

# DESARROLLO DE UNA RED SOCIAL PARA AMANTES DE LAS RECETAS DE COCINA



TRABAJO FIN DE GRADO  
CURSO 2024-2025

AUTOR

ELENA CANTALEJO SÁNCHEZ: 8,5  
VICENTE GARRIDO JUÁREZ: 9,5  
PABLO GONZÁLEZ CORREDOR: 8,5

DIRECTOR

ANTONIO SARASA CABEZUELO

DOBLE GRADO EN:

ADMINISTRACIÓN DE EMPRESAS E INGENIERÍA INFORMÁTICA  
FACULTAD DE INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID

30 DE ABRIL DE 2025

TITLE  
**DEVELOPMENT OF A SOCIAL NETWORK FOR LOVERS OF  
COOKING RECIPES**

FINAL DEGREE PROJECT IN COMPUTER  
ENGINEERING

AUTHOR  
ELENA CANTALEJO SÁNCHEZ  
VICENTE GARRIDO JUÁREZ  
PABLO GONZÁLEZ CORREDOR

DIRECTOR  
ANTONIO SARASA CABEZUELO

**SUBMISSION PERIOD: JUNE 2024**

DOUBLE DEGREE IN GRADO EN INGENIERÍA INFORMÁTICA  
FACULTY OF COMPUTER SCIENCE  
UNIVERSIDAD COMPLUTENSE DE MADRID

30TH APRIL 2025

## DEDICATORIA

A La Universidad Complutense de Madrid,  
por brindarnos la oportunidad  
de crecer como ingenieros  
y como personas

## **AGRADECIMIENTOS**

A nuestro tutor, Antonio, por su dedicación a nuestra guía durante todo este año, y por su paciencia y profesionalidad durante el transcurso del curso.

# Resumen

## Red social para amantes de las recetas de cocina

Con el proyecto de la red social para amantes de las recetas se ha querido desarrollar una aplicación web de recetas que promueva la interacción social entre los usuarios y facilite la gestión de recetas y menús personalizados.

Se focaliza en una buena gestión de usuarios, permitiendo el registro, inicio y cierre de sesión, así como la modificación y eliminación de cuentas con medidas de seguridad robustas. Además, se implementan funciones para la creación, modificación y organización de recetas, y la posibilidad de crear listas personalizadas para agruparlas. Posteriormente, se tiene en cuenta la interacción social, incorporando funciones como la creación de menús personalizados, la opción de agregar recetas a listas de favoritos y comentar o puntuar las recetas de otros usuarios. Con esta aplicación, se da la posibilidad a que los usuarios puedan seguirse entre sí, interactuar en foros públicos y compartir recomendaciones de recetas.

Con este enfoque iterativo, lo que se busca no es solo la gestión de recetas, sino también la creación de una comunidad interactiva donde los usuarios puedan compartir y disfrutar de experiencias gastronómicas.

## Palabras clave

Recetas, usuarios, gestión, interacción, social, menús, listas, comunidad, seguridad, recomendaciones

# Abstract

The development of a social network for lovers of cooking recipes was developed with the objective of creating a web application for recipe management that fosters social interaction among users while facilitating the organization of recipes and personalized menus.

The project placed particular emphasis on efficient user management, enabling registration, login, and logout, as well as account modification and deletion, all implemented with robust security measures. Additionally, functionalities were incorporated to allow the creation, modification, and organization of recipes, along with the possibility of generating personalized lists for classification purposes.

Subsequently, social interaction features were introduced, including the ability to create personalized menus, add recipes to favorite lists, and provide ratings and comments on other users' recipes. Furthermore, the application allows users to follow one another, participate in public forums, and exchange recipe recommendations.

Through this iterative approach, the network aims not only to facilitate recipe management but also to foster the development of an interactive community where users can share and engage in gastronomic experiences.

## Keywords

Recipes, users, management, iteration, social, menu, lists, community, security, recommendations

# ÍNDICE

1.	Introducción .....	11
1.1	Motivación .....	11
1.2	Objetivos.....	12
1.3	Plan de trabajo.....	12
2.	Introduction .....	15
2.1	Motivation .....	15
2.3	Goals.....	16
2.4	Work plan .....	16
3.	Estado de la cuestión.....	19
3.1	Cookpad.....	19
3.2	Tasty .....	19
3.3	Yummly .....	20
3.4	Diferenciación de la red social para amantes de las recetas .....	20
4.	Tecnologías empleadas .....	21
4.1	Software.....	21
4.2	Lenguajes de programación empleados.....	22
5.	Arquitectura de la aplicación.....	25
5.1	Interacción entre el frontend y el backend .....	25
5.2	Patrones de diseño .....	26
5.3	Gestión de la base de datos .....	27
6.	Implementación de la plataforma.....	35
6.1	Gestión de usuarios.....	35
6.1.1	Registro y eliminación de usuario.....	35
6.1.2	Inicio y cierre de sesión.....	38
6.1.3	Modificación de perfil .....	39
6.2	Gestión de recetas .....	40
6.2.1	Creación, modificación y eliminación de receta .....	41

6.2.2	Búsqueda de receta.....	42
6.3	Gestión de listas.....	44
6.3.1	Creación de lista y visualización .....	44
6.3.2	Adición y exclusión de la lista .....	44
6.4	Gestión de interacciones sociales .....	45
6.4.1	Creación, edición y eliminación de comentarios y mensajes en el foro .....	46
6.4.2	Seguimiento entre usuarios .....	48
6.5	Gestión de menú.....	49
6.6	Administración .....	49
7.	Conclusiones y trabajo futuro .....	50
8.	Conclusions and future work .....	52
9.	Contribuciones Personales .....	53
10.	Bibliografía .....	56
1.	Actores .....	57
1.1.	Usuario registrado.....	57
2.	Especificación de requisitos .....	57
2.1.	Gestión de usuario .....	57
2.2.	Gestión de recetas .....	62
2.2.1.	FUNCIONALIDADES BÁSICAS .....	62
2.2.2.	INTERACCIÓN CON RECETAS AJENAS .....	67
2.2.3.	BÚSQUEDA Y VISUALIZACIÓN DE RECETAS .....	70
2.2.4.	COMENTARIOS EN RECETAS AJENAS.....	72
2.2.5.	COMENTARIOS EN RECETAS PROPIAS .....	75
2.3.	LISTAS.....	76
2.3.1.	OPERACIONES CON LISTAS .....	77
2.4.	MENÚS.....	80
2.4.1.	INTERACCIÓN CON MENÚS.....	82

2.5.	FOROS .....	87
2.5.1.	INTERACCIÓN CON FOROS .....	88
2.6.	SOCIAL .....	91
2.6.1.	ACTUALIZACIONES EN EL SEGUIMIENTO .....	92
2.7.	RECOMENDACIONES .....	93
2.8.	Administración .....	94
2.8.1.	INTERACCIÓN CON USUARIOS.....	94
2.8.2.	ELIMINACIÓN DE CONTENIDO INAPROPIADO .....	96
11.	Anexo B: Manual de usuario .....	101

## TABLA DE ILUSTRACIONES

Ilustración 1. Diagrama de Gantt .....	13
Ilustración 2. Diagrama entidad-relación .....	28
Ilustración 3. Registro .....	36
Ilustración 4. Notificación.....	37
Ilustración 5. Eliminación de cuenta .....	38
Ilustración 6. Validación de campos .....	39
Ilustración 7. Validación y actualización.....	40
Ilustración 8. Archivo create_recipe.php .....	41
Ilustración 9. Búsqueda y filtrado de recetas.....	43
Ilustración 10. Añadir receta.....	45
Ilustración 11. Edición de comentarios.....	47
Ilustración 12. Funcionalidad de seguimiento. ....	48
Ilustración 13. Creación de receta (1).....	101
<i>Ilustración 14. Creación de receta (2)</i> .....	102
Ilustración 15. Creación de receta (3).....	103
Ilustración 16. Nueva receta creada.....	103
Ilustración 17. Acceso por otro usuario.....	104
Ilustración 18. Comentario de otro usuario. ....	105
Ilustración 19. Edición o eliminación de comentario. ....	105
Ilustración 20. Entrada en el foro.....	106
Ilustración 21. Nuevo mensaje en el foro.....	106
Ilustración 22. Mensaje disponible en el foro. ....	107
Ilustración 23. interacción en el foro.....	107
Ilustración 24. Creación de un hilo.....	108
Ilustración 25. Añadir receta existente a lista personalizada (1) .....	108
Ilustración 26. Añadir receta existente a lista de favoritos (2). Selección de lista. ....	109
Ilustración 27. Añadir receta existente a lista de favoritos (3). Confirmación de lista...	109
Ilustración 28. Comprobación (1). ....	110
Ilustración 29. Comprobación (2) .....	110
Ilustración 30. Comprobación 3. ....	111

Ilustración 31. Acceso a "Menús".....	111
Ilustración 32. Crear menú .....	112
Ilustración 33. Proceso de creación de menú. cumplimentación de campos.....	113
Ilustración 34. Inserción en listado principal. ....	113
Ilustración 35. contenido del menú. ....	114

# 1. Introducción

Hoy en día, los libros de recetas y programas de cocina han pasado a un segundo plano, debido, en gran parte al auge que ha experimentado el sector tecnológico en la última década. Y con la normalización de dispositivos con acceso a internet en el grueso de la población de nuestro país, la audiencia ha migrado hacia medios de comunicación más accesibles e inmediatos.

La red social para los amantes de las recetas nace bajo el pretexto de acercar a usuarios en la necesidad de encontrar a personas de perfil similar, igual de interesadas por compartir su pasión en un medio que permita ampliar los horizontes culinarios, crear nuevas relaciones, explorar y aprender.

En definitiva, el enfoque principal de este proyecto no es, sino el de aprovechar las mejores facetas de las redes sociales, así como las funcionalidades de seguimiento, compartición de contenido, y foros aptos para la discusión. A fin de adaptarlas a un tema que, si bien ha sido explotado, no ha llegado a calar lo suficiente en la audiencia como para darle una segunda oportunidad.

## 1.1 Motivación

Siempre se ha visto la cocina como algo más que una tarea diaria, se ha querido tratar de entenderla como un lugar de creatividad y desarrollo personal. Compartir la receta, mejorarla gracias a alguien más o probar algo nuevo. Una comida une a las personas, ya sea en la mesa o en la pantalla. Con todo esto en mente, se opta por desarrollar esta aplicación web de recetas que no se trata sólo de administrar platos y menús.

## 1.2 Objetivos

La aplicación web, tiene como meta principal facilitar a los usuarios el acceso a una plataforma de temática culinaria que permita la interacción entre ellos, así como la creación de contenido relacionado con esta materia.

- Estudio de la competencia: Análisis exhaustivo de la situación actual del mercado e identificación de "gaps".
- Selección de las bases del diseño visual, incluyendo paleta de colores, distribución en pantalla, logotipo y botones.
- Selección del ámbito tecnológico: Consideración de los lenguajes de programación aprendidos durante los cinco años de carrera, seleccionando los más propicios para la tarea.
- Desarrollo de la base de datos: Selección del lenguaje de programación de la base de datos, del diagrama de ésta, así como de las relaciones a las que está sujeta.
- Implementación de las funcionalidades: Convenio acerca de las partes clave del proyecto y la mejor manera de implementarlas en la aplicación web. Posterior desarrollo, cohesión y corrección de todas las aportaciones.

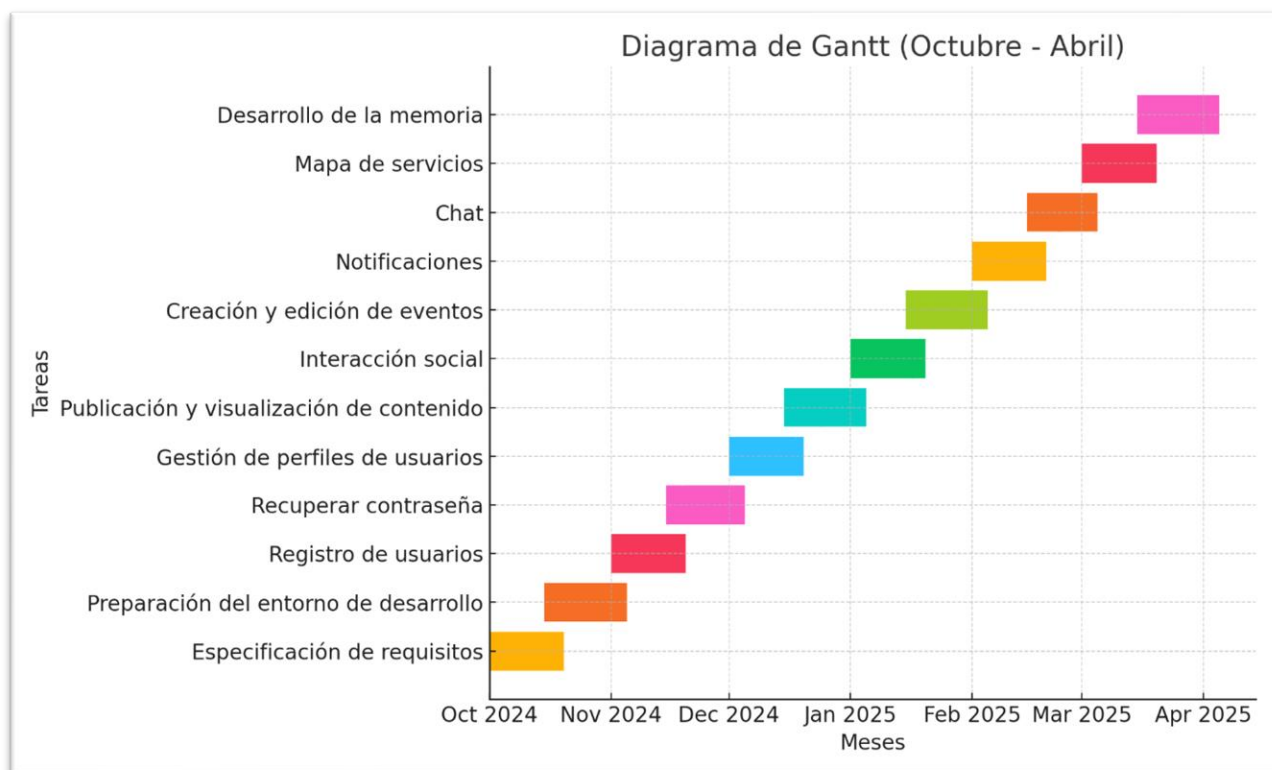
## 1.3 Plan de trabajo

Para lograr la consecución de los objetivos previstos, el desarrollo de la aplicación web de recetas se ha dividido en cuatro iteraciones<sup>1</sup>, en las cuales se han ido añadiendo nuevas funcionalidades y características con el objetivo de mejorar la experiencia del usuario y la interacción dentro de la plataforma. Para mejor visualización del contenido, se incluye el posterior calendario, tal y como se muestra en la Ilustración 1:

---

<sup>1</sup> (Pressman & Maxim, 2014)

Ilustración 1. Diagrama de Gantt



### **Iteración 1: Gestión de Usuarios**

En la primera fase, se implementaron las funcionalidades básicas que permiten a los usuarios de la aplicación gestionar sus cuentas en la plataforma.

- Registro de usuarios: Contando con una validación de datos y confirmación. Inicio y cierre de sesión: Usando métodos de autenticación segura, con el fin de evitar ataques por diccionario.
- Modificación de datos personales: Así se trata de permitir ciertos cambios en nombre, correo y contraseña.
- Eliminar una cuenta: Asegurando que los datos se eliminen o se anonimicen según las diferentes necesidades que puedan surgir.

## ***Iteración 2: Gestión de Recetas y Listas***

En esta segunda iteración, las tareas se centran en la creación y organización de recetas y en la posibilidad de compartirlas y agruparlas, incluyéndose las siguientes funcionalidades:

- Creación, modificación, publicación, eliminación y compartición de recetas.
- Búsqueda y visualización de recetas dentro de la plataforma.
- Creación de listas de recetas personalizadas por los usuarios.
- Gestión de listas, permitiendo añadir, eliminar y mover recetas entre diferentes listas.

## ***Iteración 3: Interacción con Recetas y Menús***

Durante este proceso, se lleva a cabo la interacción social dentro de la aplicación, donde se añadieron funciones con el fin de lograr la integración entre los usuarios:

- Añadir recetas de otros usuarios a listas de favoritos.
- Comentar y puntuar recetas ajenas.
- Responder comentarios en recetas propias.
- Creación y gestión de menús, agrupando recetas en planes personalizados.
- Búsqueda y visualización de menús compartidos por la comunidad.

## ***Iteración 4: Interacción Social y Comunidad***

La última fase del desarrollo se enfocó en ampliar las funcionalidades sociales y la interacción entre los usuarios:

1. Seguir y dejar de seguir a otros usuarios para ver sus actualizaciones.
2. Visualizar un feed de novedades con las actividades recientes de los usuarios seguidos.
3. Creación de foros públicos para debatir sobre recetas y menús.

4. Interacción en los foros, permitiendo a los usuarios comentar, modificar y eliminar sus intervenciones.
5. Recomendar recetas a otros usuarios y gestionar las recomendaciones recibidas.

Seguir este enfoque iterativo ha permitido desarrollar una aplicación completa y funcional, que no solo permite la gestión eficiente de recetas, sino que también fomenta la interacción social y la construcción de una comunidad gastronómica en línea.

## 2. Introduction

Nowadays, traditional cookbooks and cooking shows have taken a backseat, largely due to the rapid growth of the technology sector over the past decade. With internet-enabled devices becoming increasingly common among the general population in our country, audiences have shifted towards more accessible and immediate digital platforms. **The network** was created to bring together users seeking to connect with others who share a similar passion for cooking. The platform aims to offer a space where culinary horizons can be expanded, relationships can be built, and knowledge can be shared and explored collaboratively. In essence, the main objective of the web is to leverage the best features of social media—such as following users, sharing content, and forum-based discussions—and adapt them to a theme that, while previously explored, has not yet made a lasting impact on users in a way that invites a second chance

### 2.1 Motivation

We have always seen cooking as more than just a daily task. For us, it is a space for creativity and personal expression. Sharing a recipe, trying something new, or improving a dish thanks to someone else's input—food has the power to connect people, whether at the table or through a screen. With this vision in mind, we decided to develop this web application not just as a way to manage dishes and menus, but as a tool to bring people together through their passion for cooking.

## 2.3 Goals

The main goal of the web application is to provide users with access to a culinary-themed platform that fosters interaction and content creation around recipes and food.

- **Competitor analysis:** A thorough review of the current market and identification of relevant gaps.
- **Visual design foundation:** Definition of the color palette, layout structure, logo, and button style.
- **Technology stack selection:** Choosing the most suitable programming languages based on those learned during our five years of study.
- **Database development:** Designing the database schema, choosing the right language, and establishing the necessary relationships between entities.
- **Feature implementation:** Defining the key modules of the platform and determining the most efficient way to implement them. This included individual development, integration, and final adjustments.

## 2.4 Work plan

To meet the project objectives, the development of the recipe web application was divided into four iterations. In each phase, new functionalities and improvements were introduced to enhance the user experience and interaction within the platform. The Gantt chart below provides an overview of the timeline and structure.

### ***Iteration 1: User Management***

In the first stage, core functionalities were implemented to allow users to manage their accounts on the platform:

- User registration, including data validation and confirmation mechanisms.

- Login and logout secured with proper authentication techniques to prevent dictionary attacks.
- Editing personal information, allowing users to update their name, email, and password.
- Account deletion, ensuring data is removed or anonymized according to the user's needs.

### ***Iteration 2: Recipe and List Management***

The second iteration focused on the creation, editing, and organization of recipes, as well as the ability to group and share them. The following features were developed:

- Creating, editing, publishing, deleting, and sharing recipes.
- Searching and viewing recipes on the platform.
- Creating custom recipe lists.
- Managing lists by adding, removing, and moving recipes between them.

### ***Iteration 3: Interaction with Recipes and Menus***

This phase introduced social interaction features aimed at fostering user engagement:

- Adding other users' recipes to personal favorite lists.
- Commenting on and rating others' recipes.
- Replying to comments on one's own recipes.
- Creating and managing menus by grouping recipes into personalized meal plans.
- Searching for and viewing menus shared by the community.

### ***Iteration 4: Social Interaction and Community***

The final development phase enhanced the platform's community and interaction features:

- Following and unfollowing users to receive updates on their activity.
- Viewing a feed of recent actions from followed users.
- Creating public forums for discussing recipes and menus.
- Interacting within forums by posting, editing, or deleting comments.
- Recommending recipes to other users and managing received recommendations.

By following this iterative approach, we were able to develop a complete and functional application that not only supports efficient recipe management but also promotes social interaction and the growth of a digital culinary community.

## **3. Estado de la cuestión**

En la actualidad, el desarrollo de aplicaciones web ha experimentado un crecimiento exponencial, impulsado por la digitalización de diversos sectores y el auge de las redes sociales. Dentro de este panorama, las plataformas de gestión de recetas han adquirido una gran relevancia, proporcionando a los usuarios herramientas para organizar, compartir e interactuar en torno a la gastronomía. En este apartado, se analizarán algunas aplicaciones existentes en el ámbito de las recetas y la interacción culinaria, con el fin de comprender mejor el contexto en el que se enmarca la red social y evaluar sus características diferenciadoras.

### **3.1 Cookpad**

Cookpad es una de las plataformas más consolidadas en el ámbito de las recetas, ofreciendo un espacio donde los usuarios pueden compartir sus propias creaciones culinarias y acceder a una amplia base de datos de recetas de otros usuarios. Su modelo de interacción se basa en la publicación de recetas, la posibilidad de valorarlas y comentarlas, y un sistema de búsqueda basado en ingredientes y categorías. No obstante, su enfoque está centrado principalmente en la recopilación de recetas, sin una estructura social avanzada más allá de la interacción en las publicaciones.

### **3.2 Tasty**

Tasty, desarrollada por BuzzFeed, es una aplicación que se ha popularizado gracias a su formato de recetas en video, diseñadas para una rápida visualización y fácil ejecución. Su principal fortaleza radica en su contenido audiovisual atractivo y en la personalización de recetas según los intereses del usuario. Sin embargo, a pesar de su popularidad, Tasty no ofrece una interacción social significativa más allá de la personalización de recetas, lo que limita la posibilidad de generar una comunidad de usuarios activa dentro de la plataforma. Además, se centra sobre todo en contenido de vídeo.

### **3.3 Yummly**

Yummly es una aplicación que combina la gestión de recetas con la inteligencia artificial, permitiendo a los usuarios obtener recomendaciones personalizadas en función de sus preferencias alimenticias. Además, integra herramientas para la planificación de menús y la generación de listas de compras automáticas. A pesar de estas funcionalidades avanzadas, su modelo de interacción se basa más en la personalización individual que en la creación de una comunidad activa entre los usuarios. Además, cabe destacar que, en principio, la web se presenta solamente en inglés.

### **3.4 Diferenciación de la red social para amantes de las recetas**

Si bien las aplicaciones mencionadas cuentan con diversas funcionalidades relacionadas con la gestión y compartición de recetas, esta red busca ir un paso más allá al integrar una dimensión social más robusta. La plataforma no solo permite la creación y organización de recetas, sino que también fomenta la interacción entre los usuarios a través de la posibilidad de seguirse mutuamente, participar en foros públicos, compartir recomendaciones y calificar recetas. Además, la aplicación incorpora herramientas para la planificación de menús personalizados, ofreciendo una experiencia más integral para quienes desean gestionar su alimentación de manera eficiente mientras forman parte de una comunidad activa.

De este modo, la red social se configura como un espacio dinámico donde los usuarios pueden intercambiar experiencias gastronómicas y fortalecer sus vínculos a través de la cocina.

## 4. Tecnologías empleadas

### 4.1 Software

#### 4.1.1 XAMPP

XAMPP es un paquete de software libre que facilita la instalación y gestión de un entorno de desarrollo web local. Su nombre es un acrónimo que representa los componentes principales que incluye:

X (multiplataforma): Disponible para Windows, macOS y Linux.

A (Apache): Servidor web ampliamente utilizado.

M (MySQL/MariaDB): Sistema de gestión de bases de datos relacional.

P (PHP): Lenguaje de programación para desarrollo web.

P (Perl): Otro lenguaje compatible con servidores web.

XAMPP está diseñado para que los desarrolladores puedan probar y ejecutar sitios web en un entorno local sin necesidad de configurar manualmente cada componente. Incluye además herramientas como phpMyAdmin para la gestión de bases de datos y FileZilla para la administración de archivos. Puesto que todos los miembros del equipo sabían utilizarlo gracias a la asignatura de aplicaciones web cursada el año pasado, y su accesibilidad y facilidad de uso, se concretó como herramienta principal de testeo de versiones de la aplicación web.

#### 4.1.2 MariaDB

MariaDB es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, diseñado como una alternativa completamente compatible con MySQL. Es una bifurcación (fork) de MySQL, lo que significa que es totalmente compatible con la mayoría de las aplicaciones y comandos de MySQL. En la mayoría de los casos, una base de datos en MySQL puede migrarse a MariaDB sin cambios en el código. Además, incluye optimizaciones en la ejecución de consultas y mejoras en la gestión de memoria.

### **4.1.3 Apache**

Apache HTTP Server, es un servidor web de código abierto desarrollado por la Apache Software Foundation. Su función principal es procesar y entregar páginas web a los usuarios, permitiendo que los navegadores accedan a sitios web alojados en un servidor. Apache proporciona código abierto y gratuito, es multiplataforma y tiene compatibilidad con múltiples lenguajes (PHP, Python, Perl, Ruby).

### **4.1.4 Visual Studio Code**

Visual Studio Code (VS Code) es un editor de código fuente muy utilizado por desarrolladores por su ligereza, versatilidad y gran cantidad de extensiones<sup>2</sup>. Permite escribir, editar y organizar código en múltiples lenguajes de programación dentro de un entorno unificado. En el desarrollo de la aplicación web, se ha utilizado Visual Studio Code como entorno principal para editar todos los documentos del proyecto CSS y PHP, aprovechando funciones como el resaltado de sintaxis, el autocompletado, el control de versiones con Git y la integración con terminales para facilitar un desarrollo ágil y eficiente. Además, ha resultado de utilidad a la hora de tratar con partes del código en SQL.

## **4.2 Lenguajes de programación empleados**

### **4.2.1 HTML**

HTML (*HyperText Markup Language*) es el lenguaje de marcado estándar utilizado para la creación y estructuración de páginas web. Se basa en el uso de etiquetas (*tags*) que definen la organización y el contenido de una página web, como texto, imágenes, enlaces, formularios y otros elementos interactivos.

HTML no es un lenguaje de programación, ya que no tiene lógica de control como condicionales o bucles. En cambio, se encarga de estructurar la información de una página para que los navegadores web puedan interpretarla y mostrarla

---

<sup>2</sup> (Chacon & Straub, 2014)

correctamente. Es por ello que se decide utilizarlo para el diseño de páginas. En el proyecto, tiene un papel clave en la representación visual de la aplicación.

#### **4.2.2 CSS**

CSS (*Cascading Style Sheets*) es un lenguaje de hojas de estilo utilizado para definir la apariencia y el diseño de las páginas web<sup>3</sup>. Permite controlar aspectos como colores, fuentes, márgenes, disposición de elementos y adaptabilidad a diferentes dispositivos, separando la estructura del contenido (HTML) de su presentación visual.

#### **4.2.3 PHP**

PHP (*Hypertext Preprocessor*) es un lenguaje de programación del lado del servidor utilizado para desarrollar aplicaciones web dinámicas<sup>4</sup>. Permite la interacción con bases de datos, la gestión de sesiones y la generación de contenido personalizado, integrándose fácilmente con HTML y ejecutándose en el servidor antes de enviar la página al navegador del usuario. La selección de HTML y PHP sobre otras alternativas más modernas como python, se debe a la familiaridad del equipo con éste tipo de documentos gracias a la exploración de este tipo de lenguajes a lo largo de las asignaturas vistas en la carrera.

#### **4.2.4 JavaScript**

JavaScript es un lenguaje de programación del lado del cliente que permite añadir interactividad y dinamismo a las páginas web<sup>5</sup>. Se utiliza para manipular el contenido HTML y CSS, responder a eventos del usuario, realizar peticiones a servidores y desarrollar aplicaciones web modernas, incluyendo frameworks como React, Angular y Vue.js.

#### **4.2.5 Ajax**

---

<sup>3</sup> (Duckett, 2011)

<sup>4</sup> (PHP: The Right Way, s.f.)

<sup>5</sup> (Flanagan, 2020)

AJAX (Asynchronous JavaScript and XML) es una técnica que permitió actualizar partes de una página web sin recargarla<sup>6</sup>. Usa JavaScript para enviar y recibir datos desde un servidor en segundo plano. AJAX mejora la experiencia del usuario al hacer las páginas más rápidas y dinámicas.

#### **4.2.6 Bootstrap**

Bootstrap es un framework CSS basado en código abierto que permitió en la app crear un diseño de página responsiva y moderno. Bootstrap se construye alrededor de un sistema de cuadrícula conforme al diseño funcional de la página. Además del sistema de cuadrícula, se hace uso de la biblioteca incluyendo elementos predefinidos: botones, formas de diferentes grados de complejidad, alarmas, etc...

#### **4.2.7 SQL**

SQL es un lenguaje diseñado para gestionar bases de datos relacionales y permite realizar operaciones como consultar, modificar o eliminar datos de forma estructurada. Este lenguaje interactúa directamente con la versión de MariaDB en uso, que lo utiliza como medio principal para recibir instrucciones, interpretarlas y ejecutar las acciones solicitadas sobre la base de datos, facilitando así el control y la organización eficiente de la información almacenada.

---

<sup>6</sup> (Mozilla Developer Network, s.f.)

## 5. Arquitectura de la aplicación

A lo largo de este capítulo, va a describirse y explorarse todo el sistema que conforma la aplicación web. El objetivo principal es el de exponer de manera clara, didáctica y ordenada, toda la arquitectura del proyecto.

El enfoque del equipo ha sido basado en una filosofía minimalista, clara y funcional, a fin de mantener una arquitectura simple y fácil de editar y entender para facilitar el entendimiento y la correcta interacción entre ésta y el equipo mientras se trabajaba en su desarrollo.

### 5.1 Interacción entre el frontend y el backend

Para el desarrollo del *frontend* de la aplicación se ha utilizado el framework *Bootstrap*<sup>7</sup>, con el objetivo de facilitar la creación de una interfaz clara, responsive y coherente. Bootstrap proporciona una amplia variedad de componentes prediseñados —como botones, formularios, menús o tarjetas— así como un sistema de rejilla adaptable que permite organizar el contenido de forma eficiente en distintos dispositivos, garantizando una correcta visualización tanto en ordenadores como en móviles o tablets.

Aunque su uso puede conllevar ciertos inconvenientes, como la similitud estética entre proyectos que no personalizan los estilos o la inclusión de código no utilizado en aplicaciones más simples, las ventajas superan ampliamente estas limitaciones. La rapidez en el desarrollo, la consistencia visual y la compatibilidad entre navegadores hacen que la utilización de Bootstrap resulte especialmente beneficiosa en proyectos académicos como este.

AJAX (Asynchronous JavaScript and XML) se ha empleado en el desarrollo del frontend como mecanismo de comunicación asíncrona con el servidor. Su utilización

---

<sup>7</sup> (Bootstrap, s.f.)

permite enviar y recibir datos del backend sin necesidad de recargar la página por completo, lo que contribuye a una experiencia de usuario más fluida e interactiva.

A través de este sistema es posible actualizar partes concretas de la interfaz, formularios, tablas o resultados dinámicos, a medida que el usuario interactúa. Aunque AJAX se ejecuta en el navegador, actúa como un puente entre el frontend y el backend, facilitando el intercambio de datos de forma eficiente, habitualmente mediante el uso de objetos JSON y tecnologías como JavaScript o bibliotecas jQuery.

## 5.2 Patrones de diseño

### ***Patrones estructurales:***

- **Proxy:** Ha sido utilizado para controlar el acceso a ciertos objetos, actuando como intermediario entre el cliente y un recurso real (por ejemplo, en el acceso a recursos externos, como APIs o servicios protegidos, o en la gestión de cachés). Este patrón permite agregar una capa de control sin modificar el objeto original.
- **Decorator:** Se ha empleado para extender funcionalidades de clases sin modificar su código original, especialmente útil para añadir comportamientos de forma flexible (como validaciones adicionales, formatos, o funcionalidades condicionales).<sup>8</sup>

### ***Patrones de comportamiento:***

- **Observer:** Ha sido útil en contextos donde varios componentes deben reaccionar a un mismo evento (por ejemplo, en el frontend para actualizar dinámicamente elementos de la interfaz, o en el backend para notificar a diferentes servicios cuando se produce un cambio de estado). Facilita la implementación de sistemas reactivos o eventos.
- **Strategy:** Ha sido implementado para definir diferentes algoritmos intercambiables que cumplen una misma tarea, permitiendo que el comportamiento pueda cambiar

---

<sup>8</sup> (Gamma et al., 1995; Freeman et al., 2004)

dinámicamente en tiempo de ejecución (por ejemplo, distintas formas de ordenamiento, validación o lógica de negocio adaptable).

**Patrones creacionales:**

- **Singleton:** Se ha utilizado para garantizar que determinadas clases —como controladores de configuración o gestión de base de datos— solo tengan una única instancia global accesible desde toda la aplicación. Esto asegura un punto único de acceso y control de recursos compartidos.

### **5.3 Gestión de la base de datos**

A continuación, se procede a el análisis y exposición de la base de datos. Ésta, con la ayuda de la plataforma "XAMPP", se ha implementado utilizando MariaDB. Para ello, se adjunta un diagrama entidad - relación que fue trazado previo a el desarrollo de la red

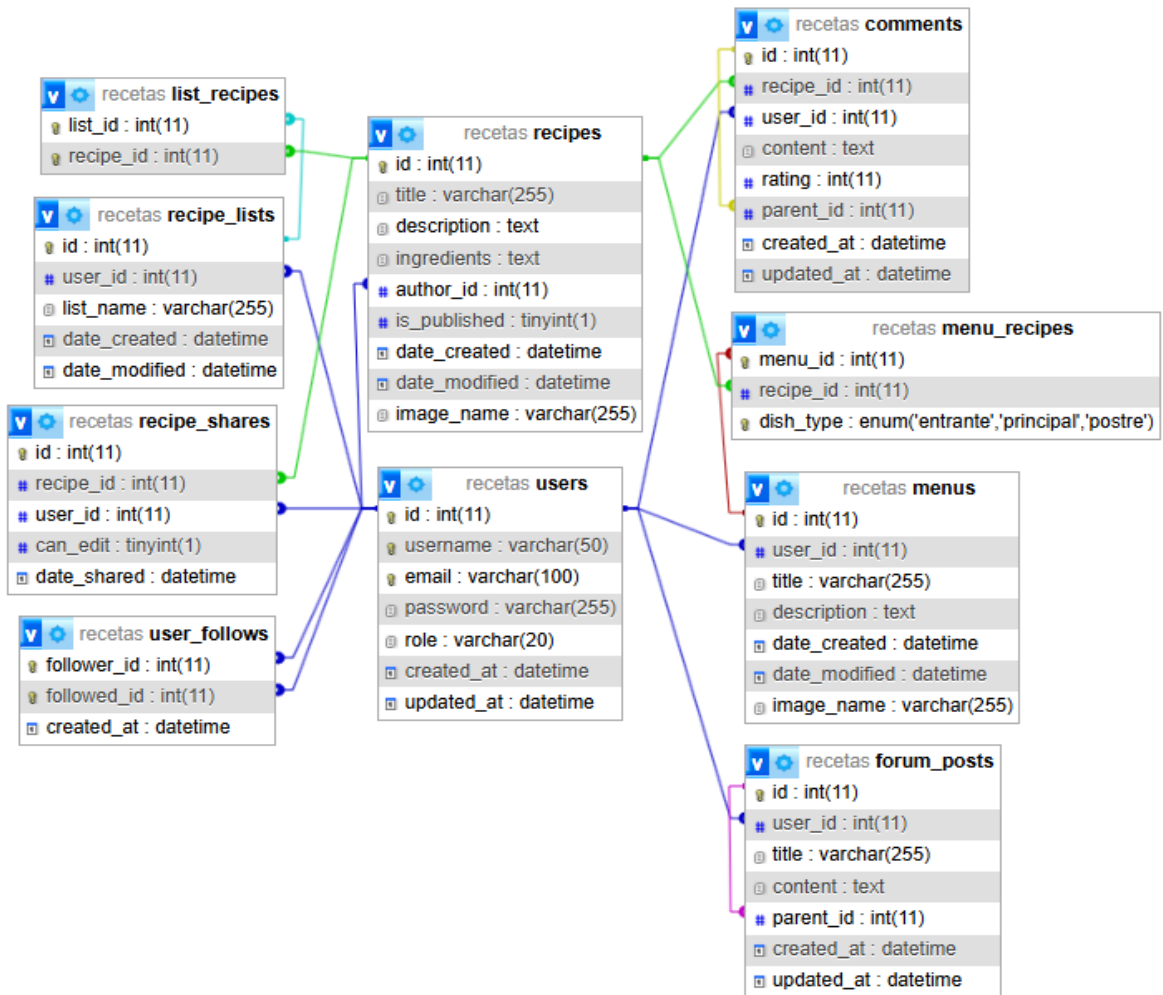


Ilustración 2. Diagrama entidad-relación

Tal como se muestra en la Ilustración 2, el proyecto se basa en una estructura simple pero funcional. El corazón del sistema está en las tablas “recipes” y “users”, que conforman la base conforme a la cual, posteriormente se teje la red de conexiones entre todas las demás tablas, para dar cabida a todas las funcionalidades necesarias. La tabla de “recipes”, la cual se vincula con otras entidades para potenciar la experiencia del usuario, tales como comentarios, menús a medida, foros, listas de recetas y acciones sociales como compartir o seguir a otros usuarios. Se puede ver una lógica evidente que vincula a los usuarios con recetas a través de diversas interacciones, facilitando no solo

la generación y conservación de contenido, sino también su estructuración, adaptación y propagación dentro de la comunidad. El modelo se centra tanto en el contenido como en la interacción entre los usuarios, proporcionando así solidez y capacidad de expansión.

Partiendo de la vista del diagrama anterior, se propone el desarrollo consiguiente a través del uso de MaríaDB para la gestión de la base de datos. Se ha resumido en las páginas siguientes, el contenido de la base de datos y de todas sus tablas:

#### **a) Tabla *comments***

- ◆ **Resumen:** Esta tabla almacena información sobre los comentarios que los usuarios pueden realizar en la aplicación.
- ◆ **Campos:**
  - **id:** (Entero) Identificador único del comentario (clave primaria).
  - **recipe\_id:** (Entero) Identificador único de las recetas incluidas en la lista (clave foránea que hace referencia a la tabla *recipes*).
  - **user\_id:** (Entero) Identificador único del usuario al que se asocia el comentario (clave foránea que hace referencia a la tabla *users*).
  - **content:** (text) Representa el contenido del comentario, es decir, el texto.
  - **rating:** (int) Identifica la puntuación que se apunta en la reseña. En la interfaz visual del usuario, se representa con un diagrama de 5 estrellas.
  - **parent\_id:** (int) Identificador que representa un comentario padre del que se desprende este. Es decir, el id del comentario al que se responde (clave foránea que hace referencia a la misma tabla).
  - **created\_at:** (datetime) Fecha de creación del comentario.
  - **updated\_at:** (datetime) Fecha de la última modificación del contenido del comentario.

### ***b) Tabla forum\_post***

- ◆ **Resumen:** Esta tabla almacena información sobre las publicaciones en la foto y quien lo publicó.
- ◆ **Campos:**
  - **id:** (Entero) Identificador único de la publicación (clave primaria).
  - **user\_id:** (Entero) Identificador único del usuario al que se asocia la publicación (clave foránea que hace referencia a la tabla users).
  - **title:** (varchar) Representa el título de la publicación.
  - **content:** (varchar) Representa el contenido de la publicación.
  - **parent\_id:** (int) Identificador que representa una publicación cabecera de la que se desprende este. Es decir, el id del comentario al que se responde (clave foránea que hace referencia a la misma tabla).
  - **created\_at:** (datetime) Fecha de creación de la publicación en el foro.
  - **updated\_at:** (datetime) Fecha de la última modificación de la publicación en el foro.

### ***c) Tabla list\_recipes***

- ◆ **Resumen:** Esta tabla almacena información sobre las listas de recetas creadas en la plataforma, indicando qué recetas pertenecen a cada lista.
- ◆ **Campos:**
  - **list\_id:** (int) Identificador único de la lista de recetas (clave primaria).
  - **recipe\_id:** (int) Identificador único de las recetas incluidas en la lista (clave primaria y foránea que hace referencia a la tabla de recetas).

### ***d) Tabla menus***

- ◆ **Resumen:** Esta tabla almacena información de los menus posibles
- ◆ **Campos:**

- **id:** (int) Identificador único del menú (clave primaria).
- **user\_id:** (int) Identificador único del usuario al que se asocia el menú (clave foránea que hace referencia a la tabla users).
- **title:** (varchar) Representa el título del menú.
- **description:** (text) Representa el contenido del menú.
- **date\_created:** (datetime) Fecha de creación de dicho menú
- **date\_modified:** (datetime) Fecha de la última modificación del menú
- **image\_name:** (varchar) Representa la imagen del menú.

#### e) *Tabla menu\_recipes*

- ◆ **Resumen:** Esta tabla almacena información sobre las listas de menús creadas en la plataforma.
- ◆ **Campos:**
  - **menu\_id:** (int) Identificador único de la lista de recetas (clave primaria).
  - **recipe\_id:** (int) Identificador único de las recetas incluidas en el menú (clave primaria y foránea que hace referencia a la tabla de recetas).
  - **dish\_type:** (enum(entrante, principal, postre)) Identificador único de las recetas incluidas en la lista (clave primaria).

#### f) *Tabla recipes*

- ◆ **Resumen:** Esta tabla almacena información acerca de las recetas que han creado los usuarios.
- ◆ **Campos**
  - **id:**(Entero) Identificador único de cada receta.(clave primaria).
  - **title:** (Varchar) Título de la receta publicada.
  - **description:** (Texto) Resumen del contenido de la receta.
  - **ingredients:** (Texto) Ingredientes que forman parte de la receta

- **author\_id:** (Entero) Referencia el id del usuario propietario de la receta. (clave foránea)
- **is\_published:** (Booleano) Se activa si la receta ha sido publicada. Si no, significa que la receta existe como borrador.
- **date\_created:**(Fecha) Representa la fecha de creación.
- **date\_modified:**(Fecha) Representa la última fecha de modificación.
- **image\_name:** (varchar) Representa la imagen de la receta.

#### **g) Tabla *recipe\_lists***

- ◆ **Resumen:** La tabla *recipe\_lists* almacena información sobre las listas creadas por los usuarios para agrupar recetas dentro de la plataforma.
- ◆ **Campos:**
  - **id:** (int) Identificador único autoincremental de cada lista (clave primaria).
  - **user\_id:** (int) Identificador del usuario propietario de la lista (clave foránea referenciada a la tabla usuarios).
  - **list\_name:** (varchar) Nombre asignado a la lista por el usuario.
  - **date\_created:** (datetime) Fecha y hora en la que se creó la lista (valor por defecto es la fecha y hora actuales).
  - **date\_modified:** (datetime) Fecha y hora en la que se realizó la última modificación de la lista. Este campo es opcional y puede tener valor nulo.

#### **h) Tabla *recipe\_shares***

- ◆ **Resumen:** La tabla *recipe\_share* almacena información sobre las recetas que han sido compartidas de un usuario a otro de la aplicación.
- ◆ **Campos:**
  - **id:** (int) Identificador único autoincremental del reenvío (clave primaria).

- **recipe\_id:** (int) Identificador de la receta que está siendo compartida (clave foránea referenciada a la tabla de recetas).
- **user\_id:** (int) Identificador del usuario (clave foránea referenciada a la tabla usuarios).
- **can\_edit:** (tinyint) Representa la posibilidad de editar la receta.
- **date\_shared:** (datetime) Fecha y hora en la que se realizó el intercambio de la receta en cuestión.

### ***i) Tabla users***

- ◆ **Resumen:** La tabla users se encarga de almacenar la información personal de los usuarios que estén registrados en la página web.
- ◆ **Campos:**
  - **id:** (int) Corresponde, identificador único autoincremental de los usuarios de la tabla (clave primaria).
  - **username:** (varchar) Se refiere a el nombre del usuario.
  - **email:** (varchar) Representa el correo electrónico del usuario.
  - **password:** (varchar) Hace referencia a la contraseña elegida por el usuario.
  - **role:** (datetime) Fecha y hora en la que se realizó el intercambio de la receta en cuestión.
  - **created\_at:** (datetime) Fecha de creación del perfil de usuario.
  - **updated\_at:** (datetime) Fecha de la última modificación del perfil de usuario.

### ***j) Tabla user\_follows***

- ◆ **Resumen:** La tabla user\_follows recoge toda la información relativa al seguimiento entre usuarios.
- ◆ **Campos:**
  - **follower\_id:** (int) Identificador del usuario seguidor (clave primaria).
  - **followed\_id:** (int) Identificador del usuario al que se sigue (clave primaria)

- **created\_at:** (datetime) Representa la fecha en la que se realiza el "follow".

## 6. Implementación de la plataforma

Para el correcto desarrollo de la aplicación web, se presentó una serie de funcionalidades y casos de uso que el equipo ha decidido seguir durante su construcción. Para ello, se aporta la siguiente lista de servicios que han sido implementados en ésta. La interfaz se diseñó siguiendo principios de usabilidad que facilitan la navegación intuitiva y accesible<sup>9</sup>

### 6.1 Gestión de usuarios.

Este módulo es considerado fundamental, para el funcionamiento del proyecto, ya que permite el acceso, la personalización y el control de las cuentas de los usuarios. A través de estas funcionalidades, se garantiza una experiencia personalizada, segura y adaptada a las necesidades de cada usuario.

#### 6.1.1 Registro y eliminación de usuario

Permite a los usuarios crear una cuenta en la plataforma o eliminarla en cualquier momento, gestionando adecuadamente su presencia en el sistema. Tal y como se muestra en la Ilustración 3, se recoge la información enviada desde el formulario y se comprueba que todos los campos estén rellenos y que las contraseñas coincidan. A continuación, se verifica si el nombre de usuario o email ya existen en la base de datos. Si no existen, se instancia el nuevo usuario, se registra y se redirige automáticamente a la página de inicio de sesión, utilizando el patrón observer para registrar el evento.

---

<sup>9</sup> (Nielsen, 1999; Norman, 2013; Shneiderman et al., 2017).

```

C: > xampp > htdocs > TFGINFO > register.php
1  <?php
2  // register.php
3  require_once 'classes/User.php';
4  require_once 'classes/LoggerObserver.php';
5  session_start();
6
7  if ($_SERVER['REQUEST_METHOD'] === 'POST') {
8      $username    = trim($_POST['username']);
9      $email       = trim($_POST['email']);
10     $password     = trim($_POST['password']);
11     $confirm_password = trim($_POST['confirm_password']);
12
13     if(empty($username) || empty($email) || empty($password) || empty($confirm_password)){
14         $error = "Todos los campos son obligatorios.";
15     } elseif($password !== $confirm_password) {
16         $error = "Las contraseñas no coinciden.";
17     } else {
18         // Verificar si el usuario o email ya existen
19         if (User::findByUsernameOrEmail($username) || User::findByUsernameOrEmail($email)) {
20             $error = "El usuario o email ya está registrado.";
21         } else {
22             $user = new User($username, $email, $password);
23             $logger = new LoggerObserver();
24             $user->attach($logger);
25             $user->register();
26             header("Location: login.php?registered=1");
27             exit();
28         }
29     }
30 }

```

Ilustración 3. Registro

Tal y como se muestra en la ilustración 4, puede observarse que:

En la clase LoggerObserver, que implementa el patrón Observer. Cada vez que se registra un usuario, esta clase genera una entrada en el archivo logs.txt, anotando la acción realizada junto con el nombre del usuario y la fecha, lo que permite mantener un registro de actividades importantes en la plataforma.

En el caso de que no exista el usuario, se llama a la clase User para la creación de un nuevo usuario. Se instancia el objeto con los datos recibidos y se aplica un hash seguro a la contraseña. Luego, mediante el método register(), se inserta el nuevo usuario en la base de datos. Además, se implementa el patrón Observer, permitiendo adjuntar observadores con attach(), eliminarlos con detach() y notificar eventos con notify(). En este caso, tras registrar al usuario, se lanza una notificación que registra en el log la acción realizada.

```
C:\xampp > htdocs > TFGINFO > classes > User.php
6 class User implements Subject {
18     public function __construct($username, $email, $password, $role = 'usuario', $id = null, $created_at = null, $updated_at = null) {
19         $this->username = $username;
20         $this->email = $email;
21         if ($id === null) {
22             $this->password = password_hash($password, PASSWORD_BCRYPT);
23         } else {
24             $this->password = $password;
25         }
26         $this->role = $role; // Asignar el rol recibido o el valor por defecto
27         $this->id = $id;
28         $this->created_at = $created_at ?? date('Y-m-d H:i:s');
29         $this->updated_at = $updated_at;
30     }
31
32     // Métodos del Patrón Observador
33     public function attach(Observer $observer) {
34         $this->observers[] = $observer;
35     }
36     public function detach(Observer $observer) {
37         // Lógica para eliminar un observador si es necesario
38     }
39     public function notify($eventData) {
40         foreach ($this->observers as $observer) {
41             $observer->update($eventData);
42         }
43     }
44
45     // Registrar nuevo usuario en la base de datos
46     public function register() {
47         $db = Database::getInstance()->getConnection();
48         $stmt = $db->prepare("INSERT INTO users (username, email, password, created_at) VALUES (:username, :email, :password, :created_at)");
49         $stmt->execute([
50             'username' => $this->username,
51             'email' => $this->email,
52             'password' => $this->password,
53             'created_at' => $this->created_at
54         ]);
55         $this->id = $db->lastInsertId();
56         $this->notify(['action' => 'register', 'user' => $this]);
57     }
58 }
```

Ilustración 4. Notificación

Desde el archivo profile.php, tal como se muestra en la Ilustración 5, se ofrece un botón con estilo de alerta que redirige al usuario al script delete\_account.php, encargado de gestionar la eliminación de cuenta.

Posteriormente, tal como se muestra en la Ilustración 5, correspondiente a delete\_account.php, se comprueba primero que el usuario esté autenticado mediante sesión. Luego se recupera el usuario desde la base de datos y, si la solicitud llega por método POST, se llama al método deleteAccount() que elimina al usuario, se destruye la sesión y se redirige a la página de registro con un mensaje de confirmación.

```
C: > xampp >htdocs > TFGINFO > delete_account.php
1  <?php
2  // delete_account.php
3  require_once 'classes/User.php';
4  require_once 'classes/LoggerObserver.php';
5  session_start();
6
7  if (!isset($_SESSION['user_id'])) {
8      header("Location: login.php");
9      exit();
10 }
11
12 $user = User::findByUsernameOrEmail($_SESSION['username']);
13 if (!$user) {
14     header("Location: logout.php");
15     exit();
16 }
17
18 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
19     $user->deleteAccount();
20     session_destroy();
21     header("Location: register.php?deleted=1");
22     exit();
23 }
```

Ilustración 5. Eliminación de cuenta

Por último, en la clase User se encuentra la función deleteAccount(), encargada de eliminar de forma permanente al usuario en la base de datos. Se utiliza una sentencia DELETE preparada para proteger contra inyecciones SQL, y tras la ejecución se notifica la acción mediante el patrón Observer, lo cual permite dejar constancia del evento en el registro de logs.

### 6.1.2 Inicio y cierre de sesión

Incluye las funcionalidades necesarias para que los usuarios puedan iniciar sesión de forma segura, así como cerrar la sesión al finalizar su uso de la aplicación.

Tal como se muestra en la ilustración 6, que corresponde al archivo login.php, donde se validan los campos del formulario y se utiliza una estrategia de autenticación (StandardLoginStrategy) para verificar las credenciales. Si los datos son correctos, se

crea un observador para registrar el evento de inicio de sesión, se actualizan las variables de sesión (user\_id, username, role) y se redirige al usuario a la página principal. En caso contrario, se muestra un mensaje de error.

```
C: > xampp > htdocs > TFGINFO > login.php
1  <?php
2  // File: login.php
3  require_once 'classes/StandardLoginStrategy.php';
4  require_once 'classes/LoggerObserver.php';
5  session_start();
6
7  if ($_SERVER['REQUEST_METHOD'] === 'POST') {
8      $identifier = trim($_POST['identifier']);
9      $password  = trim($_POST['password']);
10
11     if(empty($identifier) || empty($password)) {
12         $error = "Ambos campos son obligatorios.";
13     } else {
14         $loginStrategy = new StandardLoginStrategy();
15         $user = $loginStrategy->login($identifier, $password);
16         if($user) {
17             $logger = new LoggerObserver();
18             $user->attach($logger);
19             $user->notify(['action' => 'login', 'user' => $user]);
20
21             $_SESSION['user_id'] = $user->id;
22             $_SESSION['username'] = $user->username;
23             $_SESSION['role'] = $user->role;
24             header("Location: index.php");
25             exit();
26         } else {
27             $error = "Credenciales incorrectas.";
28         }
29     }
30 }
```

Ilustración 6. Validación de campos

Y en cuanto a la finalización de sesión, se hace uso del botón "Cerrar sesión" desde la clase profile.php, que a su vez llama a la clase logout.php, que será la última y encargada de destruir la sesión.

### 6.1.3 Modificación de perfil

Ofrece la opción de personalizar el perfil del usuario, actualizando datos como su nombre, imagen o preferencias gastronómicas. Tal como se muestra en la Ilustración 7, el formulario HTML de edición de perfil dentro del archivo profile.php. Los campos de

nombre de usuario y correo electrónico están precargados con los valores actuales del usuario, y opcionalmente se permite cambiar la contraseña. Al enviar el formulario, se realiza una petición POST al mismo archivo, donde se validan y actualizan los datos según sea necesario. Esta vista proporciona también accesos directos para eliminar la cuenta o cerrar sesión.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Perfil de Usuario</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <link rel="stylesheet" href="assets/css/style.css">
</head>
<body class="bg-light">
  <div class="container mt-5">
    <h2 class="text-center mb-4">Perfil de Usuario</h2>
    <p class="text-center">Bienvenido, <?= $displayUser->getDisplayUserName(); ?></p>
    <?php if(isset($error)): ?>
      <div class="alert alert-danger"><?= htmlspecialchars($error) ?></div>
    <?php endif; ?>
    <?php if(isset($success)): ?>
      <div class="alert alert-success"><?= htmlspecialchars($success) ?></div>
    <?php endif; ?>
    <form action="profile.php" method="post">
      <div class="form-group">
        <label for="username">Nombre de Usuario</label>
        <input type="text" class="form-control" id="username" name="username" required value="<?= htmlspecialchars($user->username) ?>">
      </div>
      <div class="form-group">
        <label for="email">Correo Electrónico</label>
        <input type="email" class="form-control" id="email" name="email" required value="<?= htmlspecialchars($user->email) ?>">
      </div>
      <div class="form-group">
        <label for="password">Nueva Contraseña (opcional)</label>
        <input type="password" class="form-control" id="password" name="password">
      </div>
      <div class="form-group">
        <label for="confirm_password">Confirmar Nueva Contraseña</label>
        <input type="password" class="form-control" id="confirm_password" name="confirm_password">
      </div>
      <button type="submit" class="btn btn-primary btn-block">Actualizar Perfil</button>
    </form>
    <div class="mt-3 text-center">
      <a href="delete_account.php" class="btn btn-danger">Eliminar Cuenta</a>
    </div>
    <div class="mt-3 text-center">
      <a href="logout.php">Cerrar Sesión</a>
    </div>
  </div>
</body>
</html>
```

Ilustración 7. Validación y actualización

## 6.2 Gestión de recetas

Este módulo constituye el núcleo de la plataforma, ya que permite a los usuarios crear, editar y compartir recetas con la comunidad, facilitando su organización, visualización y recuperación.

## 6.2.1 Creación, modificación y eliminación de receta

Incluye los formularios y métodos necesarios para añadir nuevas recetas o editar las existentes. Tal como se muestra en la Ilustración 8, se presenta el archivo `create_recipe.php`, encargado de gestionar el formulario de creación de recetas. Se valida que el usuario esté autenticado mediante la sesión, y posteriormente se recogen y validan los campos enviados por POST. Se comprueba que los campos obligatorios estén completos y que el nombre de la imagen tenga una extensión válida (.jpg, .jpeg o .png). Si los datos son correctos, se crea una instancia de la clase `Recipe` con los valores recibidos y se almacena en la base de datos mediante el método `create()`.

```
C:\> xampp > htdocs > TFGINFO > create_recipe.php
1  <?php
2  // create_recipe.php
3  session_start();
4  if (!isset($_SESSION['user_id'])) {
5      header("Location: login.php");
6      exit();
7  }
8  require_once 'classes/Recipe.php';
9
10 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
11     // Recoger y validar datos
12     $title = trim($_POST['title']);
13     $description = trim($_POST['description']);
14     $ingredients = trim($_POST['ingredients']);
15     $image_name = trim($_POST['image_name']);
16     $is_published = isset($_POST['is_published']) ? 1 : 0;
17
18     if(empty($title) || empty($description) || empty($ingredients)) {
19         $error = "Todos los campos son obligatorios.";
20     } elseif (!empty($image_name) && !preg_match('/\.(jpg|jpeg|png)$/i', $image_name)) {
21         $error = "La imagen debe tener extensión .jpg, .jpeg o .png.";
22     } else {
23         $recipe = new Recipe($title, $description, $ingredients, $_SESSION['user_id'], $is_published, $image_name);
24         $recipe->create();
25         header("Location: recetas.php?created=1");
26         exit();
27     }
28 }
```

Ilustración 8. Archivo `create_recipe.php`

El método `create()` de la clase `recipe`, será el encargado de insertar una nueva receta en la base de datos utilizando una consulta preparada para mayor seguridad. Se guardan los valores correspondientes al título, descripción, ingredientes, autor, estado de publicación, fecha de creación y nombre de la imagen. Tras la inserción, se almacena el ID generado y se lanza una notificación usando el patrón `Observer`, que permite registrar la acción si se desea.

Respecto a la modificación de una receta, se utiliza la clase `edit_recipe.php` para rellenar, vía formulario, los campos necesarios para la actualización de las variables que representan a la receta. Los campos se cargan automáticamente con los valores actuales de la receta (título, descripción, ingredientes, nombre de imagen y estado de publicación). Al enviar el formulario mediante POST, los nuevos datos se procesan para actualizar la información en la base de datos. Esta vista permite al usuario revisar y modificar su receta de manera sencilla y controlada.

Por otro lado, para eliminar una receta de la base de datos, se ha implementado el método `delete()` dentro de la clase `Recipe`. Este método comienza estableciendo una conexión con la base de datos a través de la clase `Database`. A continuación, se prepara una sentencia SQL DELETE que elimina el registro de la tabla `recipes` cuyo identificador (`id`) coincide con el de la receta actual. La ejecución se realiza mediante una consulta preparada, lo que garantiza la seguridad frente a inyecciones SQL. Finalmente, se invoca el método `notify()` del patrón `Observer`, notificando que se ha realizado una acción de tipo "delete\_recipe", lo cual permite dejar un registro del evento si se cuenta con un observador adjunto.

Además, la creación de recetas está concebida para que éstas no se publiquen de manera inmediata si el usuario no lo desea. La opción de publicar puede activarse gracias a una casilla de verificación durante la creación y edición de recetas.

### **6.2.2 Búsqueda de receta**

Tal como se muestra en la Ilustración 9, El fragmento de código mostrado implementa un sistema flexible de búsqueda y filtrado de recetas. En primer lugar, si el usuario ha introducido texto en el campo de búsqueda, se construye una consulta SQL dinámica que permite buscar coincidencias parciales en el título, la descripción o los ingredientes de las recetas utilizando la cláusula `LIKE`. Una vez definidas todas las condiciones, se ordenan los resultados por fecha de creación, ya sea ascendente o descendente según el valor de `$sortOrder`. Finalmente, se preparan y ejecutan las sentencias con parámetros vinculados para evitar inyecciones SQL, y se obtienen los resultados mediante `fetchAll`. Además, se ejecuta una segunda consulta que obtiene la

lista de autores únicos que han publicado recetas, lo cual permite rellenar dinámicamente un selector de filtro por autor en el formulario de búsqueda.

```
31 // Si se ingresó texto en el campo de búsqueda, se busca en título, descripción o ingredientes
32 if (!empty($search)) {
33     $sql .= " AND (title LIKE :search OR description LIKE :search OR ingredients LIKE :search)";
34     $params[':search'] = "%$search%";
35 }
36
37 // Si se ha seleccionado un autor para filtrar
38 if (!is_null($filterAuthor)) {
39     $sql .= " AND author_id = :author_id";
40     $params[':author_id'] = $filterAuthor;
41 }
42
43 // Si se ha ingresado algún filtro por ingredientes (búsqueda parcial)
44 if (!empty($filterIng)) {
45     $sql .= " AND ingredients LIKE :filter_ing";
46     $params[':filter_ing'] = "%$filterIng%";
47 }
48
49 // Ordenar por fecha de creación
50 $sql .= " ORDER BY date_created $sortOrder";
51
52 // Preparar y ejecutar la consulta
53 $stmt = $db->prepare($sql);
54 $stmt->execute($params);
55 $recipes = $stmt->fetchAll(PDO::FETCH_ASSOC);
56
57 // Para el filtro por autor, obtenemos la lista de usuarios que hayan publicado recetas
58 $authors = [];
59 $authStmt = $db->query("SELECT DISTINCT u.id, u.username FROM users u
60                       INNER JOIN recipes r ON u.id = r.author_id
61                       ORDER BY u.username ASC");
62 $authors = $authStmt->fetchAll(PDO::FETCH_ASSOC);
63
```

Ilustración 9. Búsqueda y filtrado de recetas

La búsqueda de las recetas se realiza a través de la clase "recetas.php", que es la que, además, permite la visualización de todas aquellas que pertenezcan al autor, junto con aquellas de otros usuarios que estén publicadas. Es en esta interfaz en la que se adjunta la interfaz gráfica con la que el usuario puede interactuar a fin de crear nuevas recetas, o eliminar y editar aquellas existentes que sean de su propiedad (o todas ellas, si se posee el rol de administrador).

## 6.3 Gestión de listas

Las listas permiten a los usuarios organizar sus recetas favoritas, ideas futuras o colecciones temáticas, facilitando su acceso y gestión.

### 6.3.1 Creación de lista y visualización

En el archivo `create_list.php`, se implementa la lógica para permitir a un usuario crear una nueva lista personalizada. Primero se comprueba que el usuario esté autenticado mediante la variable de sesión `user_id`. Luego, si se ha enviado el formulario por método POST, se recoge el nombre de la lista desde el campo `list_name` y se valida que no esté vacío. En caso de que el campo sea válido, se instancia un nuevo objeto de la clase `RecipeList`, pasándole el ID del usuario y el nombre de la lista, y se guarda en la base de datos mediante el método `create()`. Finalmente, el usuario es redirigido a la página donde se visualizan sus listas, con una notificación de éxito.

Respecto a la visualización de las listas, pueden verse todas ellas en la clase `my_lists.php`, desplegable desde la barra superior. Si se pulsa el botón de “ver”, “eliminar” o “editar”, se llamará a “`view_list.php`”, “`delete_list.php`” o “`edit_list.php`” respectivamente, que a su vez conducen a las opciones para ver, editar o eliminar una lista, y cuyas implementaciones son bastante redundantes con respecto a fragmentos de la aplicación ya comentados con anterioridad.

### 6.3.2 Adición y exclusión de la lista

Tal como se muestra en la Ilustración 10, permite añadir o eliminar recetas de las listas existentes de forma sencilla y rápida. Para añadir una receta a la lista, tan solo hay que acceder a la receta correspondiente desde “`recetas.php`”, y hacer click sobre el botón “añadir a lista” en la vista de la receta en cuestión.

```

16 // Se asume que el usuario selecciona la lista a través de un formulario (por GET o POST)
17 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
18     $listId = isset($_POST['list_id']) ? intval($_POST['list_id']) : 0;
19     if ($listId <= 0) {
20         die("Selecciona una lista válida.");
21     }
22     // Buscar la lista del usuario
23     $list = RecipeList::findById($listId, $user_id);
24     if (!$list) {
25         die("No tienes permisos sobre esa lista.");
26     }
27     // Agregar la receta a la lista
28     if ($list->addRecipe($recipeId)) {
29         header("Location: view_recipe.php?id=$recipeId&added=1");
30         exit();
31     } else {
32         die("La receta ya se encuentra en la lista.");
33     }
34 }

```

Ilustración 10. Añadir receta

Esto automáticamente hace una llamada a un formulario que permite especificar la lista a la que se desea añadir la receta, y mediante la clase `add_to_list.php`, que recibe dicho formulario, se busca la lista y se agrega a ésta. En primer lugar, se valida que la solicitud se haya enviado mediante método POST y que el `list_id` recibido sea válido (mayor que cero). Luego, se intenta recuperar la lista con el método `findById`, verificando que efectivamente pertenezca al usuario autenticado. Si la lista es válida, se llama al método `addRecipe()` para asociar la receta correspondiente.

El proceso de eliminación de una receta de la lista consiste únicamente en clicar sobre el botón “eliminar de la lista”, presente en la clase “`view_list.php`”, mediante la cual se ofrece la vista estándar de todas las recetas en la lista. En este punto se ofrece también la posibilidad de mover las recetas que se deseen a otra lista.

## 6.4 Gestión de interacciones sociales

Este módulo potencia la comunidad dentro de la web de recetas mediante funciones de comentarios, participación en foros y seguimiento de otros usuarios.

### **6.4.1 Creación, edición y eliminación de comentarios y mensajes en el foro**

Esta sección trata de gestionar la creación de nuevos comentarios sobre recetas. A la hora de añadir un nuevo comentario, primero se valida que la petición llegue mediante el método POST. A continuación, se recogen los datos enviados por el formulario: el ID de la receta (`recipe_id`), el contenido del comentario, la puntuación (`rating`) y un posible ID de comentario padre (`parent_id`) en caso de tratarse de una respuesta. Se verifica que la información sea válida (por ejemplo, que el contenido no esté vacío y que la puntuación esté entre 1 y 5). Si pasa la validación, se creará una nueva instancia de la clase `Comment` y se guardará en la base de datos. Finalmente, se redirige al usuario de nuevo a la receta correspondiente, con un parámetro de éxito en la URL.

En este fragmento, tal y como muestra la Ilustración 11, se gestiona la edición de un comentario existente. Primero se comprueba que el usuario tenga permisos para editarlo, es decir, que sea el autor del comentario o tenga rol de administrador. Si se pasa esta validación, se recogen los nuevos datos del formulario (contenido y valoración) y se verifica que no estén vacíos ni fuera de los valores permitidos (la valoración debe estar entre 1 y 5). Si los datos son válidos, se crea una nueva instancia del objeto `Comment` con la información existente y los nuevos valores actualizados. Luego, se llama al método `update()` para aplicar los cambios en la base de datos. En caso de éxito, se redirige al usuario a la vista de la receta con un mensaje de confirmación.

```

28
29 // Verificar permisos; el usuario debe ser el autor o admin
30 $user_id = $_SESSION['user_id'];
31 $user_role = isset($_SESSION['role']) ? $_SESSION['role'] : '';
32 if ($commentData['user_id'] != $user_id && $user_role != 'admin') {
33     die("No tienes permisos para editar este comentario.");
34 }
35
36 $error = "";
37 $success = "";
38 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
39     $content = isset($_POST['content']) ? trim($_POST['content']) : "";
40     $rating = isset($_POST['rating']) ? intval($_POST['rating']) : 0;
41
42     if (empty($content) || $rating < 1 || $rating > 5) {
43         $error = "Por favor, completa el comentario y asegúrate de que la valoración esté entre 1 y 5.";
44     } else {
45         // Actualizar el comentario
46         // Se crea un objeto Comment usando los datos existentes y se actualizan los campos
47         $comment = new Comment($commentData['recipe_id'], $commentData['user_id'], $content, $rating, $commentData['parent_id'], $commentData['id'], $commentData['created_at']);
48         if ($comment->update()) {
49             $success = "Comentario actualizado con éxito.";
50             // Redirigir a la receta o mostrar mensaje de éxito
51             header("Location: view_recipe.php?id=" . $commentData['recipe_id'] . "&comment_edited=1");
52             exit();
53         } else {
54             $error = "Error al actualizar el comentario.";
55         }
56     }
57 }

```

Ilustración 11. Edición de comentarios

Para la eliminación de los comentarios, se hace uso del botón "eliminar" sobre la vista de los comentarios de una receta. Ello redirecciona la petición a la clase "delete\_comment.php", y que realiza un procedimiento similar al del eliminado de recetas o cuentas de usuario, llamando a la base de datos y eliminando el contenido correspondiente.

Para el caso de los mensajes en el foro, el procedimiento es muy similar. Para añadirlos, en primer lugar, se accede al foro desde la barra superior de navegación, y desde "foro.php" se ofrecen opciones para añadir mensajes y ver hilos. Estas funcionalidades tienen una implementación de naturaleza similar a los casos mencionados anteriormente. En "create\_post.php" se adjunta el código necesario para la adición de nuevos contenidos en el foro. Esto se realiza rellenando el formulario pertinente, cuyos campos son comprobados posteriormente por esta clase. En ella, el programa se asegura de que no haya campos obligatorios vacíos, y de si la publicación es o no respuesta de otro mensaje anterior.

La vista en "view\_thread.php" muestra todos los hilos del foro. Desde esta pantalla se pueden eliminar o modificar los comentarios escritos por el usuario (o todos, si el rol del usuario es el de administrador), así como responder a mensajes de un hilo ya existente.

## 6.4.2 Seguimiento entre usuarios

La funcionalidad de seguimiento se implementa mediante dos archivos principales: "follow.php" y "followed\_users.php". El primero permite al usuario autenticado seguir o dejar de seguir a otros mediante acciones recibidas por GET. Para ello, se validan los datos recibidos, se evita que se siga a sí mismo y se gestionan las relaciones en la tabla "user\_follows", bien se añade una nueva fila con INSERT IGNORE si se trata de un seguimiento, o se elimina con DELETE si la acción es dejar de seguir. Las respuestas se devuelven en formato JSON para facilitar su integración en el frontend, por ejemplo, mediante peticiones AJAX. Por otro lado, el archivo "followed\_users.php", tal y como se muestra en la Ilustración 12, se encarga de mostrar la lista de personas que el usuario autenticado sigue. Para ello, realiza una consulta con INNER JOIN entre la tabla "user\_follows" y "users", filtrando los resultados por el ID del seguidor y ordenando alfabéticamente por nombre de usuario. Ambos archivos trabajan sobre la misma tabla de relaciones y permiten mantener actualizada y accesible la red de conexiones dentro de la plataforma.

```
C: > xampp > htdocs > TFGINFO > followed_users.php
1  <?php
2  // followed_users.php
3  session_start();
4  if (!isset($_SESSION['user_id'])) {
5      header("Location: login.php");
6      exit();
7  }
8  require_once 'classes/Database.php';
9
10 $db = Database::getInstance()->getConnection();
11
12 // Obtener la lista de usuarios seguidos
13 $stmt = $db->prepare("SELECT uf.followed_id, u.username
14                      FROM user_follows uf
15                      INNER JOIN users u ON uf.followed_id = u.id
16                      WHERE uf.follower_id = :follower_id
17                      ORDER BY u.username ASC");
18 $stmt->execute([':follower_id' => $_SESSION['user_id']]);
19 $followedUsers = $stmt->fetchAll(PDO::FETCH_ASSOC);
20
```

Ilustración 12. Funcionalidad de seguimiento.

## **6.5 Gestión de menú**

La gestión de menús sigue una lógica muy similar a la gestión de listas, ya que ambas funcionalidades permiten agrupar recetas de forma personalizada por parte del usuario. Al igual que con las listas, los menús pueden crearse, visualizarse, modificarse o eliminarse, asociando recetas existentes a una estructura definida por el usuario. Sin embargo, los menús están pensados para una organización más estructurada y funcional, ya que pueden representar planes de comida y permiten funcionalidades específicas como la compartición con otros usuarios o su visualización pública. A nivel técnico de la implementación, muchas de las operaciones como las inserciones, actualizaciones o filtrado de recetas dentro del menú reutilizan estructuras similares a las utilizadas en la gestión de listas, como la búsqueda, creación y eliminación, cuya estructura de código es similar a casos ya mencionados, aportando coherencia a la estructura seguida durante todo el proyecto.

## **6.6 Administración**

Este módulo incluye las funcionalidades exclusivas del administrador, relacionadas con la gestión general del sistema, el control de contenidos y el mantenimiento de la comunidad segura y activa. La administración del sistema supone un caso especial dentro de la implementación, ya que consiste, en realidad, en el uso del campo "rol" del usuario.

El rol del administrador le permite interactuar de maneras que el resto de los usuarios no pueden, como por ejemplo el acceso a toda la información de la plataforma, y la potestad para editar o eliminar mensajes y menús sin la necesidad de ser propietarios de estos elementos, ni la necesidad de autorización previa por parte del usuario propietario.

## 7. Conclusiones y trabajo futuro

A lo largo de este Trabajo de Fin de Grado se ha desarrollado una aplicación web enfocada en la gestión y compartición de recetas, integrando funcionalidades sociales y de organización personal que enriquecen la experiencia del usuario. El objetivo principal ha sido diseñar una plataforma intuitiva, funcional y segura<sup>10</sup>, que permita a los usuarios crear, modificar y compartir recetas, así como organizarlas en listas o menús personalizados y participar activamente en una comunidad mediante comentarios, foros y sistema de seguimiento.

Durante el proceso, se ha implementado características clave como el registro y gestión de usuarios, el sistema de creación y edición de contenido, y la estructuración lógica mediante patrones de diseño como el Observer, que permite llevar un seguimiento de acciones relevantes del sistema. Además, se ha tenido en cuenta la modularidad y escalabilidad de la aplicación, permitiendo que cada funcionalidad esté bien aislada, reutilizable y fácilmente ampliable en el futuro.

El resultado es una aplicación web coherente, con un diseño centrado en el usuario<sup>11</sup>, que ofrece una experiencia completa y fluida en torno a la cocina y el intercambio de conocimientos gastronómicos. Gracias a la estructuración clara del código y a las funcionalidades desarrolladas, se ha logrado sentar una base sólida sobre la que podrían añadirse futuras mejoras.

Si bien el proyecto ha abarcado una parte importante de las funcionalidades esenciales, el tiempo disponible ha condicionado el alcance final. No obstante, se ha identificado diversas funcionalidades complementarias que enriquecerían significativamente la plataforma y que quedan propuestas como trabajo futuro:

---

<sup>10</sup> (Norman, 2013)

<sup>11</sup> (Nielsen, 1999)

- **Exportación de menús o listas:** Incorporar la posibilidad de exportar un menú o lista en formato PDF para su impresión o uso offline.
- **Aplicación móvil:** Desarrollar una versión móvil con React Native para mejorar el acceso y la interacción desde dispositivos móviles, especialmente útil para consultar recetas desde la cocina.
- **Etiquetado de recetas y filtros avanzados:** Introducir etiquetas (por ejemplo, "vegana", "sin gluten", "rápida") y permitir una búsqueda más avanzada y personalizada.
- **Ampliar el componente social,** añadiendo más funcionalidades que permitan la creación de comunidades, añadiendo servicios como la integración con otras redes sociales.
- **Añadir herramientas que amplíen los servicios ofrecidos,** como la integración de funcionalidades que aporten información sobre los ingredientes y dónde obtenerlos, o información acerca del valor nutricional de los menús, así como la adición de un sistema de menús y listas que permita hacer de las listas algo más que un diccionario de recetas.
- **Aprovechamiento de funcionalidades basadas en inteligencia artificial generativa** en las búsquedas o como consulta respecto al contenido de las recetas y menús.

Estas mejoras, además de enriquecer la experiencia de usuario, aportarían mayor dinamismo y escalabilidad a la plataforma, acercándola a un entorno más profesional y competitivo, alineando al proyecto con las tendencias en innovación y experiencia de usuario<sup>12</sup>. En definitiva, el desarrollo realizado constituye una base sólida y coherente que puede evolucionar hacia un sistema aún más completo y adaptado a los hábitos actuales de consumo y creación de contenido digital en el ámbito gastronómico.

---

<sup>12</sup> (O'Reilly, 2005; Lockwood, 2010)

A continuación, para una revisión de la implementación de estas mejoras, adjuntamos el enlace al código fuente:

<https://drive.google.com/file/d/1rZ8aXnOSGi7Y-xGjcJGzAwByJEjyyjcx/view?usp=sharing>

## 8. Conclusions and future work

Throughout this Final Degree Project, a web application has been developed focused on the management and sharing of recipes, integrating social and personal organization functionalities that enrich the user experience. The main goal has been to design an intuitive, functional, and secure platform that enables users to create, modify, and share recipes, organize them into personalized lists or menus, and actively participate in a community through comments, forums, and a following system. During the development process, key features were implemented such as user registration and management, content creation and editing tools, and logical structuring through design patterns like the Observer, which allows for tracking relevant system actions. Furthermore, the modularity and scalability of the application were taken into account, ensuring that each functionality is well-isolated, reusable, and easy to expand in the future. The result is a coherent web application with a user-centered design, offering a complete and seamless experience around cooking and the exchange of culinary knowledge. Thanks to the clear code structure and the functionalities developed, a solid foundation has been laid upon which future improvements can be built.

Although the project has covered a significant portion of the essential functionalities, the available time inevitably limited the final scope. Nevertheless, several complementary features have been identified that would significantly enrich the platform and are proposed as future work:

- **Exporting menus or lists:** Incorporating the ability to export a menu or list in PDF format for printing or offline use.

- **Mobile application:** Developing a mobile version using React Native to improve accessibility and interaction from mobile devices, especially useful for consulting recipes while cooking.
- **Recipe tagging and advanced filters:** Introducing tags (e.g., “vegan”, “gluten-free”, “quick”) and enabling more advanced and personalized search functionalities.

**Enhancing the social component:** Adding more community-focused features such as integration with other social media platforms to foster engagement.

- **Extending service-oriented tools:** Implementing features that provide information about ingredients and where to find them, nutritional value of menus, and advanced systems for managing menus and lists that go beyond a simple recipe index.
- **Leveraging generative AI functionalities:** Using artificial intelligence for advanced searches and contextual queries related to recipes and menus.

These improvements would not only enrich the user experience but also bring greater dynamism and scalability to the platform, bringing it closer to a more professional and competitive environment. Ultimately, the development carried out provides a solid and coherent base from which the system can evolve into an even more complete and user-adapted platform.

## 9. Contribuciones Personales

### Vicente Garrido Juarez

Durante el desarrollo de este Trabajo Fin de Grado, mi principal contribución se ha centrado en el diseño e implementación de la interfaz de usuario (frontend) de la aplicación red social, trabajando en estrecha colaboración con mis compañeros para garantizar que la aplicación sea visualmente atractiva, intuitiva y fácil de usar. Me focalicé en el **diseño y desarrollo del frontend:** Utilizando HTML, CSS, JavaScript y el framework Bootstrap, he creado las diferentes páginas y componentes de la interfaz, asegurando la correcta visualización y funcionamiento en distintos dispositivos, además

de en la **gestión de la experiencia de usuario** participando en la definición de la arquitectura de la información y la usabilidad de la aplicación, buscando siempre la mejor experiencia posible para el usuario. Además de en la realización de **pruebas y depuración del frontend**, realizando pruebas exhaustivas de la interfaz en diferentes navegadores y dispositivos para identificar y corregir posibles errores o problemas de visualización.

### **Pablo González Corredor**

Mi principal responsabilidad en el proyecto ha sido el diseño e implementación del backend de la aplicación web. Esto ha implicado la creación de la lógica del servidor, la gestión de la base de datos y la implementación de las funcionalidades principales de la aplicación. Incluyendo el **diseño e implementación del backend**, desarrollando la lógica del servidor utilizando PHP, encargándome de la gestión de usuarios, recetas, listas, menús, foros y otras funcionalidades de la aplicación. Por otro lado, la **gestión de la base de datos**, diseñando y gestionado la base de datos utilizando MariaDB, incluyendo la creación de las tablas, las relaciones entre ellas y las consultas necesarias para el correcto funcionamiento de la aplicación.

### **Elena Cantalejo Sánchez**

Mi rol en el desarrollo del proyecto se ha centrado en la planificación, coordinación y gestión del proyecto, así como en la implementación de funcionalidades específicas tanto en el frontend como en el backend. Realizando la **planificación y gestión del proyecto**, participando en la definición de los objetivos, la elaboración del plan de trabajo y el seguimiento del progreso del proyecto, asegurando el cumplimiento de los plazos y la correcta asignación de tareas. Además de la **especificación de requisitos y casos de uso**, definiendo los requisitos funcionales y no funcionales de la aplicación, así como en la elaboración de los casos de uso que describen la interacción de los usuarios con el sistema. Además de la **implementación de funcionalidades**, desarrollando funcionalidades específicas tanto en el frontend como en el backend, incluyendo la gestión de listas de recetas, la implementación de los foros y la gestión de las interacciones sociales entre usuarios.

El desarrollo de red social para amantes de las recetas de cocina ha sido un esfuerzo colaborativo en el que cada miembro del equipo ha aportado sus conocimientos y habilidades para lograr un resultado final satisfactorio.

Hemos trabajado de manera coordinada y eficiente, complementándonos en las diferentes áreas del proyecto. La comunicación fluida y el intercambio constante de ideas han sido fundamentales para superar los retos y alcanzar los objetivos establecidos.

Consideramos que el resultado final refleja el esfuerzo y la dedicación de cada uno de nosotros, y estamos orgullosos de haber desarrollado una aplicación web que cumple con los requisitos planteados y que tiene el potencial de ser una herramienta útil y atractiva para los usuarios interesados en la gastronomía.

## 10. Bibliografía

- Bootstrap. (s.f.). *Bootstrap documentation*. <https://getbootstrap.com>
- Chacon, S., & Straub, B. (2014). *Pro Git* (2.ª ed.). Apress. <https://git-scm.com/book/en/v2>
- Duckett, J. (2011). *HTML and CSS: Design and build websites*. Wiley.
- Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7.ª ed.). O'Reilly Media.
- Freeman, E., Robson, E., Bates, B., & Sierra, K. (2004). *Head First Design Patterns*. O'Reilly Media.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Garrido, A., & Rossi, G. (2002). Web application development using the model/view/controller design pattern. En *Web Engineering* (pp. 180–197). Springer.
- Lockwood, T. (2010). *Design Thinking: Integrating Innovation, Customer Experience, and Brand Value*. Allworth Press.
- Mozilla Developer Network. (s.f.). *AJAX introduction*. <https://developer.mozilla.org/es/docs/Web/Guide/AJAX>
- Nielsen, J. (1999). *Designing Web Usability: The Practice of Simplicity*. New Riders.
- Norman, D. A. (2013). *The Design of Everyday Things* (Rev. ed.). MIT Press.
- O'Reilly, T. (2005). *What is Web 2.0?* O'Reilly Media. <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>
- PHP: The Right Way. (s.f.). *A modern PHP best practices guide*. <https://phptherightway.com>
- Pressman, R. S., & Maxim, B. R. (2014). *Ingeniería del software: Un enfoque práctico* (8.ª ed.). McGraw-Hill.

## **1. Actores**

A continuación, se enumeran cuáles son los principales actores del sistema y se realiza la descripción correspondiente de cada uno de ellos:

### **1.1. Usuario registrado.**

Aquellos usuarios que se registren en la aplicación podrán disponer de una serie de privilegios frente a aquellos usuarios que no lo hagan. Podrán interactuar con las recetas sin ningún tipo de restricción. Esto incluye su creación, modificación o eliminación, así como funcionalidades relacionadas con la búsqueda y visualización de recetas, usuarios, menús y listas. Además, dispone de la posibilidad de crear, eliminar, modificar y mover listas y menús de recetas para agruparlas. Por último, tiene permitido utilizar las funciones sociales de la plataforma, que se resumen en el seguimiento de usuarios, la publicación de comentarios, aceptar recomendaciones, crear foros públicos y ver actualizaciones de usuarios sugeridos.

### **1.1. Usuario no registrado.**

Este tipo de usuario podrá navegar por la aplicación y utilizar alguna de sus funcionalidades, pero no podrá disfrutar de todo el repertorio ofrecido por la aplicación. En concreto, este tipo de usuario podrá acceder a búsquedas y a visualización de información limitada, pero no podrá compartir contenido, comentar en foros, ni crear ni acceder a menús ni a listas de recetas.

### **1.2. Administrador.**

Este usuario se dedicará a la gestión de las cuentas de usuario, y, podrá administrar los datos almacenados en la aplicación (Eliminar y modificar usuarios, recetas y comentarios con contenido inapropiado).

## **2. Especificación de requisitos**

### **2.1. Gestión de usuario**

REQUISITO 1	REGISTRAR USUARIO
<b>Identificador</b>	1.1
<b>Prioridad</b>	Alta
<b>Precondición</b>	NA
<b>Descripción</b>	Los usuarios pueden crearse una cuenta en la aplicación.
<b>Entrada</b>	Nombre de usuario, Correo Electrónico Contraseña.
<b>Salida</b>	NA
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación y selecciona "Registrarse".</li> <li>2. El sistema muestra un formulario de registro.</li> <li>3. El usuario completa los campos y envía la solicitud.</li> <li>4. El sistema valida la información.</li> <li>5. Se crea la cuenta y el usuario es redirigido a la pantalla de loginl.</li> </ol>
<b>Postcondición</b>	Se ha creado la cuenta de usuario.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. Error si la contraseña no cumple los requisitos.</li> <li>2. Error si el correo electrónico ya está registrado.</li> </ol>
<b>Actores</b>	Usuario no registrado

REQUISITO 2	LOGIN
<b>Identificador</b>	1.2
<b>Prioridad</b>	Alta
<b>Precondición</b>	El usuario debe estar registrado
<b>Descripción</b>	Los usuarios registrados pueden iniciar sesión
<b>Entrada</b>	Nombre de usuario o correo, contraseña
<b>Salida</b>	NA
<b>Secuencia normal</b>	1. El usuario accede a la pantalla de inicio de sesión y completa los datos.
	2. El sistema valida los datos y permite el acceso.
<b>Postcondición</b>	El usuario ha iniciado la sesión correctamente.
<b>Excepciones</b>	Error en el inicio de sesión debido a credenciales incorrectas.
<b>Actores</b>	Usuario registrado, Admin

REQUISITO 3	LOGOUT
<b>Identificador</b>	1.3
<b>Prioridad</b>	Alta
<b>Precondición</b>	El usuario debe haber iniciado sesión
<b>Descripción</b>	Los usuarios que hayan iniciado sesión pueden cerrarla
<b>Entrada</b>	NA
<b>Salida</b>	NA
<b>Secuencia normal</b>	1. El usuario clics sobre "Cerrar sesión" en la barra de navegación superior.
	2. El sistema valida los datos y cierra la sesión.
<b>Postcondición</b>	El usuario ha cerrado la sesión correctamente.

<b>REQUISITO 4</b>	<b>MODIFICAR DATOS</b>
<b>Identificador</b>	1.4
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Descripción</b>	Los usuarios pueden modificar sus datos personales.
<b>Entrada</b>	Nombre de usuario, Correo Electrónico, Contraseña.
<b>Salida</b>	NA
<b>Secuencia normal</b>	1. El usuario accede a su perfil y modifica los campos.
	2. El sistema muestra el formulario con los datos actuales.
	3. El usuario modifica los datos y guarda los cambios.
<b>Postcondición</b>	Los datos del usuario se han actualizado correctamente.
<b>Excepciones</b>	El sistema muestra un error si el correo está en uso o los datos son inválidos.
<b>Actores</b>	Usuario registrado, Admin

<b>REQUISITO 5</b>	<b>DARSE DE BAJA</b>
<b>Identificador</b>	1.4
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Descripción</b>	Los usuarios pueden eliminar su cuenta de la aplicación.
<b>Entrada</b>	ID de usuario.
<b>Salida</b>	NA
<b>Secuencia normal</b>	1. El usuario accede a su perfil y selecciona "Eliminar cuenta".
	2. El sistema solicita confirmación.
	3. El usuario confirma la eliminación de la cuenta.
	4. El sistema elimina los datos del usuario.
<b>Postcondición</b>	La cuenta del usuario se elimina permanentemente.
<b>Excepciones</b>	Error si la cuenta no se puede eliminar por razones de sistema.
<b>Actores</b>	Usuario registrado, Admin

## 2.2. Gestión de recetas

### 2.2.1. FUNCIONALIDADES BÁSICAS

REQUISITO 6	CREAR RECETA
Identificador	2.1.1
Prioridad	Alta
Precondición	El usuario debe haber iniciado sesión.
Descripción	Los usuarios pueden crear nuevas recetas ingresando detalles como título, descripción, ingredientes, y pasos.
Entrada	Título, Descripción, Ingredientes, Pasos, Imágenes.
Salida	Receta creada.
Secuencia normal	<ol style="list-style-type: none"><li>1. El usuario selecciona la opción "Crear receta".</li><li>2. El sistema muestra un formulario para ingresar los detalles de la receta.</li><li>3. El usuario completa el formulario y guarda la receta.</li></ol>
Postcondición	La receta se ha creado correctamente.
Excepciones	Error si los campos obligatorios no están completos o si ocurre un fallo en la subida de imágenes.
Actores	Usuario registrado

<b>REQUISITO 7</b>	<b>MODIFICAR RECETA</b>
<b>Identificador</b>	2.1.2
<b>Prioridad</b>	Alta
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser el propietario de la receta o administrador.
<b>Descripción</b>	Los usuarios pueden modificar las recetas que han creado previamente.
<b>Entrada</b>	Título, Descripción, Ingredientes, Pasos, Imágenes.
<b>Salida</b>	Receta modificada.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona una de sus recetas y accede a la opción de modificarla.</li> <li>2. El sistema muestra los detalles actuales de la receta.</li> <li>3. El usuario modifica los campos deseados y guarda los cambios.</li> </ol>
<b>Postcondición</b>	Los cambios en la receta se han guardado correctamente.
<b>Actores</b>	Usuario registrado, Administrador

<b>REQUISITO 8</b>	<b>PUBLICAR RECETA</b>
<b>Identificador</b>	2.1.3
<b>Prioridad</b>	Alta
<b>Precondición</b>	El usuario debe haber iniciado sesión y haber completado la creación de la receta.
<b>Descripción</b>	Los usuarios pueden publicar sus recetas para que estén disponibles para otros usuarios en la plataforma.
<b>Entrada</b>	ID de receta.
<b>Salida</b>	Receta publicada.
<b>Secuencia normal</b>	1. El usuario selecciona una receta creada y elige la opción de publicarla.
	2. El sistema confirma la publicación de la receta.
<b>Postcondición</b>	La receta está disponible públicamente.
<b>Excepciones</b>	Error si el proceso de publicación falla debido a problemas técnicos.
<b>Actores</b>	Usuario registrado

<b>REQUISITO 9</b>	<b>ELIMINAR RECETA</b>
<b>Identificador</b>	2.1.4
<b>Prioridad</b>	Alta
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser el propietario de la receta o administrador.
<b>Descripción</b>	Los usuarios pueden eliminar permanentemente sus recetas de la plataforma.
<b>Entrada</b>	ID de receta.
<b>Salida</b>	Receta eliminada.
<b>Secuencia normal</b>	1. El usuario accede a su lista de recetas y selecciona la receta que desea eliminar.
	2. El sistema solicita una confirmación de eliminación.
	3. El usuario confirma la eliminación.
<b>Postcondición</b>	La receta se ha eliminado permanentemente.
<b>Excepciones</b>	Error si la receta no se puede eliminar debido a problemas técnicos.

<b>REQUISITO 10</b>	<b>COMPARTIR RECETA</b>
<b>Identificador</b>	2.1.5
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión
<b>Descripción</b>	Los usuarios pueden compartir recetas en redes sociales o mediante enlaces directos copiando el enlace.
<b>Entrada</b>	URL de la receta
<b>Salida</b>	Enlace compartido o receta publicada en redes.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario copia la dirección URL.</li> <li>2. El sistema genera un enlace de la receta o muestra opciones para compartir en redes sociales.</li> </ol>
<b>Postcondición</b>	La receta se comparte en la red o se genera un enlace para compartir.
<b>Excepciones</b>	Error si el enlace no se puede generar o la conexión a la red social falla.
<b>Actores</b>	Usuario registrado

## 2.2.2. INTERACCIÓN CON RECETAS AJENAS

REQUISITO 11	AÑADIR RECETA DE OTRO USUARIO A LA LISTA DE FAVORITOS
Identificador	2.2.1
Prioridad	Media
Precondición	El usuario debe haber iniciado sesión.
Descripción	Los usuarios pueden añadir recetas de otros usuarios a su lista de favoritos para guardarlas y consultarlas más tarde.
Entrada	ID de receta.
Salida	Receta añadida a la lista de favoritos.
Secuencia normal	<ol style="list-style-type: none"><li>1. El usuario selecciona una receta de otro usuario.</li><li>2. El usuario selecciona la opción "Agregar a lista".</li><li>3. El usuario selecciona su lista llamada "favoritos"</li></ol>
Postcondición	La receta se ha añadido a la lista de favoritos del usuario.
Actores	Usuario registrado

<b>REQUISITO 12</b>	<b>Eliminar receta de otro usuario de lista favoritos</b>
<b>Identificador</b>	2.2.2
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y la receta debe estar en su lista de favoritos.
<b>Descripción</b>	Los usuarios pueden eliminar recetas de otros usuarios de su lista de favoritos.
<b>Entrada</b>	ID de receta.
<b>Salida</b>	Receta eliminada de la lista de favoritos.
<b>Secuencia normal</b>	1. El usuario selecciona una receta de una lista.
	2. El usuario selecciona la opción "Eliminar".
	3. El sistema confirma que la receta se ha eliminado de la lista de favoritos del usuario mediante un cuadro de confirmación.
<b>Postcondición</b>	La receta se ha eliminado de la lista de favoritos del usuario.

<b>REQUISITO 13</b>	<b>VISUALIZACIÓN LISTA FAVORITOS</b>
<b>Identificador</b>	2.2.3
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y tener recetas en su lista de favoritos.
<b>Descripción</b>	Los usuarios pueden visualizar las recetas de otros usuarios que han añadido a su lista de favoritos.
<b>Entrada</b>	NA
<b>Salida</b>	Lista de recetas favoritas del usuario.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a su perfil y selecciona la opción "Mis listas de recetas" y selecciona la lista "favoritos".</li> <li>2. El sistema muestra la lista de recetas favoritas del usuario.</li> </ol>
<b>Postcondición</b>	El usuario puede visualizar las recetas en su lista de favoritos.
<b>Excepciones</b>	Error si la lista de favoritos está vacía o si ocurre un problema técnico.
<b>Actores</b>	Usuario registrado

### 2.2.3. BÚSQUEDA Y VISUALIZACIÓN DE RECETAS

REQUISITO 14	VISUALIZAR RECETAS
<b>Identificador</b>	2.3.1
<b>Prioridad</b>	Alta
<b>Precondición</b>	El usuario debe haber accedido a la plataforma, ya sea registrado o no.
<b>Descripción</b>	Los usuarios pueden visualizar las recetas publicadas en la plataforma, viendo sus detalles como ingredientes, preparación y comentarios.
<b>Entrada</b>	Clic sobre la receta
<b>Salida</b>	Detalles de la receta seleccionada.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona una receta desde la lista o los resultados de búsqueda.</li><li>2. El sistema muestra los detalles de la receta seleccionada (ingredientes, pasos de preparación, imágenes, comentarios).</li></ol>
<b>Postcondición</b>	La receta es visualizada por el usuario.
<b>Excepciones</b>	Error si la receta no está disponible o si ocurre un fallo en el sistema.
<b>Actores</b>	Usuario registrado, Usuario no registrado, Administrador

<b>REQUISITO 15</b>	<b>BUSCAR RECETAS</b>
<b>Identificador</b>	2.3.2
<b>Prioridad</b>	Alta
<b>Precondición</b>	NA
<b>Descripción</b>	Los usuarios pueden buscar recetas utilizando filtros como autores, ingredientes o fecha.
<b>Entrada</b>	Criterios de búsqueda (por ejemplo, autor, ingredientes, nombre de la receta).
<b>Salida</b>	Lista de recetas que cumplen con los criterios de búsqueda ingresados.
<b>Secuencia normal</b>	1. El usuario accede a la barra de búsqueda y selecciona "recetas".
	2. El usuario ingresa los filtros o criterios de búsqueda.
	3. El sistema procesa la búsqueda y muestra los resultados correspondientes.
<b>Postcondición</b>	Se muestra una lista de recetas que coinciden con los filtros seleccionados por el usuario.
<b>Excepciones</b>	Error si no se encuentran recetas que coincidan con los criterios de búsqueda o si ocurre un problema técnico.
<b>Actores</b>	Usuario registrado, Usuario no registrado, Administrador

#### 2.2.4. COMENTARIOS EN RECETAS AJENAS

REQUISITO 16	AÑADIR COMENTARIO EN UNA RECETA AJENA
Identificador	2.4.1
Prioridad	Alta
Precondición	El usuario debe haber iniciado sesión.
Descripción	Los usuarios pueden añadir un comentario y/o puntuación (por ejemplo, estrellas o likes) en las recetas de otros usuarios.
Entrada	Comentario, Puntuación (por ejemplo, estrellas o likes).
Salida	Comentario o puntuación añadida a la receta.
Secuencia normal	<ol style="list-style-type: none"><li>1. El usuario selecciona una receta y accede a la sección de comentarios o puntuaciones.</li><li>2. El usuario escribe un comentario o selecciona una puntuación.</li><li>3. El sistema guarda el comentario o puntuación en la receta.</li></ol>
Postcondición	El comentario o puntuación se añade correctamente a la receta.
Excepciones	Error si el comentario no cumple con los requisitos o si ocurre un fallo técnico.
Actores	Usuario registrado

<b>REQUISITO 17</b>	<b>ELIMINAR COMENTARIO EN RECETA</b>
<b>Identificador</b>	2.4.2
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y haber añadido previamente el comentario o puntuación.
<b>Descripción</b>	Los usuarios pueden eliminar un comentario o puntuación que han agregado previamente en las recetas de otros usuarios.
<b>Entrada</b>	Comentario o puntuación (propios) / Administrador puede eliminar cualquiera.
<b>Salida</b>	Comentario o puntuación eliminada de la receta.
<b>Secuencia normal</b>	1. El usuario selecciona la receta y accede a su comentario o puntuación.
	2. El usuario elige la opción de eliminar el comentario o puntuación.
	3. El sistema confirma la eliminación.
<b>Postcondición</b>	El comentario o puntuación se elimina correctamente de la receta.
<b>Excepciones</b>	Error si el comentario no se puede eliminar por razones técnicas o de acceso.
<b>Actores</b>	Usuario registrado y Administrador

<b>REQUISITO 18</b>	<b>MODIFICAR COMENTARIO EN RECETA AJENA</b>
<b>Identificador</b>	2.4.3
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y haber agregado previamente el comentario o puntuación.
<b>Descripción</b>	Los usuarios pueden modificar un comentario o puntuación previamente agregada en las recetas de otros usuarios.
<b>Entrada</b>	Comentario modificado, Puntuación modificada (por ejemplo, estrellas o likes).
<b>Salida</b>	Comentario o puntuación actualizada en la receta.
<b>Secuencia normal</b>	1. El usuario selecciona la receta y accede a su comentario o puntuación.
	2. El usuario elige la opción de modificar el comentario o puntuación.
	3. El sistema guarda los cambios en el comentario o puntuación.
<b>Postcondición</b>	El comentario o puntuación se modifica correctamente.
<b>Excepciones</b>	Error si los cambios no se pueden guardar debido a un problema técnico.
<b>Actores</b>	Usuario registrado, Admin

### 2.2.5. COMENTARIOS EN RECETAS PROPIAS

REQUISITO 19	RESPUESTAS A COMENTARIOS
<b>Identificador</b>	2.5
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y ser el propietario de la receta.
<b>Descripción</b>	Los usuarios pueden responder a comentarios que otros usuarios han hecho en sus propias recetas.
<b>Entrada</b>	Respuesta al comentario.
<b>Salida</b>	Respuesta añadida al comentario de la receta.
<b>Secuencia normal</b>	1. El usuario accede a los comentarios de su receta.
	2. El usuario selecciona un comentario y elige la opción de responder.
	3. El sistema guarda y publica la respuesta bajo el comentario.
<b>Postcondición</b>	La respuesta se añade correctamente al comentario en la receta.
<b>Excepciones</b>	Error si la respuesta no se puede guardar debido a un problema técnico.
<b>Actores</b>	Usuario registrado (propietario de la receta)

### 2.3. LISTAS

REQUISITO 20	CREAR LISTA RECETAS
Identificador	2.6
Prioridad	Media
Precondición	El usuario debe haber iniciado sesión.
Descripción	Los usuarios pueden crear listas personalizadas para agrupar recetas según sus preferencias, como "Favoritas" o "Pendientes".
Entrada	Nombre de la lista de recetas, Recetas seleccionadas para añadir.
Salida	Lista de recetas creada.
Secuencia normal	1. El usuario selecciona la opción "Crear Nueva Receta".
	2. El sistema solicita un nombre para la lista y permite al usuario seleccionar recetas para agregar.
	3. El usuario guarda la lista.
Postcondición	La lista de recetas se ha creado correctamente.
Excepciones	Error si el sistema no puede crear la lista debido a problemas técnicos.
Actores	Usuario registrado

### 2.3.1. OPERACIONES CON LISTAS

REQUISITO 21	AÑADIR RECETA A LISTA
<b>Identificador</b>	2.7.1
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y tener listas de recetas creadas.
<b>Descripción</b>	Los usuarios pueden añadir recetas a sus listas personalizadas de recetas, como "Favoritas" o "Para probar".
<b>Entrada</b>	ID de la receta, Lista de recetas a la que se va a añadir la receta.
<b>Salida</b>	Receta añadida a la lista seleccionada.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona una receta que desea añadir a una lista.</li><li>2. El sistema muestra las listas del usuario.</li><li>3. El usuario elige la lista y añade la receta.</li></ol>
<b>Postcondición</b>	La receta se añade correctamente a la lista seleccionada.
<b>Excepciones</b>	Error si la receta ya está en la lista seleccionada o si ocurre un fallo técnico.
<b>Actores</b>	Usuario registrado

<b>REQUISITO22</b>	<b>ELIMINAR RECETA DE LISTA</b>
<b>Identificador</b>	2.7.2
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y tener listas de recetas creadas.
<b>Descripción</b>	Los usuarios pueden eliminar recetas de sus listas.
<b>Entrada</b>	ID de la receta, Lista de recetas a la que se va a eliminar la receta.
<b>Salida</b>	Receta eliminada de la lista seleccionada.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona una lista de la que desea eliminar una receta</li> <li>2. El sistema muestra las listas del usuario.</li> <li>3. El usuario elige la lista y elimina la receta.</li> <li>4. El sistema muestra confirmación de la receta eliminada.</li> </ol>
<b>Postcondición</b>	La receta se añade correctamente a la lista seleccionada.
<b>Excepciones</b>	Error si la receta ya está en la lista seleccionada o si ocurre un fallo técnico.
<b>Actores</b>	Usuario registrado

<b>REQUISITO 23</b>	<b>MOVER RECETA DE UNA LISTA A OTRA</b>
<b>Identificador</b>	2.7.3
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y tener al menos dos listas de recetas creadas.
<b>Descripción</b>	Los usuarios pueden mover recetas de una lista de recetas a otra dentro de sus listas personalizadas.
<b>Entrada</b>	ID de la receta, Lista de origen, Lista de destino.
<b>Salida</b>	Receta movida de la lista de origen a la lista de destino.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a su lista de recetas personalizada.</li> <li>2. El usuario selecciona una receta y elige la opción de moverla.</li> <li>3. El sistema muestra las listas del usuario.</li> <li>4. El usuario selecciona la lista de destino y la receta se mueve a esa lista.</li> </ol>
<b>Postcondición</b>	La receta se mueve correctamente de una lista a otra.
<b>Excepciones</b>	Error si la receta no puede ser movida debido a un fallo técnico.
<b>Actores</b>	Usuario registrado

## 2.4. MENÚS

<b>REQUISITO 24</b>	<b>VISUALIZAR MENU</b>
<b>Identificador</b>	2.8.1
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Descripción</b>	Los usuarios pueden visualizar los menús creados por otros usuarios en la plataforma.
<b>Entrada</b>	NA
<b>Salida</b>	Lista de menús disponibles.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario accede a la sección de menús.</li><li>2. El sistema muestra una lista de menús disponibles creados por otros usuarios.</li><li>3. El usuario selecciona un menú para visualizar sus detalles.</li></ol>
<b>Postcondición</b>	El usuario puede ver los menús disponibles en la plataforma.
<b>Excepciones</b>	Error si no hay menús disponibles o si ocurre un problema técnico.
<b>Actores</b>	Usuario registrado

<b>REQUISITO 25:</b>	<b>BUSCAR MENÚS</b>
<b>Identificador</b>	2.8.2
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Descripción</b>	Los usuarios pueden buscar menús creados por otros usuarios utilizando filtros como tipo de menú, ingredientes, etc.
<b>Entrada</b>	Nombre del menú buscado
<b>Salida</b>	Lista de menús que cumplen los criterios de búsqueda.
<b>Secuencia normal</b>	1. El usuario accede a la barra de búsqueda en la sección de menús.
	2. El usuario ingresa el nombre del menú que se busca.
	3. El sistema muestra los resultados que coinciden con los filtros ingresados.
<b>Postcondición</b>	El usuario puede ver una lista de menús que coinciden con los criterios de búsqueda seleccionados.
<b>Excepciones</b>	Error si no se encuentran menús que coincidan con los criterios de búsqueda o si ocurre un problema técnico.
<b>Actores</b>	Usuario registrado

### 2.4.1. INTERACCIÓN CON MENÚS

REQUISITO 26	CREAR MENÚ AGRUPANDO RECETAS
Identificador	2.9.1
Prioridad	Media
Precondición	El usuario debe haber iniciado sesión.
Descripción	Los usuarios pueden crear menús agrupando diferentes recetas, como menús semanales o menús para ocasiones especiales.
Entrada	Nombre del menú, Recetas seleccionadas para incluir en el menú.
Salida	Menú creado.
Secuencia normal	<ol style="list-style-type: none"><li>1. El usuario selecciona la opción "Crear menú".</li><li>2. El sistema solicita un nombre para el menú y permite al usuario seleccionar recetas para incluir en él.</li><li>3. El usuario guarda el menú.</li></ol>
Postcondición	El menú se ha creado correctamente.
Excepciones	Error si el sistema no puede crear el menú debido a problemas técnicos.
Actores	Usuario registrado

<b>REQUISITO 27</b>	<b>ELIMINAR MENU AGRUPANDO RECETAS</b>
<b>Identificador</b>	2.9.2
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y tener un menú previamente creado.
<b>Descripción</b>	Los usuarios pueden eliminar menús previamente creados que agrupan recetas.
<b>Entrada</b>	ID del menú.
<b>Salida</b>	Menú eliminado.
<b>Secuencia normal</b>	1. El usuario accede a su lista de menús creados.
	2. El usuario selecciona el menú a eliminar.
	3. El sistema confirma la eliminación del menú.
<b>Postcondición</b>	El menú se elimina correctamente.
<b>Excepciones</b>	Error si el menú no se puede eliminar debido a problemas técnicos.
<b>Actores</b>	Usuario registrado

<b>REQUISITO 28</b>	<b>MODIFICAR MENÚ AGRUPANDO RECETAS</b>
<b>Identificador</b>	2.9.3
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y tener un menú previamente creado.
<b>Descripción</b>	Los usuarios pueden modificar los menús previamente creados, agregando o eliminando recetas o modificando detalles del menú.
<b>Entrada</b>	ID del menú, Recetas modificadas, Detalles modificados del menú.
<b>Salida</b>	Menú modificado.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a su lista de menús creados.</li> <li>2. El usuario selecciona el menú que desea modificar.</li> <li>3. El sistema permite al usuario editar recetas o modificar detalles del menú.</li> <li>4. El usuario guarda los cambios.</li> </ol>
<b>Postcondición</b>	El menú se modifica correctamente.
<b>Excepciones</b>	Error si el sistema no permite modificar el menú debido a problemas técnicos.
<b>Actores</b>	Usuario registrado

<b>REQUISITO 29</b>	<b>PUBLICAR MENÚ</b>
<b>Identificador</b>	2.9.4
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y tener un menú previamente creado.
<b>Descripción</b>	Los usuarios pueden publicar menús creados, agrupando recetas, para que sean visibles para otros usuarios.
<b>Entrada</b>	ID del menú.
<b>Salida</b>	Menú publicado.
<b>Secuencia normal</b>	1. El usuario selecciona un menú previamente creado y elige la opción de publicarlo.
	2. El sistema confirma la publicación del menú.
<b>Postcondición</b>	El menú está disponible públicamente para otros usuarios.
<b>Excepciones</b>	Error si el menú no se puede publicar debido a problemas técnicos.
<b>Actores</b>	Usuario registrado

<b>REQUISITO 30</b>	<b>COMPARTIR MENÚ</b>
<b>Identificador</b>	2.9.5
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y tener un menú previamente creado o publicado.
<b>Descripción</b>	Los usuarios pueden compartir menús agrupando recetas, en redes sociales o mediante enlaces directos.
<b>Entrada</b>	URL del menú
<b>Salida</b>	Enlace compartido o menú publicado en redes sociales.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario copia la URL del menú.</li> <li>2. El sistema genera un enlace del menú o muestra opciones para compartir en redes sociales.</li> </ol>
<b>Postcondición</b>	El menú se comparte en la red o se genera un enlace para compartir.
<b>Excepciones</b>	Error si el enlace no se puede generar o la conexión a la red social falla.
<b>Actores</b>	Usuario registrado

## 2.5. FOROS

<b>REQUISITO 31:</b>	<b>CREAR FORO PÚBLICO</b>
<b>Identificador</b>	2.10
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Descripción</b>	Los usuarios pueden crear foros públicos para discutir sobre una receta o un menú específico en la plataforma.
<b>Entrada</b>	Título del foro y contenido
<b>Salida</b>	Hilo público creado
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario accede a una receta o menú y selecciona la opción de crear un foro público.</li><li>2. El sistema solicita un nombre y una descripción del foro.</li><li>3. El usuario rellena los campos y crea el foro.</li></ol>
<b>Postcondición</b>	El foro público se ha creado correctamente y está disponible para otros usuarios.
<b>Excepciones</b>	Error si el foro no se puede crear por problemas técnicos o falta de permisos.
<b>Actores</b>	Usuario registrado

### 2.5.1. INTERACCIÓN CON FOROS

<b>REQUISITO 32</b>	<b>AÑADIR INTERVENCION FORO</b>
<b>Identificador</b>	2.11.1
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y estar participando en un foro.
<b>Descripción</b>	Los usuarios pueden añadir intervenciones (comentarios) en un foro, ya sea un foro propio o en el de otros usuarios.
<b>Entrada</b>	Contenido de la intervención (comentario).
<b>Salida</b>	Intervención añadida al foro.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario accede a un foro propio o de otro usuario.</li><li>2. El usuario redacta su intervención.</li><li>3. El sistema guarda y publica la intervención en el foro.</li></ol>
<b>Postcondición</b>	La intervención se publica correctamente en el foro.

<b>REQUISITO 33</b>	<b>ELIMINAR INTERVENCIÓN FORO</b>
<b>Identificador</b>	2.11.2
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y haber hecho la intervención en el foro.
<b>Descripción</b>	Los usuarios pueden eliminar sus propias intervenciones de un foro, ya sea un foro propio o en el de otros usuarios.
<b>Entrada</b>	ID de la intervención.
<b>Salida</b>	Intervención eliminada del foro.
<b>Secuencia normal</b>	1. El usuario accede al foro y selecciona su intervención.
	2. El usuario elige la opción de eliminar la intervención.
	3. El sistema confirma y elimina la intervención del foro.
<b>Postcondición</b>	La intervención se elimina correctamente del foro.
<b>Excepciones</b>	Error si no se puede eliminar la intervención debido a problemas técnicos.
<b>Actores</b>	Usuario registrado

<b>REQUISITO 34</b>	<b>MODIFICAR INTERVENCIÓN FORO</b>
<b>Identificador</b>	2.11.3
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y haber hecho la intervención en el foro.
<b>Descripción</b>	Los usuarios pueden modificar sus propias intervenciones en un foro, ya sea en un foro propio o en el de otros usuarios.
<b>Entrada</b>	Contenido modificado de la intervención (comentario).
<b>Salida</b>	Intervención modificada en el foro.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede al foro y selecciona su intervención.</li> <li>2. El usuario elige la opción de modificar la intervención.</li> <li>3. El sistema guarda y publica la intervención modificada.</li> </ol>
<b>Postcondición</b>	La intervención se modifica correctamente en el foro.
<b>Excepciones</b>	Error si la intervención no se puede modificar debido a problemas técnicos.
<b>Actores</b>	Usuario registrado

## 2.6. SOCIAL

REQUISITO 35	SEGUIR A UN USUARIO
<b>Identificador</b>	2.12.1
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Descripción</b>	Los usuarios pueden seguir a otros usuarios de la plataforma para recibir actualizaciones sobre sus actividades y publicaciones.
<b>Entrada</b>	ID del usuario a seguir.
<b>Salida</b>	Usuario seguido.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario busca o selecciona un perfil de otro usuario en la plataforma.</li><li>2. El usuario selecciona la opción "Seguir".</li><li>3. El sistema confirma que el usuario está ahora siguiendo al otro usuario.</li></ol>
<b>Postcondición</b>	El usuario empieza a recibir actualizaciones del usuario seguido.
<b>Excepciones</b>	Error si el sistema no puede procesar la solicitud de seguimiento debido a problemas técnicos.
<b>Actores</b>	Usuario registrado

### 2.6.1. ACTUALIZACIONES EN EL SEGUIMIENTO

REQUISITO 37	ACTUALIZACIONES SEGUIMIENTO
<b>Identificador</b>	2.13
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión y estar siguiendo a uno o más usuarios en la plataforma.
<b>Descripción</b>	Los usuarios pueden ver las actualizaciones recientes (nuevas recetas, menús o actividades) de los usuarios que siguen.
<b>Entrada</b>	NA
<b>Salida</b>	Lista de actualizaciones de los usuarios seguidos.
<b>Secuencia normal</b>	1. El usuario accede a la sección de "Recetas".
	2. El sistema muestra las recetas publicadas por usuarios a los que se sigue
	3. El usuario selecciona una receta para verla en detalle.
<b>Postcondición</b>	El usuario visualiza las actividades recientes de los usuarios seguidos.
<b>Excepciones</b>	Error si no hay actualizaciones disponibles o si ocurre un problema técnico.
<b>Actores</b>	Usuario registrado

## 2.7. RECOMENDACIONES

REQUISITO 38	VER MIS RECOMENDACIONES
<b>Identificador</b>	2.14.1
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Descripción</b>	Los usuarios pueden ver las recomendaciones de recetas o menús en la pantalla principal de la web.
<b>Entrada</b>	ID de las recomendaciones compartidas.
<b>Salida</b>	Lista de recomendaciones compartidas por la página.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario accede a la página principal.</li><li>2. El sistema muestra una lista de las recomendaciones al final de la página.</li><li>3. El usuario selecciona una recomendación para verla en detalle.</li></ol>
<b>Postcondición</b>	El usuario visualiza las recomendaciones recibidas.
<b>Excepciones</b>	Error si no hay recomendaciones disponibles o si ocurre un problema técnico.
<b>Actores</b>	Usuario registrado

## 2.8. Administración

### 2.8.1. INTERACCIÓN CON USUARIOS

REQUISITO 41	ELIMINAR UN USUARIO
<b>Identificador</b>	3.1.1
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Descripción</b>	Los administradores pueden eliminar usuarios de la plataforma si se da la necesidad
<b>Entrada</b>	ID del usuario a eliminar
<b>Salida</b>	Usuario eliminado
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario busca o selecciona un perfil de otro usuario en la plataforma.</li><li>2. El usuario selecciona la opción "Eliminar"</li><li>3. El usuario realiza confirmación de la eliminación</li><li>4. El sistema confirma que tanto el usuario como sus publicaciones han sido eliminadas</li></ol>
<b>Postcondición</b>	El usuario y el contenido publicado por el mismo son borrados de la aplicación
<b>Excepciones</b>	Error en el caso de que no se tengan permisos para eliminar el usuario seleccionado
<b>Actores</b>	Administración

REQUISITO 42	MODIFICAR UN USUARIO
<b>Identificador</b>	3.1.2
<b>Prioridad</b>	Baja
<b>Precondición</b>	El usuario debe haber iniciado sesión.
<b>Descripción</b>	Los administradores pueden modificar usuarios de la plataforma si se da la necesidad
<b>Entrada</b>	ID del usuario a modificar
<b>Salida</b>	Los datos del usuario se han actualizado correctamente.
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li data-bbox="446 756 1378 850">1. El usuario busca o selecciona un perfil de otro usuario en la plataforma.</li> <li data-bbox="446 850 1378 924">2. El usuario selecciona la opción "Modificar".</li> <li data-bbox="446 924 1378 1031">3. El sistema confirma que el usuario seleccionado ha sido modificado</li> </ol>
<b>Postcondición</b>	El sistema muestra un error si el correo está en uso o los datos son inválidos, o no se tienen permisos para modificar el usuario
<b>Excepciones</b>	Error en el caso de que no se tengan permisos para eliminar el usuario seleccionado

## 2.8.2. ELIMINACIÓN DE CONTENIDO INAPROPIADO

REQUISITO 43	ELIMINAR RECETA INAPROPIADA
Identificador	3.2.1
Prioridad	Media
Precondición	El usuario debe haber iniciado sesión
Descripción	Los administradores pueden eliminar recetas publicadas por otros usuarios si se dan las circunstancias
Entrada	ID de receta.
Salida	Receta eliminada.
Secuencia normal	1. El usuario busca y selecciona la receta que desea eliminar.
	2. El sistema solicita una confirmación de eliminación.
	3. El usuario confirma la eliminación.
Postcondición	La receta se ha eliminado permanentemente.
Excepciones	Error si la receta no se puede eliminar debido a problemas técnicos.
Actores	Administración

<b>REQUISITO 44</b>	<b>ELIMINAR COMENTARIO INAPROPIADO</b>
<b>Identificador</b>	3.2.2
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión
<b>Descripción</b>	Los administradores pueden eliminar comentarios publicados por otros usuarios si se dan las circunstancias
<b>Entrada</b>	Comentario para eliminar
<b>Salida</b>	Comentario eliminado
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario busca y selecciona el comentario que desea eliminar.</li> <li>2. El sistema solicita una confirmación de eliminación.</li> <li>3. El usuario confirma la eliminación.</li> </ol>
<b>Postcondición</b>	El comentario se ha eliminado permanentemente.
<b>Excepciones</b>	Error si el comentario no se puede eliminar debido a problemas técnicos.
<b>Actores</b>	Administración

<b>REQUISITO 45</b>	<b>ELIMINAR FORO INAPROPIADO</b>
<b>Identificador</b>	3.2.3
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión
<b>Descripción</b>	Los administradores pueden eliminar foros publicados por otros usuarios si se dan las circunstancias
<b>Entrada</b>	Foro para eliminar
<b>Salida</b>	Foro eliminado
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario busca y selecciona el foro que desea eliminar.</li> <li>2. El sistema solicita una confirmación de eliminación.</li> <li>3. El usuario confirma la eliminación.</li> </ol>
<b>Postcondición</b>	El foro se ha eliminado permanentemente.
<b>Excepciones</b>	Error si el foro no se puede eliminar debido a problemas técnicos.
<b>Actores</b>	Administración

<b>REQUISITO 46</b>	<b>ELIMINAR LISTA INAPROPIADA</b>
<b>Identificador</b>	3.2.4
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión
<b>Descripción</b>	Los administradores pueden eliminar listas publicadas por otros usuarios si se dan las circunstancias
<b>Entrada</b>	Lista para eliminar
<b>Salida</b>	Lista eliminada
<b>Secuencia normal</b>	1. El usuario busca y selecciona la lista que desea eliminar.
	2. El sistema solicita una confirmación de eliminación.
	3. El usuario confirma la eliminación.
<b>Postcondición</b>	La lista se ha eliminado permanentemente.
<b>Excepciones</b>	Error si la lista no se puede eliminar debido a problemas técnicos.
<b>Actores</b>	Administración

<b>REQUISITO 47</b>	<b>ELIMINAR MENÚ INAPROPIADO</b>
<b>Identificador</b>	3.2.5
<b>Prioridad</b>	Media
<b>Precondición</b>	El usuario debe haber iniciado sesión
<b>Descripción</b>	Los administradores pueden eliminar menús publicados por otros usuarios si se dan las circunstancias
<b>Entrada</b>	Lista para eliminar
<b>Salida</b>	Menú eliminado
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario busca y selecciona el menú que desea eliminar.</li> <li>2. El sistema solicita una confirmación de eliminación.</li> <li>3. El usuario confirma la eliminación.</li> </ol>
<b>Postcondición</b>	El menú se ha eliminado permanentemente.
<b>Excepciones</b>	Error si el menú no se puede eliminar debido a problemas técnicos.
<b>Actores</b>	Administración

# 11. Anexo B: Manual de usuario

Esta es la página de inicio de la aplicación, diseñada para dar una cálida bienvenida al usuario y presentarle de forma concisa lo que puede encontrar. La interfaz es limpia y fácil de navegar, con una disposición que prioriza la información clave y la interacción. En la barra de navegación superior, el usuario "Vicen" ha iniciado sesión y puede acceder a "Inicio", "Recetas", "Foro" y "Menús". El mensaje principal invita a "Explorar Recetas" mediante un botón verde, que es la acción principal para descubrir contenido y más abajo se encuentra la sección "Recetas Destacadas" donde se muestra tres ejemplos visuales: "Ensalada Fresca", "Smoothie Energético" y "Sopa Detox". Cada una incluye una imagen, título, descripción y un botón "Ver Receta" verde para acceder a los detalles.

Se comienza con el caso práctico de crear una receta, para lo que se pulsa sobre Recetas, en la barra superior de exploración.

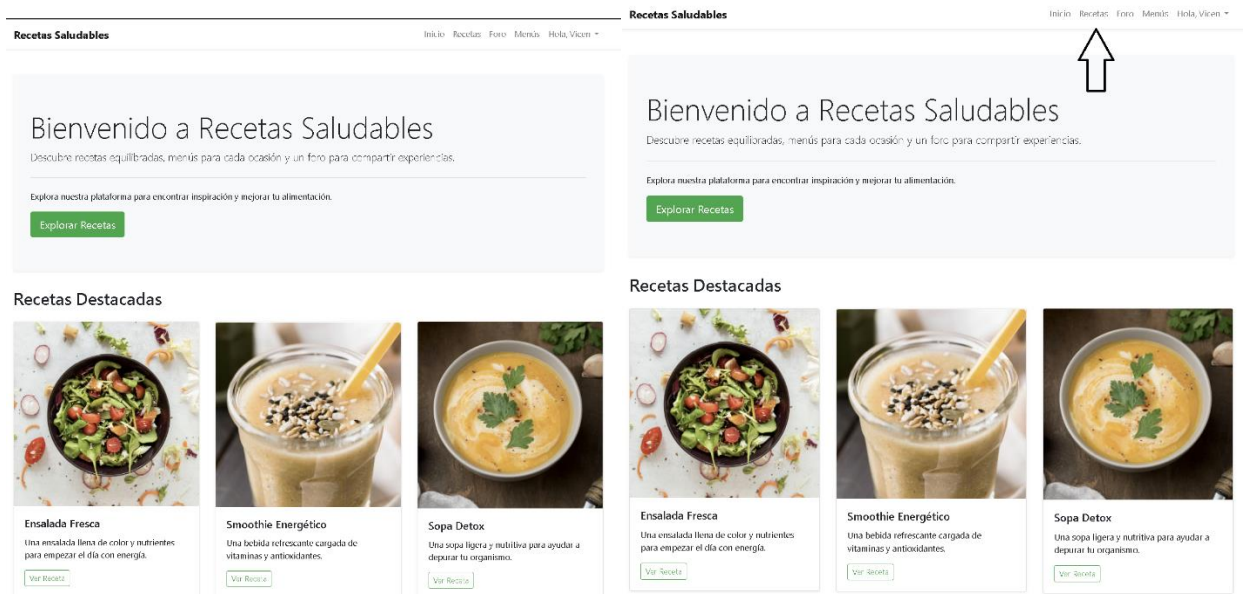


Ilustración 13. Creación de receta (1)

De la página anterior se pasa a la siguiente, donde se muestran todas las recetas disponibles y permite al usuario añadir nuevas, el elemento clave para el caso práctico es el botón verde "Crear Receta" que inicia el proceso para añadir una nueva receta. Cabe destacar que justo debajo hay filtros de búsqueda (por nombre, autor, ingrediente y fecha) para encontrar recetas específicas.

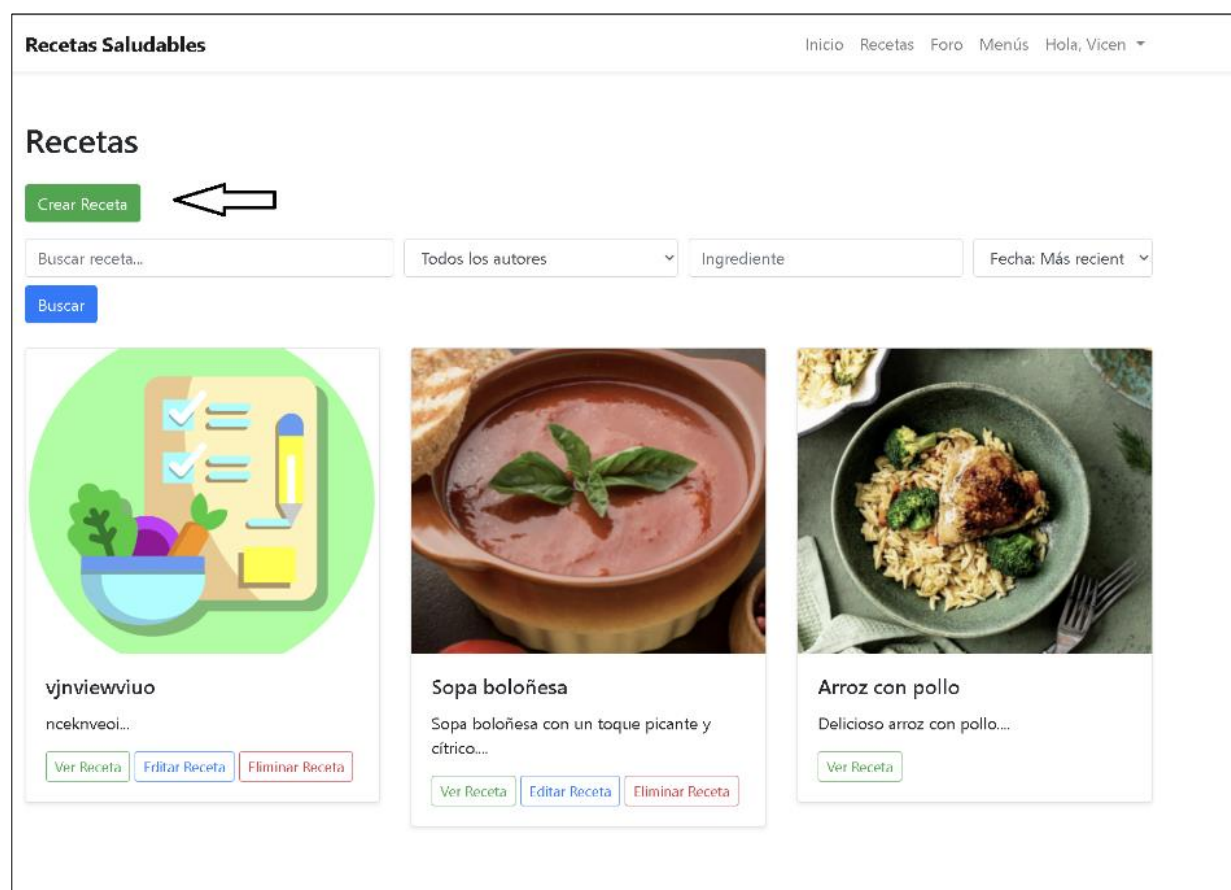


Ilustración 14. Creación de receta (2)

A continuación de que el usuario haya pulsado en el botón verde anterior, se accede a una nueva pantalla donde se deberá introducir los datos y toda la información posible a cerca de la receta (Título, descripción, ingredientes, imagen) y una finalizado el proceso de pulsa en Guardar Receta

**Recetas Saludables** Inicio Recetas Foro Menús Hola, Vicen

## Crear Receta

Título

Risotto

Descripción / Instrucciones

El arroz se cocina primero brevemente en un sofrito de cebolla y mantequilla o aceite de oliva, para cubrir cada grano en una película de grasa, llamada "sofritado". Se agrega vino blanco y debe ser absorbido por los granos. Cuando se ha absorbido, el calor se eleva a medio alto y el caldo hirviendo se agrega gradualmente en pequeñas cantidades, mientras se agita constantemente. La agitación constante, con solo una pequeña cantidad de líquido presente, obliga a los granos a frotarse entre sí y liberar las moléculas de almidón desde el exterior de los granos hacia el líquido circundante, creando una masa suave de textura cremosa. Cuando el arroz está cocido, la olla se retira del fuego para "mantecarlo", batiéndola vigorosamente en bolitas refrigeradas de queso parmesano rallado y mantequilla, para hacer que la textura sea lo más cremosa y suave posible. Puede retirarse del calor unos minutos antes y dejar que se cocine con el calor residual.

Ingredientes

Arroz, champiñones, vino blanco, ajo, queso, cebolla.

Nombre de la Imagen

risotto.png

Las imágenes se almacenan en /static/risotto.png

Publicar inmediatamente

[Guardar Receta](#) ←

Ilustración 15. Creación de receta (3)

Una vez se ha guardado, aparecerá la nueva receta donde estabas las anteriores que ya han sido creadas. En este caso la receta creada en el paso anterior es Risotto y como se puede apreciar aparece en primera posición.


**Recetas Saludables** Inicio Recetas Foro Menús Hola, Vicen

## Recetas

[Crear Receta](#)

Buscar receta...  Todos los autores  Ingrediente  Fecha: Más recent


[Buscar](#)



**Risotto**

El arroz se cocina primero brevemente en un sofrito de cebolla y mantequilla o aceite de oliva, para...


[Ver Receta](#) [Editar Receta](#) [Eliminar Receta](#)



vjnvievwio

ncknveoi...

[Ver Receta](#) [Editar Receta](#) [Eliminar Receta](#)



**Sopa boloñesa**

Sopa boloñesa con un toque picante y cítrico...

[Ver Receta](#) [Editar Receta](#) [Eliminar Receta](#)

Ilustración 16. Nueva receta creada.

Ahora otro usuario distinto al creador inicia su sesión para realizar un comentario sobre la receta anterior, el nuevo usuario es Prueba.

**Recetas Saludables** Inicio Recetas Foro Menús Hola, Prueba

# Bienvenido a Recetas Saludables

Descubre recetas equilibradas, menús para cada ocasión y un foro para compartir experiencias.


Explora nuestra plataforma para encontrar inspiración y mejorar tu alimentación.

[Explorar Recetas](#)

Ilustración 17. Acceso por otro usuario.

Este usuario Prueba accederá a la receta que deseé en este caso se hará con la receta creada anteriormente (risotto), para realizar un comentario.

**Risotto**



Creada el: 2025-05-20 10:45:41 Vicen [Dejar de Seguir](#)

**Descripción / Instrucciones:**

El arroz se cocina primero brevemente en un sofrito de cebolla y mantequilla o aceite de oliva, para cubrir cada grano en una película de grasa, llamada "tostatura". Se agrega vino blanco y debe ser absorbido por los granos. Cuando se ha absorbido, el calor se eleva a medio alto y el caldo hirviendo se agrega gradualmente en pequeñas cantidades, mientras se agita constantemente. La agitación constante, con solo una pequeña cantidad de líquido presente, obliga a los granos a frotarse entre sí y liberar las moléculas de almidón desde el exterior de los granos hacia el líquido circundante, creando una masa suave de textura cremosa.

Cuando el arroz está cocido, la olla se retira del fuego para "mantecatura", batiéndola vigorosamente en bollos refrigerados de queso parmesano rallado y mantequilla, para hacer que la textura sea lo más cremosa y suave posible. Puede retirarse del calor unos minutos antes y dejar que se cocine con el calor residual.

**Ingredientes:**

Arroz, champiñones, vino blanco, ajo, queso, cebolla.

[Volver a Recetas](#) [Agregar a Lista](#)

**Comentarios**

Valoración promedio: 0 estrellas

Mínimo Rating: Todos Ordenar por: Más recientes [Filtrar](#)

No hay comentarios para esta receta.

**Añadir Comentario**

Comentario:

Muy buena receta! Me encanta.

Valoración:


★★★★★

[Enviar Comentario](#)

Ilustración 18. Comentario de otro usuario.

Tras su publicación, el comentario se podrá editar o eliminar tal y como puede verse en la imagen:

**Risotto**



Creada el: 2025-05-20 10:45:41 Vicen [Dejar de Seguir](#)

**Descripción / Instrucciones:**

El arroz se cocina primero brevemente en un sofrito de cebolla y mantequilla o aceite de oliva, para cubrir cada grano en una película de grasa, llamada "tostatura". Se agrega vino blanco y debe ser absorbido por los granos. Cuando se ha absorbido, el calor se eleva a medio alto y el caldo hirviendo se agrega gradualmente en pequeñas cantidades, mientras se agita constantemente. La agitación constante, con solo una pequeña cantidad de líquido presente, obliga a los granos a frotarse entre sí y liberar las moléculas de almidón desde el exterior de los granos hacia el líquido circundante, creando una masa suave de textura cremosa.

Cuando el arroz está cocido, la olla se retira del fuego para "mantecatura", batiéndola vigorosamente en bollos refrigerados de queso parmesano rallado y mantequilla, para hacer que la textura sea lo más cremosa y suave posible. Puede retirarse del calor unos minutos antes y dejar que se cocine con el calor residual.

**Ingredientes:**

Arroz, champiñones, vino blanco, ajo, queso, cebolla.

[Volver a Recetas](#) [Agregar a Lista](#)

**Comentarios**

Valoración promedio: 5 estrellas

Mínimo Rating: Todos Ordenar por: Más recientes [Filtrar](#)

**Prueba**

2025-05-20 10:50:27

★★★★★

Muy buena receta! Me encanta.

[Editar](#) [Eliminar](#)

Ilustración 19. Edición o eliminación de comentario.

Ahora, desde el mismo usuario anterior de Prueba se decide entrar en el foro, accediendo a el desde la barra de navegación superior. Una vez dentro del foro, se puede observar el elemento clave el **botón verde "Nuevo Mensaje"**. Este botón es el punto de entrada para que el usuario pueda iniciar una nueva discusión o publicación en el foro.

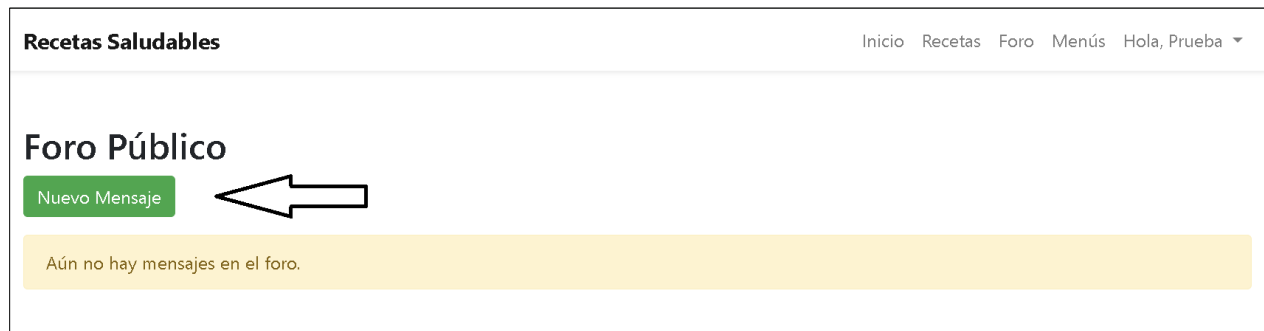


Ilustración 20. Entrada en el foro.

Una vez el usuario ha pulsado en el botón verde anterior, se le dirigirá a una nueva pantalla donde deberá rellenar la información necesaria para llevar a cabo la publicación de dicho mensaje en el foro (Título y contenido) y pulsará en Enviar Mensaje.

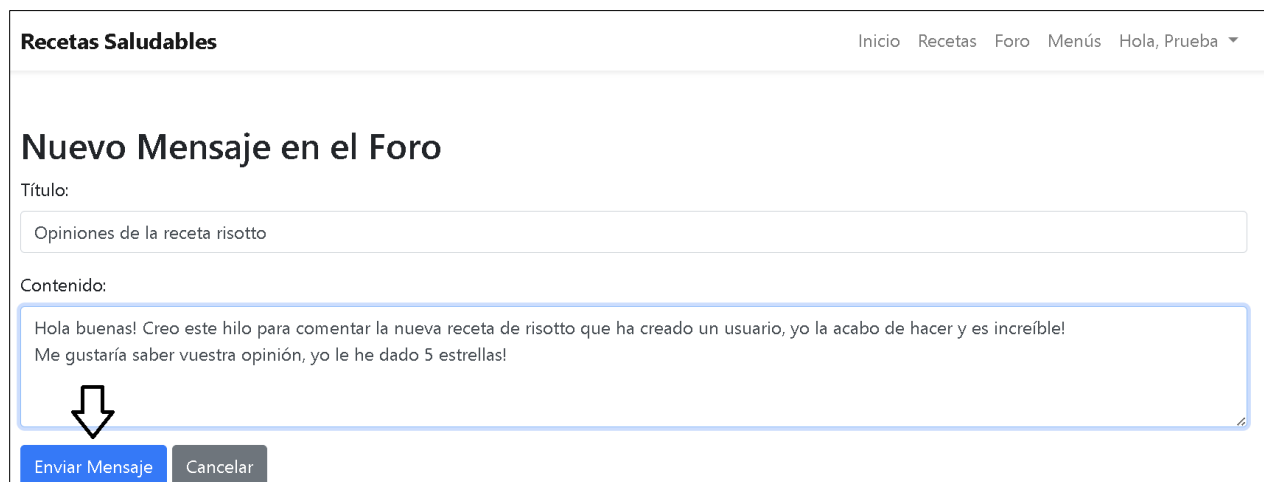


Ilustración 21. Nuevo mensaje en el foro.

A continuación, ya estará disponible dicho mensaje en el foro.



Ilustración 22. Mensaje disponible en el foro.

En la siguiente captura, se muestra la interacción dentro del "Foro Público", específicamente un "Hilo de Mensajes", siendo el espacio donde los usuarios pueden entablar conversaciones, compartir opiniones y generar comunidad alrededor de temas relacionados con las recetas y la alimentación saludable. En esta situación, el usuario Vicen responde al usuario Prueba que es quien escribió el primer mensaje.



Ilustración 23. interacción en el foro.

Se pulsa en Enviar respuesta y aparecerá unido al mensaje principal creando un hilo.



Ilustración 24. Creación de un hilo.

La siguiente captura retorna a la sección general de "Recetas", la misma que se observó en el proceso de "Crear Receta". Sin embargo, ahora el enfoque es diferente: **añadir una receta existente a una lista personalizada de recetas**. Para lo cual se debe introducir en la receta que se quiera añadir

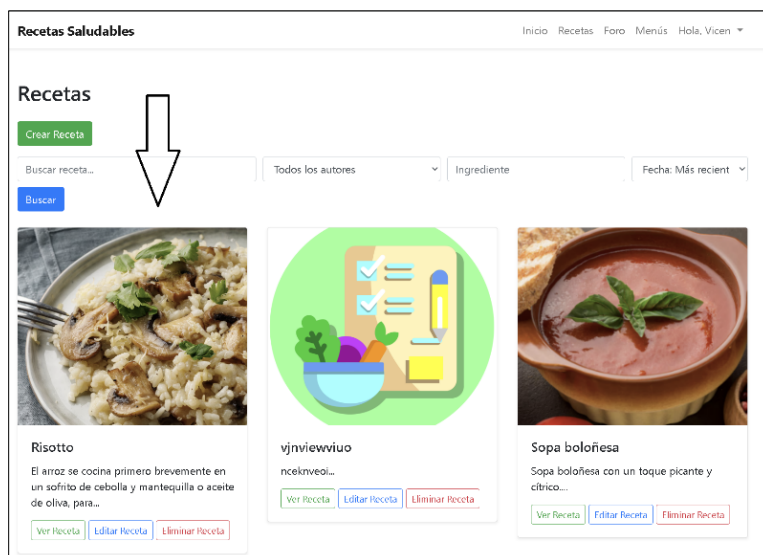
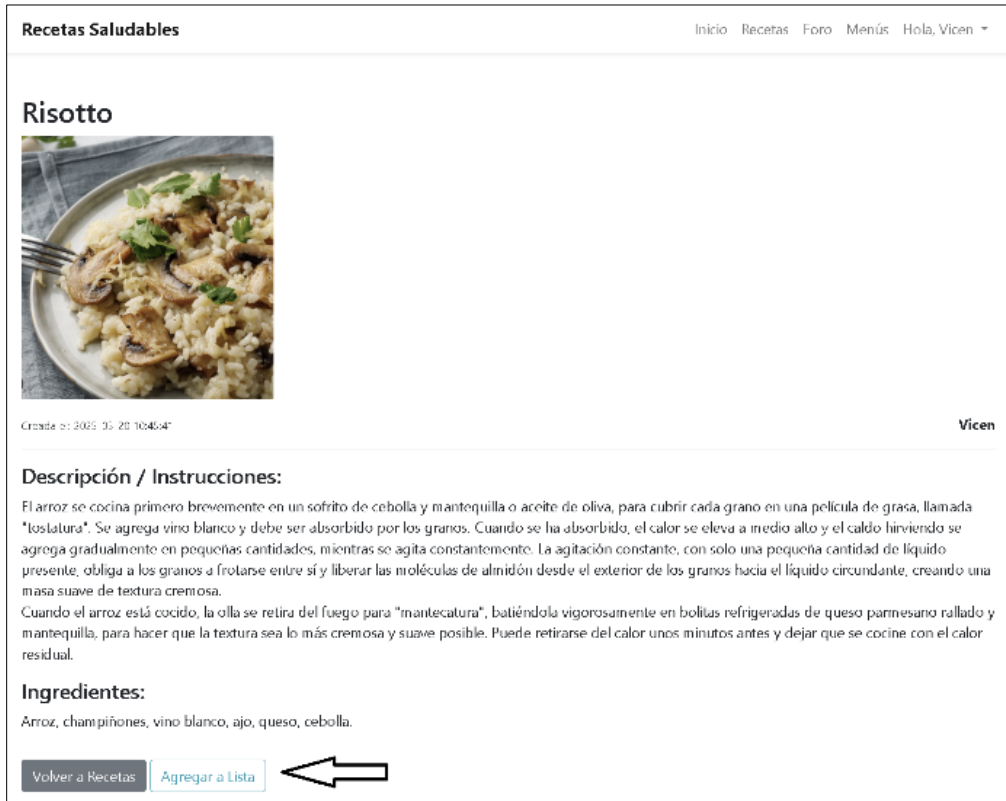



Ilustración 25. Añadir receta existente a lista personalizada (1)

Una vez el usuario se ha introducido en la receta que desea añadir, tiene la opción debajo de Agrega a la lista y posteriormente deberá seleccionar la lista en la que decida añadirla, en este caso se optó por la lista: "favs"



**Recetas Saludables** Inicio Recetas Foro Menús Hola, Vicen ▾

## Risotto



Creada el: 2025-05-20 16:45:44 Vicen

**Descripción / Instrucciones:**

El arroz se cocina primero brevemente en un sofrito de cebolla y mantequilla o aceite de oliva, para cubrir cada grano en una película de grasa, llamada "lostatura". Se agrega vino blanco y debe ser absorbido por los granos. Cuando se ha absorbido, el calor se eleva a medio alto y el caldo hirviendo se agrega gradualmente en pequeñas cantidades, mientras se agita constantemente. La agitación constante, con solo una pequeña cantidad de líquido presente, obliga a los granos a frotarse entre sí y liberar las moléculas de almidón desde el exterior de los granos hacia el líquido circundante, creando una masa suave de textura cremosa.

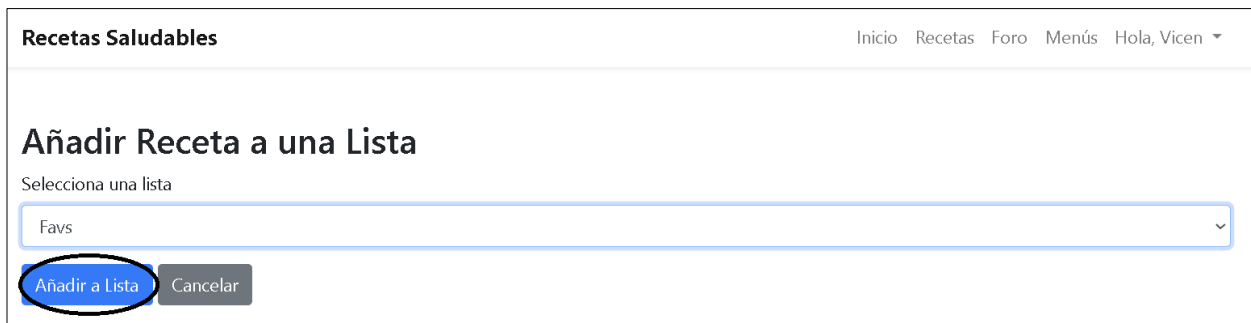
Cuando el arroz está cocido, la olla se retira del fuego para "mantecatura", batiéndola vigorosamente en bolitas refrigeradas de queso parmesano rallado y mantequilla, para hacer que la textura sea lo más cremosa y suave posible. Puede retirarse del calor unos minutos antes y dejar que se cocine con el calor residual.

**Ingredientes:**

Arroz, champiñones, vino blanco, ajo, queso, cebolla.

[Volver a Recetas](#) [Agregar a Lista](#) ←

Ilustración 26. Añadir receta existente a lista de favoritos (2). Selección de lista.



**Recetas Saludables** Inicio Recetas Foro Menús Hola, Vicen ▾

## Añadir Receta a una Lista

Selecciona una lista

Favs ▾

[Añadir a Lista](#) [Cancelar](#)

Ilustración 27. Añadir receta existente a lista de favoritos (3). Confirmación de lista.

Para comprobar que la receta se ha añadido efectivamente en la lista, se volverá de nuevo a la interfaz inicial de la web y se selecciona la opción de Mis listas de Recetas como se muestra en la captura de a continuación:

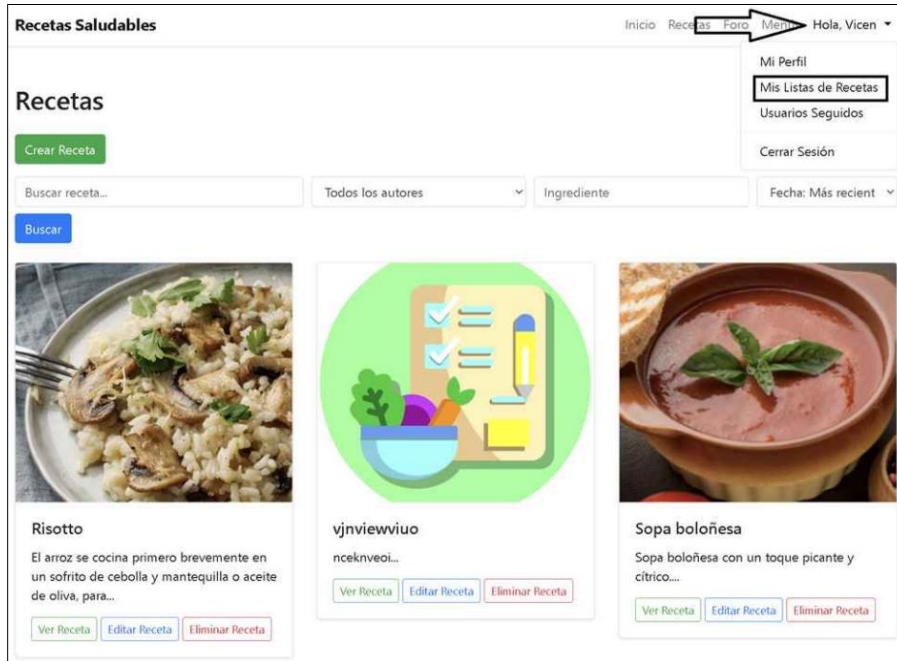


Ilustración 28. Comprobación (1).

Una vez dentro, es posible visualizar las listas de recetas que existen actualmente y entrar en la de "favs", para efectivamente comprobar que la receta "risotto" que se ha añadido ha quedado correctamente instaurada en dicha lista.

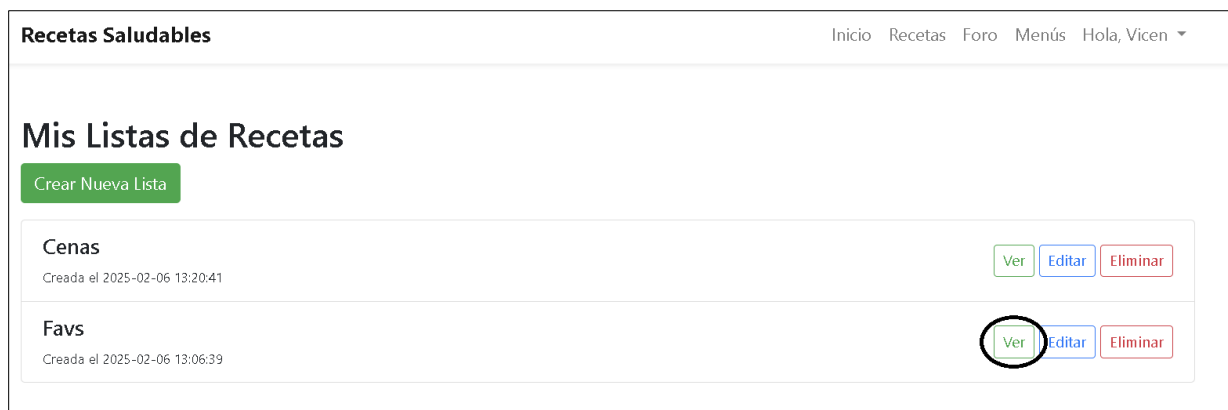


Ilustración 29. Comprobación (2)

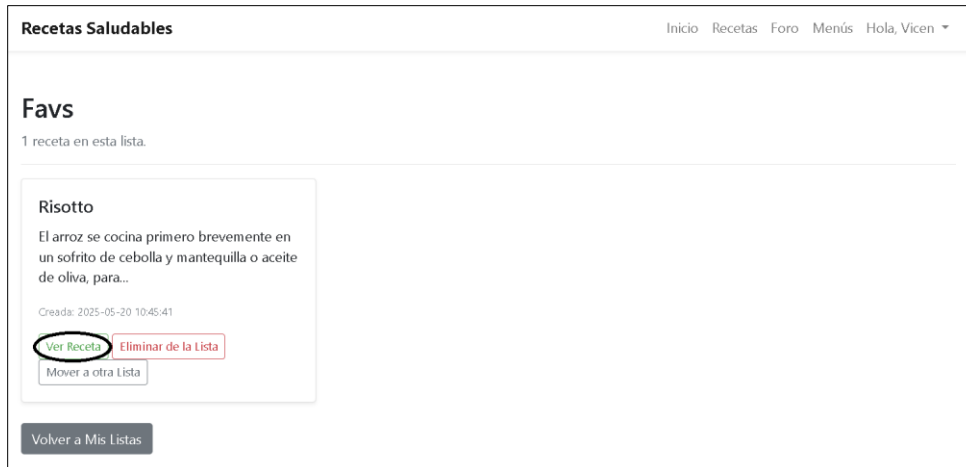


Ilustración 30. Comprobación 3.

En la siguiente imagen se muestra nuevamente la página de inicio de la aplicación "Recetas Saludables", la misma que analizamos al principio. Sin embargo, en esta ocasión, el objetivo es destacar cómo el usuario accederá a la funcionalidad de "Menús".



Ilustración 31. Acceso a "Menús".

La captura siguiente muestra la página principal de la sección "Menús", a la que el usuario accede tras hacer clic en la opción "Menús" de la barra de navegación. Este es el centro de gestión para explorar y crear planes de comidas, se destaca el **botón verde "Crear Menú"**, resaltado por una flecha, que permite al usuario empezar a diseñar un nuevo plan de comidas.

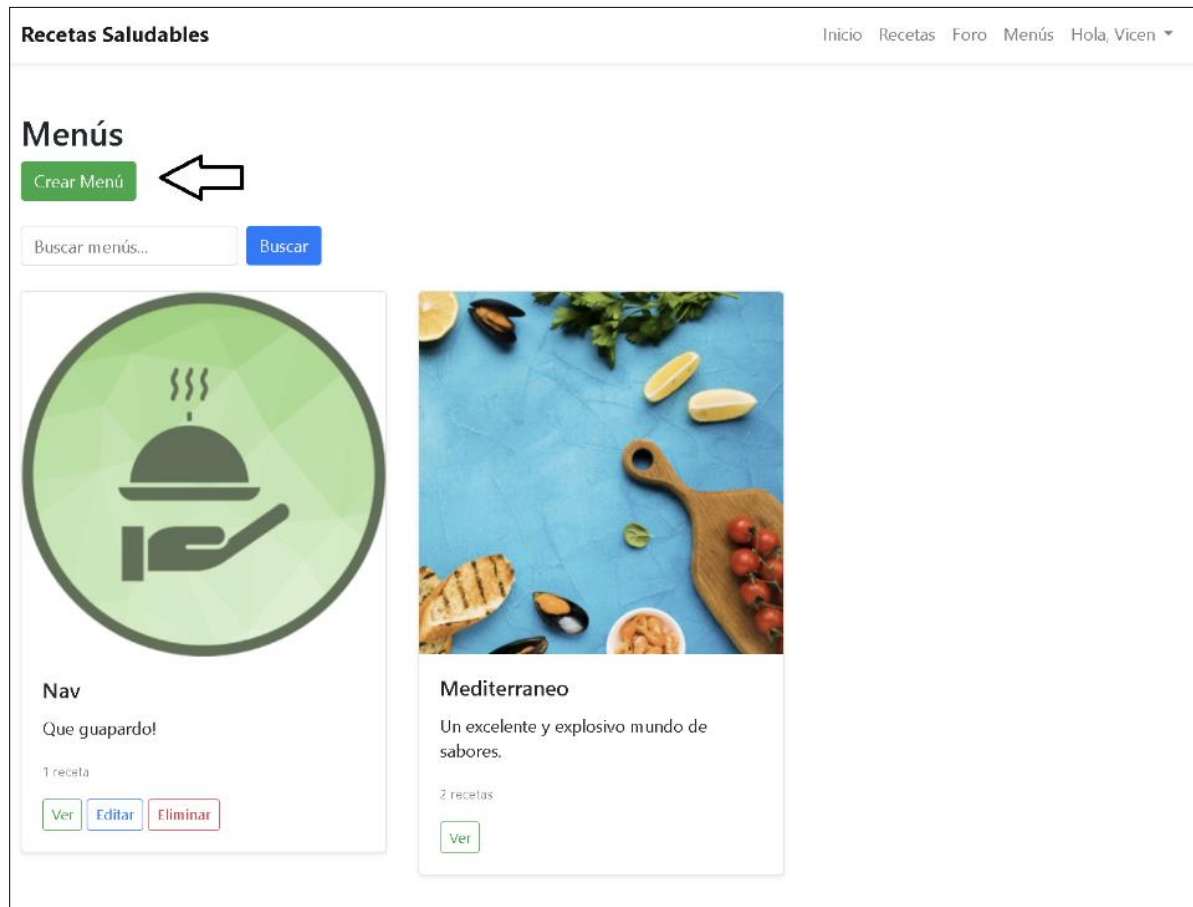
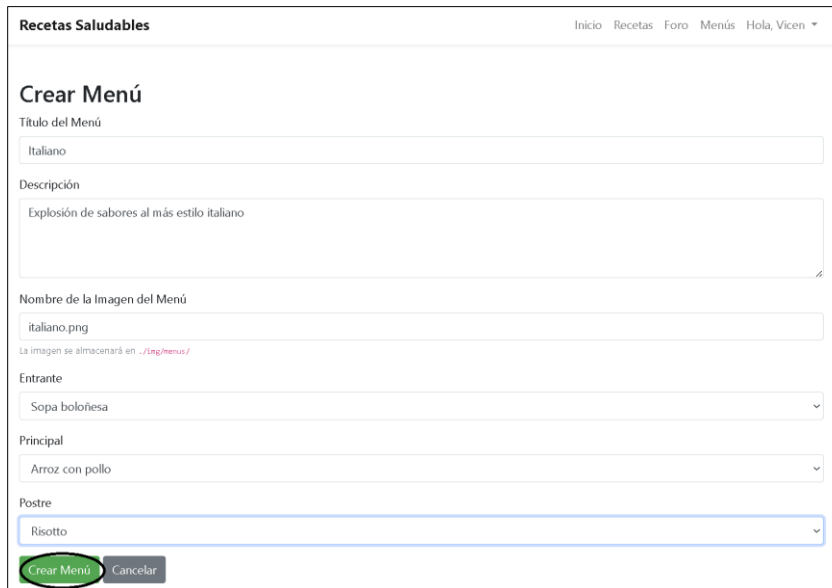


Ilustración 32. Crear menú

Para la elaboración del menú, se deberá cumplimentar los siguientes campos:



The screenshot shows a web form titled "Crear Menú" within the "Recetas Saludables" application. The form contains the following fields and options:

- Título del Menú:** A text input field containing "Italiano".
- Descripción:** A text area containing "Explosión de sabores al más estilo italiano".
- Nombre de la Imagen del Menú:** A text input field containing "italiano.png". Below it, a small red text note says "La imagen se almacenará en ./img/menus/".
- Entrante:** A dropdown menu with "Sopa boloñesa" selected.
- Principal:** A dropdown menu with "Arroz con pollo" selected.
- Postre:** A dropdown menu with "Risotto" selected.
- Buttons:** A green "Crear Menú" button and a grey "Cancelar" button.

Ilustración 33. Proceso de creación de menú. cumplimentación de campos.

Ahora se observa la captura que muestra nuevamente la página de "Menús", confirmando que el proceso de creación de un nuevo menú ha sido exitoso, demostrando la funcionalidad de la aplicación para añadir nuevos planes de comidas y que estos se integran de forma visible en el listado principal.

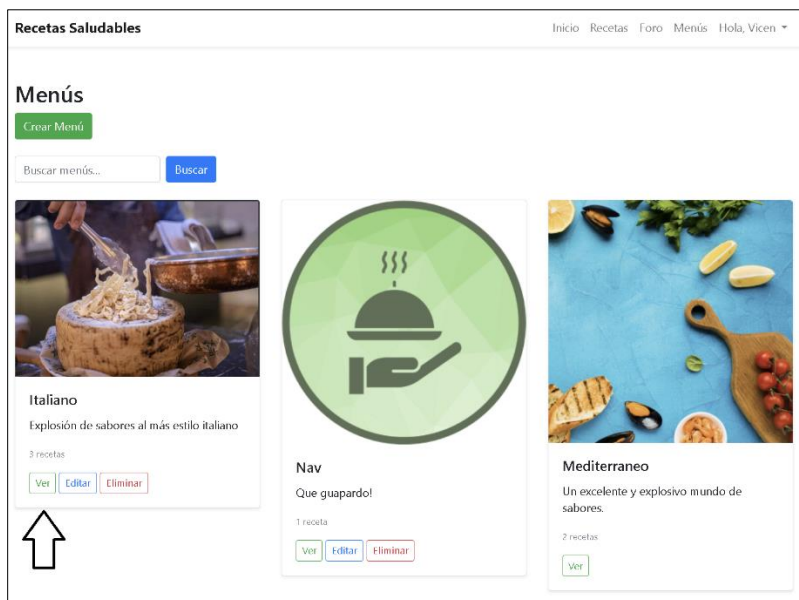



Ilustración 34. Inserción en listado principal.


Al acceder al menú creado con anterioridad se observa las características de este mismo como son el título, una breve descripción y los platos que integran el menú.

**Recetas Saludables** Inicio Recetas Foro Menús Hola, Vicen ▾


## Italiano




Explosión de sabores al más estilo italiano



**Arroz con pollo**  
Delicioso arroz con pollo....  
Principal  
[Ver Receta](#)



**Sopa boloñesa**  
Sopa boloñesa con un toque picante y cítrico....  
Entrante  
[Ver Receta](#)



**Risotto**  
El arroz se cocina primero brevemente en un sofrito de cebolla y mantequilla o aceite de oliva, para...  
Postre  
[Ver Receta](#)

[Volver a Menús](#)

Ilustración 35. contenido del menú.