del código,

reciben solicitudes, procesan la lógica de negocio (incluyendo el acceso a la base de datos) y finalmente devuelven una respuesta adecuada, ya sea una vista o una redirección.

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    error_message = None
    if request.method == 'POST':
        email = request.form['email']
        contrasena = request.form['contrasena']

        try:
            db = Database()
            user_model = UserModel(db)
            hashed_password = hashlib.sha256(contrasena.encode('utf-8')).
                hexdigest()
            user = user_model.get_user_by_email_and_password(email,
                hashed_password)

            if user and user[1] == hashed_password:
                session['email'] = email

                if user[2] == 0:
                    flash('Por favor, verifica tu correo electrónico
                        antes de iniciar sesión.')
                    return redirect(url_for('login', verificacion='
                        pendiente'))
                else:
                    return redirect(url_for('inicioUsu'))
            else:
                error_message = "Las credenciales son incorrectas. Por
                    favor, inténtalo de nuevo."

        except Exception as e:
            error_message = "Se ha producido un error. Por favor,
                inténtalo de nuevo."

    return render_template('login.html', verificacion=request.args.get('
        verificacion'), error_message=error_message)
```

**Listing 6.5:** Controlador login

En primer lugar, si el método utilizado es GET simplemente se devuelve y renderiza la plantilla para que el usuario inicie sesión.

Si por el contrario, se accede con POST, del formulario de login, login.html, se recogen los datos introducidos por el usuario, en este caso el email y la contraseña que son únicos en conjunto para cada usuario y son los campos que identifica a un usuario.

Posteriormente, obtiene la información del usuario conectándose a la base de datos mediante el modelo. Si no se encuentra ningún usuario asociado a ese mail y a esa contraseña se imprime un mensaje comentando que las credenciales son incorrectas.

Si las credenciales son correctas, se comprueba si el mail está verificado, si no lo estuviese, se le redirigirá al usuario de nuevo al login imprimiendo el mensaje para que por favor, se verifique. De estar verificado, el inicio de sesión se completará y se redirigirá al usuario a la vista inicioUsuario, añadiendo su mail a la session para el resto de plantillas a las que acceda

### 6.2.3.2. Modelo

Por otro lado, el modelo, que se gestiona dentro del controlador, representa la lógica de la aplicación, y los datos fundamentales. Para el caso del login de nuevo nos encontramos con lo siguiente:

```
class Database:
    def __init__(self):
        self.config = {
            'user': 'root',
            'password': 'rootasdeg2324',
            'host': 'db',
            'port': '3306',
            'database': 'usuarios'
        }
        self.connection = None

    def connect(self):
        if not self.connection or not self.connection.is_connected():
            self.connection = mysql.connect(**self.config)
        return self.connection

    def close(self):
        if self.connection and self.connection.is_connected():
            self.connection.close()

class UserModel:
    def __init__(self, db):
        self.db = db

    def get_user_by_email_and_password(self, email, hashed_password):
        connection = self.db.connect()
        cursor = connection.cursor()
        cursor.execute('SELECT id_cliente, contrasena, email_verificado
            FROM Cliente WHERE email = %s AND contrasena = %s', (email,
            hashed_password))
        user = cursor.fetchone()
        cursor.close()
        self.db.close()
        return user
```

**Listing 6.6:** Modelo login

En el controlador, se necesita sacar la información de la base de datos relacionada con el usuario y contraseña introducidos por el usuario. Para ello, se conecta a la base de datos y hace la consulta, donde si encuentra algún usuario se devolverá la información al controlador y sino se devolverá el usuario, pero vacío.

### 6.2.3.3. Vista

La vista se encarga de la presentación y la interfaz de usuarios. En este caso, las vistas están representadas por los archivos HTML que se renderizan en las rutas, cómo inicioUsu.html, login.html o tutorial.html.

### 6.2.4. Código del Frontend

El código frontend se ocupa de la apariencia, la visibilidad y la accesibilidad para los usuarios, definiendo la interfaz gráfica y las interacciones con la aplicación. En

la implementación de los archivos *HTML*, se han empleado diversas herramientas para simplificar el proceso.

En primer lugar, se ha aprovechado *Jinja2*, una herramienta de plantillas altamente versátil comúnmente utilizada en el desarrollo web con *Flask* y otros frameworks de *Python*. *Jinja2* permite insertar variables y expresiones en las plantillas *HTML*, lo que nos permite mostrar información específica del usuario. También proporciona la capacidad de incorporar estructuras de control, como condicionales o bucles, para iterar a través de listas de datos y mostrarlos en la interfaz, como es el caso de las tablas. Un beneficio adicional de *Jinja2* es su inclusión de mecanismos de seguridad diseñados para prevenir ataques de inyección de código, lo que disminuye las vulnerabilidades de la aplicación.

En cuanto a la estética de la interfaz gráfica, se ha utilizado *Bootstrap*. Esta herramienta simplifica el desarrollo frontend y ofrece resultados satisfactorios gracias a la diversidad de componentes y estilos predefinidos que proporciona. No obstante, estos componentes y estilos son flexibles y personalizables según las necesidades del proyecto. *Bootstrap* ha permitido ahorrar tiempo y esfuerzo en la redacción de *CSS* y *JavaScript*. Además, la capacidad de reutilizar componentes y estilos ha sido muy útil para mantener una coherencia en todas las interfaces.

### 6.3. Despliegue Web

Por último, una vez la aplicación se finaliza, se procede a desplegar la aplicación en la nube. Como se menciona en la sub-sección herramientas del apartado Enfoque Inicial, se ha utilizado name.com y DigitalOcean. En primer lugar, el hecho de haber implementando la aplicación con Docker facilita el proceso de migración de la App.

En primer lugar, usando la instancia de DigitalOcean que hemos creado exportamos todo el proyecto VeganMania a la máquina virtual mediante Github o mediante el comando `scp` que envía archivos desde un dispositivo a otro. Para poder conectarnos a la máquina virtual necesitamos crear un par de claves con el comando `$keygen` y añadimos nuestra clave pública a la máquina. Por otro lado, para poder ejecutar el comando de Docker y desplegar la web necesitamos instalar Docker teniendo en cuenta que por defecto la máquina creada en DigitalOcean tiene el sistema operativo de Ubuntu.

Una vez conseguido esto, ya podemos acceder a la aplicación desde otros dispositivos utilizando la IP asociada a la máquina virtual. No obstante, partiendo de que es difícil de recordar y queda poco profesional que los usuarios tengan que acceder a una dirección tal que `http://192.169.0.1` cada vez que quieran acceder a la página, se ha registrado un dominio `veganmania.live`. De igual modo, el hecho de utilizar un dominio (registrado en este caso mediante Name.com) hace que podamos incluir el certificado SSL para establecer conexiones seguras a través de HTTPS.

Finalmente, habiendo realizado esto, podemos acceder a nuestro dominio mediante HTTPS. Lo que va a proporcionar a nuestra web un nombre y seguridad.

# 7. Pruebas

En esta sección se van a redactar todas las pruebas realizadas a lo largo del desarrollo de la aplicación. Estas pruebas son fundamentales para garantizar que la aplicación cumpla con los requisitos y funcionalidades esperadas. También se comprueba que no existan errores, para que el usuario final no encuentre ningún bug y afecte a su experiencia antes de su lanzamiento al público. En adición, las pruebas contribuyen a la mejora continua comprobando la usabilidad y experiencia de aplicación.

## 7.1. Pruebas de Flujo de Trabajo

En primer lugar, como se ha comentado se han utilizado metodologías DevOps para realizar el proyecto, principalmente Github. Por ello, a medida que se van haciendo avances, los cambios se guardan en Github. Esto nos permite tener un sistema de control de versiones en donde si tenemos algún error podemos dar un paso atrás y recuperar un "punto de guardado" anterior.

Además, se ha utilizado la herramienta de Github, Github Actions que permite crear un flujo de trabajo y cada vez que se haga uno de los guardados (push) que se ha comentado antes, automáticamente Github va a comprobar, compilando el código, para ver si lo nuevo es correcto y no hay ningún error al ejecutar el código.

Esto ha sido vital en el proceso, sobre todo por tener varias versiones y poder acceder a cualquiera de ellas fácilmente. Ha sido muy útil para ahorrar tiempo porque cuando surgía un error añadiendo alguna nueva funcionalidad se podía volver a un estado anterior donde sí funcionaba y porque se automatiza mediante este modo el chequeo de que lo principal funcione bien.

Un flujo de trabajo es un proceso automatizado configurable que ejecutará uno o más jobs. Los flujos de trabajo se definen mediante un archivo de YAML que se verifica en tu repositorio y se ejecutará cuando lo active un evento dentro de este o puede activarse manualmente o en una programación definida.

Los flujos de trabajo se definen en el directorio `.github/workflows` de un repositorio y un repositorio puede tener varios flujos de trabajo, cada uno de los cuales puede realizar un conjunto diferente de tareas. Por ejemplo, puedes tener un flujo de trabajo para crear y probar las solicitudes de cambio, otro para desplegar tu aplicación cada vez que se cree un lanzamiento y todavía otro más que agregue una etiqueta para que cuando alguien abra una propuesta nueva. En este caso, se ha creado el siguiente archivo:

```

name: CI/CD Pipeline
on:
  push:
    branches:
      - master
jobs:
  build:
    runs-on: ubuntu-latest
    services:
      db:
        image: postgres:latest
        ports:
          - 5432:5432
        options: >-
          --health-cmd "pg_isready -U postgres"
          --health-interval 10s
          --health-timeout 5s
          --health-retries 5
        env:
          POSTGRES_DB: mydatabase
          POSTGRES_USER: postgres
          POSTGRES_PASSWORD: password
    steps:
      - name: Checkout code
        uses: actions/checkout@v3
      - name: Set up Python
        uses: actions/setup-python@v4
        with:
          python-version: '3.9'
      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r app/requirements.txt
      - name: Wait for DB to be ready
        run: |
          echo "Waiting for PostgreSQL to be ready..."
          sleep 30
      - name: Test Python code
        run: |
          python -m unittest discover app
      - name: Build Docker images
        run: docker-compose -f docker-compose.yml build
      - name: Run Docker containers
        run: docker-compose -f docker-compose.yml up -d
      - name: Wait for services to be ready
        run: sleep 30
      - name: Run tests in Docker container
        run: docker-compose -f docker-compose.yml run app python -m
          unittest discover app
      - name: Get DB container logs
        if: failure()
        run: |
          echo "Fetching logs for DB container"
          docker logs $(docker ps -q --filter name=db)

```

**Listing 7.1:** workflow.yml

Como se ha mencionado, mediante este código conseguimos que en cada push se despliegue el docker-compose junto a sus Dockerfile y se prueben todos los archivos .py para comprobar que no haya ningún error. En el caso de que todo esté correcto nos llegará un mensaje al correo y nos aparecerá esta notificación en Github:

Figura 33: Resultado push Workflow

## 7.2. Pruebas de funcionalidad

Por otro lado, un proyecto desarrollado en Flask tiene la opción de ejecutarse en modo de depuración durante la fase de desarrollo. Esta funcionalidad, conocida como *debug mode*, permite verificar las trazas de errores detectados en el código desde la perspectiva del cliente para su depuración. Durante todo el proceso de desarrollo local, se ha utilizado el *debug mode* para identificar y solucionar la mayor cantidad posible de errores dentro de la lógica de la aplicación. Además, se ha realizado un exhaustivo tratamiento de los datos de la base de datos para detectar posibles errores de diseño.

Después de una evaluación final en el estado actual de la aplicación, se presentan a continuación algunos resultados de las pruebas realizadas. Estas pruebas se han llevado a cabo para verificar el correcto funcionamiento y validar los resultados obtenidos en la aplicación.

### 7.2.1. Pruebas para un usuario sin identificar

Pruebas de funcionalidad asociadas a un usuario sin identificar en el sistema.

<b>Prueba</b>	<b>Descripción</b>	<b>Resultado Esperado</b>	<b>Resultado Real</b>
<b>Prueba 1:</b> Alguno de los campos necesarios para el registro está vacío o es incorrecto	El usuario deja campos necesarios para el registro vacíos (nombre, apellido, email, etc...)	El sistema avisa que no ha sido posible registrarse en el sistema y se invalida el registro.	Igual a la salida esperada.
<b>Prueba 2:</b> El usuario al registrarse utiliza un correo electrónico que ya está en uso.	Correo electrónico ya registrado en la base de datos	El sistema avisa de que ya existe el correo electrónico en la base de datos del sistema.	Igual a la salida esperada.
<b>Prueba 3:</b> Se introduce una contraseña que no cumpla los requisitos mínimos.	El usuario introduce en el campo una contraseña que no tenga los requerimientos mínimos.	Mensaje del sistema indicando la contraseña ha cumplir los requisitos.	Igual a la salida esperada.
<b>Prueba 4:</b> El usuario rellena el formulario de registro con todos los campos obligatorios escritos correctamente.	Campos obligatorios correctamente escritos	Mensaje indicando se que ha completado el registro correctamente.	Igual a la salida esperada.
<b>Prueba 5:</b> El usuario trata de iniciar sesión sin que haya validado la cuenta.	Cuenta no validada.	El sistema informa al usuario que debe validar la cuenta con el email enviado.	Igual a la salida esperada.
<b>Prueba 6:</b> El usuario introduce el campo vacío para recuperar contraseña.	Campos vacíos.	El sistema informa al usuario que se deben rellenar todos los campos.	Igual a la salida esperada.
<b>Prueba 7:</b> El usuario al iniciar sesión utiliza credenciales incorrectas.	Datos de inicio de sesión incorrectos.	El sistema informa al usuario que las credenciales son incorrectas.	Igual a la salida esperada.
<b>Prueba 8:</b> El usuario intenta iniciar sesión con alguno o ambos campos vacíos.	Campos de inicio de sesión vacíos.	El sistema informa al usuario que 1 o más campos están vacíos.	Igual a la salida esperada.
<b>Prueba 9:</b> El usuario rellena el formulario de inicio de sesión con unos datos correctos.	Campos obligatorios correctamente escritos.	Se redirige al usuario a la pantalla de inicio correspondiente.	Igual a la salida esperada.

**Tabla 38:** Pruebas para un usuario sin identificar

### 7.2.2. Pruebas para un usuario identificado

Pruebas de funcionalidad asociadas a un usuario identificado en el sistema.

<b>Prueba</b>	<b>Descripción</b>	<b>Resultado Esperado</b>	<b>Resultado Real</b>
<b>Prueba 1:</b> El usuario trata de añadir una comida vacía a sus comidas	El usuario deja campos necesarios para el registro de comida vacíos (nombre, tipo, descripción, etc...)	El sistema avisa que se han de rellenar todos los campos.	Igual a la salida esperada.
<b>Prueba 2:</b> El usuario trata de añadir un alimento vacío a sus alimentos	El usuario deja campos necesarios para el registro de alimento vacíos (nombre, comida, cantidad, etc...)	El sistema avisa que se han de rellenar todos los campos.	Igual a la salida esperada.
<b>Prueba 3:</b> El usuario introduce un alimento que no existe en la base de datos.	Alimento no encontrado en la base de datos.	Se le avisa al usuario de que el alimento no se ha encontrado en la base de datos.	Igual a la salida esperada.
<b>Prueba 4:</b> El usuario introduce un alimento correctamente.	Alimento correcto y existente introducido por el usuario	Mensaje indicando se que ha añadido el alimento a la comida del usuario.	Igual a la salida esperada.
<b>Prueba 5:</b> El usuario introduce una comida correctamente.	Comida correcto introducida por el usuario	Mensaje indicando se que ha añadido la comida al seguimiento del usuario.	Igual a la salida esperada.
<b>Prueba 6:</b> El usuario pulsa el botón de borrar alimento.	Solicitud para borrar alimento	Mensaje indicando que el alimento se ha eliminado correctamente.	Igual a la salida
<b>Prueba 7:</b> El usuario pulsa el botón de borrar comida .	Solicitud para borrar comida	Mensaje indicando que la comida se ha eliminado correctamente.	Igual a la salida esperada.
<b>Prueba 8:</b> El usuario pulsa el botón de eliminar cuenta.	Solicitud para eliminar cuenta	Notificación para comprobar si el usuario está seguro y si lo pulsa se borra correctamente.	Igual a la salida.
<b>Prueba 9:</b> El usuario pulsa el botón de cerrar sesión.	Solicitud para cerrar sesión	Se cierra la sesión del usuario y se regresa al index.	Igual a la salida.
<b>Prueba 10:</b> Campo buscar receta vacío.	El usuario busca una receta dejando el campo vacío	Se muestran todas las recetas de las que se disponen.	Igual a la salida.

**Tabla 39:** Pruebas para un usuario identificado PT.1

<b>Prueba</b>	<b>Descripción</b>	<b>Resultado Esperado</b>	<b>Resultado Real</b>
<b>Prueba 11:</b> Se busca una receta concreta.	El usuario busca una receta que dese.	Se muestran todas las recetas cuyo nombre sea similar al introducido.	Igual a la salida.
<b>Prueba 12:</b> Se utilizan los filtros para buscar recetas concretas.	El usuario busca una receta que dese utilizando filtros.	Se muestran todas las recetas que cumplan los filtros.	Igual a la salida.
<b>Prueba 13:</b> Se busca una receta concreta que se sepa que no existe.	El usuario busca una receta extraña.	Se manda un mensaje de que no se ha encontrado ninguna receta.	Igual a la salida.
<b>Prueba 14:</b> Campo buscar alimento vacío.	El usuario busca un alimento dejando el campo vacío	Se muestra al información nutricional del alimento introducido	Igual a la salida.
<b>Prueba 15:</b> Se busca un alimento concreto.	El usuario busca un alimento que dese.	Se muestra al información nutricional del alimento introducido.	Igual a la salida.
<b>Prueba 16:</b> Se busca un alimento concreto que se sepa que no existe.	El usuario busca un alimento extraño.	Se manda un mensaje de que no se ha encontrado ningún alimento.	Igual a la salida.
<b>Prueba 17:</b> No se introduce ningún campo y se pulsa el botón de modificar datos.	El usuario no introduce ningún campo y pulsa modificar datos del usuario	El sistema avisa de que se han modificado los datos correctamente pero ningún cambio se realiza en la base de datos.	Igual a la salida esperada.
<b>Prueba 18:</b> El usuario introduce algún campo con datos no permitidos (por ejemplo, fecha de nacimiento posterior al día actual en modificar datos del usuario ...)	El usuario introduce algún campo que no cumple los requisitos	El sistema avisa de que X campo debe ser...	Igual a la salida esperada.
<b>Prueba 19:</b> Se modifican uno o un par de campos de todos los disponibles.	El usuario introduce en los campos que dese modificar y pulsa el botón.	Mensaje que se ha modificado correctamente al usuario y solo en los campos que ha introducido.	Igual a la salida esperada.
<b>Prueba 20:</b> Se modifican todos los campos disponibles.	El usuario introduce información para todos los campos disponibles.	Mensaje que se ha modificado correctamente al usuario con todos los campos.	Igual a la salida esperada.

**Tabla 40:** Pruebas para un usuario identificado PT.2

### 7.3. Pruebas con Usuarios Reales

Por último, una vez acabada la aplicación se ha probado la aplicación mediante usuarios. Para ello, se ha desplegado la página ya con el droplet de DigitalOcean y el dominio de Name.com y se ha pedido a varias personas que hagan uso de la aplicación durante un tiempo. Con esto y pidiendo a los usuarios que traten de buscar errores usando la aplicación como la usarían en el día a día es como se encuentran errores difíciles de encontrar para el programador. A fin de cuentas, el programador es quien ha hecho la aplicación y sabe cómo funciona. Además, las opiniones y comentarios de los usuarios brindan una valiosa retroalimentación sobre la experiencia de usuario y posibles áreas de mejora. En definitiva, es la mejor prueba de cara a entrega final para depurar todos los errores.

Por otro lado, mediante esta práctica se ha querido probar también de algún modo la intuitividad de la página. Gracias a la retroalimentación se ha modificado alguna interfaz ligeramente y se ha añadido una sección a modo de tutorial donde se explica como funciona la aplicación.

### 7.4. Pruebas de Seguridad

Este apartado está tan enfocado con errores de código normales sino que más allá de eso se ha pensado en la seguridad. Hoy en día la seguridad en las aplicaciones y para los usuarios de las aplicaciones es fundamental, es poco probable que nuestra web sea objeto de ataque de algún cibercriminal, pero se han realizado unas pruebas generales para comprobar qué áreas se pueden mejorar en el tema de la ciberseguridad.

Por otro lado, a pesar de que en nuestra página no haya datos bancarios que son los que más interés pueden generar para atacar webs, contiene alguna información sensible de cada usuario como altura, genero, peso o fecha de nacimiento y no tiene porque salir a la luz. O incluso puede mirándolo desde una perspectiva más amplia, el compromiso de credenciales como el nombre de usuario y la contraseña hashada también puede suponer un problema para el usuario ya que si no usa buenas prácticas posiblemente use las mismas credenciales para otras aplicaciones.

Para ello, se ha usado la herramienta OWASP ZAP. Se ha descargado en Ubuntu y mediante la extensión de quick-start podemos realizar un escaneo automatizado eligiendo nuestra url y seleccionando el ataque, en este caso se va a utilizar "traditional spider". Que es una prueba general y completa de la aplicación web.

En este caso tras haber realizado el ataque, nos ha marcado las siguientes vulnerabilidades:

#### 1. Cross Site Scripting (XSS)

- **Descripción:** XSS permite a los atacantes inyectar scripts maliciosos en páginas web vistas por otros usuarios.
- **Prevención:** Validar y sanitizar todas las entradas del usuario. Utilizar Content Security Policy (CSP) para limitar la ejecución de scripts.

#### 2. SQL Injection

- **Descripción:** Los atacantes pueden ejecutar comandos SQL arbitrarios en la base de datos de la aplicación.

- **Prevención:** Utilizar consultas preparadas y ORM (Object-Relational Mapping) para manejar las entradas del usuario.
- 3. SQL Injection - MySQL**
    - **Descripción:** Específicamente afecta a bases de datos MySQL permitiendo la manipulación de consultas SQL.
    - **Prevención:** Utilizar consultas preparadas y procedimientos almacenados para evitar la inyección de SQL.
  - 4. Absence of Anti-CSRF Tokens**
    - **Descripción:** La falta de tokens CSRF permite a los atacantes realizar acciones en nombre de otros usuarios sin su consentimiento.
    - **Prevención:** Implementar tokens CSRF en formularios y validarlos en el servidor.
  - 5. Content Security Policy (CSP)**
    - **Descripción:** CSP ayuda a prevenir una variedad de ataques incluyendo XSS al controlar los recursos que una página puede cargar.
    - **Prevención:** Definir una política de seguridad de contenido (CSP) adecuada y actualizarla regularmente.
  - 6. Missing anti-clickjacking Header**
    - **Descripción:** La falta de encabezados anti-clickjacking permite que el contenido de la página sea cargado en un marco (frame) malicioso.
    - **Prevención:** Utilizar los encabezados X-Frame-Options o Content-Security-Policy frame-ancestors.
  - 7. Cookie No HttpOnly Flag**
    - **Descripción:** Las cookies sin la bandera HttpOnly pueden ser accedidas mediante scripts del lado del cliente, aumentando el riesgo de XSS.
    - **Prevención:** Establecer la bandera HttpOnly en las cookies sensibles.
  - 8. Cookie without SameSite Attribute**
    - **Descripción:** La falta del atributo SameSite en las cookies puede aumentar el riesgo de ataques CSRF.
    - **Prevención:** Configurar el atributo SameSite en las cookies para limitar su envío en solicitudes de origen cruzado.
  - 9. Server leaks information via "X-Powered-By" HTTP Response Header Field(s)**
    - **Descripción:** Los encabezados X-Powered-By pueden revelar información sobre la tecnología del servidor, facilitando ataques dirigidos.
    - **Prevención:** Configurar el servidor para eliminar o modificar el encabezado X-Powered-By.
  - 10. Server leaks Version Information via "Server" HTTP Response Header Field(s)**
    - **Descripción:** El encabezado Server puede revelar la versión del software del servidor, exponiendo posibles vulnerabilidades.
    - **Prevención:** Configurar el servidor para ocultar o modificar el encabezado Server.
  - 11. X-Content-Type-Options Header Missing**
    - **Descripción:** La falta del encabezado X-Content-Type-Options permite la interpretación incorrecta del tipo de contenido.

- **Prevención:** Incluir el encabezado X-Content-Type-Options con el valor nosniff.

#### 12. **Loosely Scoped Cookie**

- **Descripción:** Las cookies con un ámbito (scope) demasiado amplio pueden ser enviadas a múltiples subdominios innecesariamente.
- **Prevención:** Definir el ámbito de las cookies de forma precisa y restringida a los dominios necesarios.

#### 13. **Modern Web Application**

- **Descripción:** Las aplicaciones web modernas pueden incluir nuevas tecnologías que presenten riesgos adicionales si no se manejan adecuadamente.
- **Prevención:** Mantenerse actualizado con las mejores prácticas de seguridad y realizar auditorías de seguridad regularmente.

#### 14. **User Controllable HTML Element Attribute**

- **Descripción:** Permitir a los usuarios controlar atributos HTML puede llevar a ataques XSS u otros comportamientos no deseados.
- **Prevención:** Validar y sanitizar todos los atributos HTML controlados por el usuario.

## 8. Conclusiones y Marco Futuro

Para concluir este trabajo, se va a hacer una reflexión sobre los resultados y posibles líneas futuras para dotar a la web de más funcionalidades y seguir desarrollándola. En primer lugar, se revisará el trabajo realizado y se contrastará con los objetivos marcados inicialmente. De esta forma veremos cuanto se ha avanzado en el sentido de ver si hemos hecho más de lo que esperábamos, lo mismo o menos. Además, se considerarán las posibles mejoras que se pueden implementar en la aplicación para un posible desarrollo futuro. Por otro lado, se van a adjuntar los links más relevantes del proyecto para que se pueda tener una idea más clara o si se quiere replicar / mejorar el proyecto.

- **Vídeo de Muestra / Tutorial de la Web:** Para mostrar el funcionamiento lógico y las funcionalidades que ofrece la aplicación, se ha creado un vídeo en YouTube. Se puede acceder mediante el link <sup>1</sup>.
- **Link de la Página Web:** Para acceder a la web se puede acceder mediante el link <sup>2</sup>. Se ha creado un usuario de prueba cuyas credenciales son las siguientes: **email** `davy.elorza@gmail.com` y **contraseña** `1111`.
- **Link al repositorio del proyecto:** Como se ha mencionado el proyecto está disponible en Github <sup>3</sup>. Cualquier persona puede acceder y modificar el código a su antojo.
- **Link a DockerHUB:** Las imágenes utilizadas para los contenedores de la aplicación y la base de datos se han subido para que sean utilizables. Esto permite no hacer uso de un Dockerfile para crear la imagen y a futuro permite la inclusión de Kubernetes. Se puede acceder mediante el link a DockerHub. <sup>4</sup>

### 8.1. Objetivos

Nuestro objetivo inicial era crear una aplicación que incluyese varias herramientas para tratar de mejorar el conocimiento alimentario de la población y dar herramientas para mejorar la alimentación personal mediante el seguimiento de nutrientes, poder observar la fruta y verdura de temporada y una sección que proporcione ideas para recetas, a poder ser con los productos de temporada locales. Por una parte, queríamos que la aplicación estuviese pensada para todo tipo de persona, pero se pensó como clientes objetivo las personas veganas y vegetarianas que tienen más dificultad para llegar a muchos de sus nutrientes objetivo.

---

<sup>1</sup><https://youtu.be/CI4N3eKkBJ8?si=VHLf0wC8aAHeSGie>

<sup>2</sup><http://veganmania.live/>

<sup>3</sup><https://github.com/kepaxabier/flask-mysql-dockercompose>

<sup>4</sup>[https://hub.docker.com/repository/docker/davidelorza/vegan\\_mania/general](https://hub.docker.com/repository/docker/davidelorza/vegan_mania/general)

En líneas generales, los objetivos se han cumplido satisfactoriamente, ya que la web realizada tiene todas las funcionalidades establecidas desde el comienzo. Con un interfaz intuitiva y fácil de usar donde los datos y toda la información se muestran de forma visible.

Por otro lado, en lo que a los objetivos personales respecta, la creación de esta página web ha sido una experiencia enriquecedora tanto a nivel personal como profesional. En primer lugar, he desarrollado habilidades técnicas, por una parte usando herramientas que he conocido a lo largo de la carrera y he puesto en práctica, pero también por nuevas herramientas que desconocía y he aprendido para añadir funcionalidades a la web. En segundo lugar, el desarrollo de contenido para la sección de recetas, productos de temporada y artículos educativos, me ha permitido profundizar en el campo de la nutrición. De esta forma, tengo más noción sobre cómo llevar una alimentación equilibrada e inconscientemente lo he incorporado a mi día a día.

## 8.2. Reflexión Personal

Haber finalizado este proyecto con satisfacción me ha hecho experimentar una profunda sensación de auto-realización. A lo largo del proyecto, he tenido la oportunidad de materializar una idea que aborda una necesidad específica dentro de una comunidad y además ofrece recursos valiosos para cualquier persona interesada en mejorar su alimentación y conocimiento nutricional. En cierto modo era lo que me interesaba, crear una aplicación práctica y que con suerte sea utilizada y de ayuda para la gente.

Por una parte, este proyecto me ha dado la oportunidad de poner un broche final a mi carrera enfrentándome a un proyecto grande y solo. De esta forma, no solo he conseguido pulir y afianzar conocimientos aprendidos durante la carrera sino que he aprendido nuevas tecnologías enriqueciendo mi comprensión y enfoque del proceso de desarrollo.

En mi experiencia profesional, es el proyecto más grande que he hecho hasta la fecha y de manera individual. Me ha permitido conocerme mejor ante situaciones en las que han sido demandantes u otras de bloqueo mental en las que no conseguía avanzar tanto como quería. Sobre todo en el ecuador del proyecto, donde sabía que podía realizar el proyecto pero dudando de realizar aquella aplicación que fuese de utilidad para las personas y que tuviese valores añadidos frente a otras existentes en el mercado. Va que como he comentado mi objetivo es que no quede en el olvido sino que pueda ayudar a la gente que necesite.

Finalmente, mediante una semana alejado del proyecto y habiendo realizado algún curso sobre cómo mejorar el UX/UI [14] de la web llegó una ráfaga de inspiración y conseguí empezar a avanzar a pasos agigantados logrando que la aplicación empezase a quedar como tenía en mente.

Por último, me gustaría finalizar este trabajo que ha cerrado una etapa en mi vida profesional pero también ha abierto otra:

*“El éxito es la suma de pequeños esfuerzos repetidos día tras día.”  
– Robert Collier*

Esta frase define a la perfección la esencia de comenzar y terminar una etapa, como ha sido en este caso VeganMania.

## 9. Trabajo Futuro

A pesar de que se hayan cumplido los objetivos siempre existe margen de mejora y sobretodo en un área tecnológica como esta página web. A continuación se mencionarán alguna de las mejoras posibles para la página aunque realmente se puedan encontrar muchas por el motivo que se ha mencionado.

### 9.1. Base de Datos

En primer lugar, se ha observado que la base de datos que utilizamos, (mediante API) Edamam, no es la mejor opción. En primer lugar, no es la forma más eficiente ya que al venir la información en inglés y tener que hacer una traducción hay un lapso de tiempo hasta que carga la información de los alimentos. Por otro lado, a pesar de que en su web se puedan hacer consultas de todo tipo incluyendo productos locales, la API abierta no tiene marcas conocidas locales y el no tener los alimentos om-premise, nos dificulta a la hora de filtrar los alimentos. Sin embargo, como ya se ha mencionado en el apartado 3 de las bases de datos posibles es la mejor opción que hemos encontrado ya que el resto no estaban disponibles o eran del mismo tipo.

Una posible solución que se ha encontrado para solventar el problema es que con el paso del tiempo, los usuarios puedan incluir los productos que no dispongamos. De esta forma, dispondríamos de una base de datos actualizada constantemente. Aunque esto puede suponer también el problema de una base de datos más defectuosa si los usuarios incluyen cosas que no son correctas.

### 9.2. Tecnologías

Aunque se haya tratado de usar las tecnologías más adecuadas y novedosas para el proyecto, recientemente se ha puesto de moda el concepto de SPA (aplicación de una sola página). Esto implica desarrollar una web que cargue tan solo una vez el HTML y utilice JS para actualizar el contenido de la página sin necesidad de recargarla completamente y dinamizarla. De esta forma el usuario puede conseguir una experiencia más fluida y rápida y se puede obtener mediante algún framework como React, Vue.js o Angular.

Además, a pesar de que MySQL y Docker fuesen unas opciones más que razonables para este proyecto, tal vez, el microframework de Flask se podría haber sustituido por otras opciones que no se habían tenido en cuenta al inicio del proyecto, como Node o Spring Java. Ya que a pesar de que Flask ofrezca rapidez, ligereza y flexibilidad, Node actualmente tiene más usuarios por lo que está más actualizado.

### **9.3. App Móvil**

Hoy en día, que los móviles son prácticamente una extensión de nuestro brazo y a pesar de que la página web sea accesible para todos los dispositivos mediante el navegador, no hay nada como una aplicación para este tipo de herramientas. Por ello ofrecer también una aplicación móvil puede ser un objetivo de futuro.

### **9.4. Soporte Multilingüe**

Otro objetivo a futuro puede ser ampliar el soporte para múltiples idiomas, permitiendo a usuarios de diferentes regiones acceder a la web en su lengua materna. En este caso, las mayores prioridades serían euskera e inglés.

### **9.5. Recomendaciones Personalizadas**

Actualmente, no se ofrecen recomendaciones personalizadas. Mediante el uso de un modelo IA se podría implementar un sistema de recomendaciones basado en las preferencias y datos de seguimiento nutricional del usuario con el objetivo de completar los nutrientes en los que flaquea, con comida saludable y a poder ser en conjunción con sus preferencias de gusto. . Haciendo uso de unas técnicas de aprendizaje automático para que a cada día se vuelvan más precisas.

### **9.6. Login mediante Google/Github/Facebook...**

Por otro lado, a pesar de que nuestra aplicación incluya autenticación de cuenta mediante mail, hoy en día la mayoría de aplicaciones ofrecen servicios para iniciar sesión con aplicaciones de terceros como Google, Github o Facebook siendo Google la más extendida.

### **9.7. Blog Interactivo**

Para permitir que el Blog de igual modo se amplíe con el tiempo, se puede hacer un Blog donde haya interacción entre los usuarios. Donde cada uno pueda escribir sus experiencias y consejos y que se puedan comentar y dar like. Esto es una opción bastante interesante ya que se pondría como objetivo conseguir una comunidad.

# Bibliografía

- [1] Statista, *Número de casos de obesidad registrados en España de 2011 a 2022*. Statista, 2024. <https://es.statista.com/estadisticas/1046199/numero-de-casos-de-obesidad-en-espana/>
- [2] Ministerio de Sanidad, Consumo y Bienestar Social, *Fuente de datos de Número de casos de obesidad registrados en España de 2011 a 2022*. 19-04-2024. <https://pestadistico.inteligenciadegestion.sanidad.gob.es/publicoSNS/S/base-de-datos-de-clinicos-de-atencion-primaria-bdcap/>
- [3] The Lancet Journal *Trends in adult body-mass index in 200 countries from 1975 to 2014: a pooled analysis of 1698 population-based measurement studies with 19.2 million participants* The Lancet Journal, 02-04-2016. [https://www.thelancet.com/journals/lancet/article/PIIS0140-6736\(16\)30054-X/fulltext](https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(16)30054-X/fulltext)
- [4] La Vanguardia *En el mundo, ya hay más obesos que flacos* La Vanguardia, 02-04-2016. <https://www.lavanguardia.com/vangdata/20160403/40847295800/obesidad-grafico-mapa-mundo-raking-paises.html>
- [5] World Obesity Federation 2023. *World Obesity Atlas 2023* World Obesity Federation 2023., 03-2023. [https://s3-eu-west-1.amazonaws.com/wof-files/World\\_Obesity\\_Atlas\\_2023\\_Report.pdf](https://s3-eu-west-1.amazonaws.com/wof-files/World_Obesity_Atlas_2023_Report.pdf)
- [6] Medium *Acceso a datos nutricionales y búsqueda de recetas mediante el uso de las APIs de Edamam* Samuel Cortez 2021. <https://medium.com/@oscars.cortezmo/acceso-a-datos-nutricionales-y-b%C3%BAscueda-de-recetas-mediante-el-uso-de-las-apis-de-edamam-717ef59ae527>
- [7] Edamam *Leading provider of nutrition data and analytics* <https://developer.edamam.com/edamam-docs-recipe-api>
- [8] DigitalOcean *Cómo instalar y usar Docker en Ubuntu 20.04* Brian Hogan, 2020 <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04-es>
- [9] Docker *The #1 containerization software for developers and teams* <https://www.docker.com/products/docker-desktop/>
- [10] Union Vegetariana Española *El Veganismo y su contribución a los Objetivos de Desarrollo Sostenible* <https://unionvegetariana.org/el-veganismo-y-su-contribucion-a-los-objetivos-de-desarrollo-sostenible/>
- [11] Digital Ocean *How To Install an SSL Certificate from a Commercial Certificate Authority* <https://www.digitalocean.com/community/tutorials/how-to-install-an-ssl-certificate-from-a-commercial-certificate-authority>
- [12] Tarifa Luz Hora *¿Cuál es el precio kWh de luz y gas natural de Iberdrola?* <https://tarifaluzhora.es/companias/iberdrola/precio-kwh>
- [13] National Academies Press *National Academies Press* [www.nap.edu](http://www.nap.edu)

[14] Udemy *Improve Your Web Design Experience with Squarespace* <https://www.udemy.com/course/improve-your-web-design-experience-with-squarespace/>