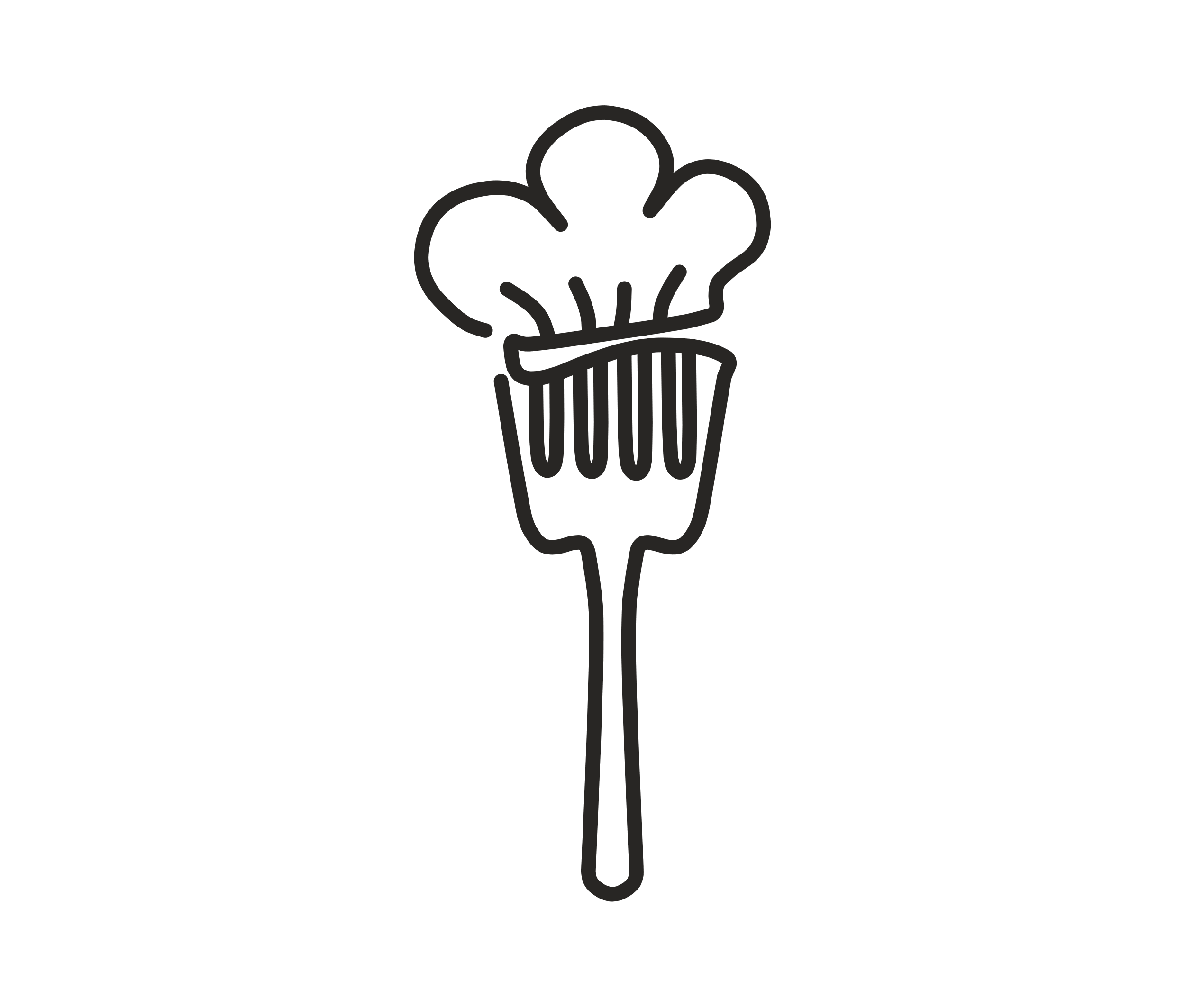
****

Tutor: Hugo MarÍa Vegas Carrasco

SERGIO ÁLVAREZ DE RON

JONATHAN HIDALGO

SERGIO MONTOIRO

IES LUIS BRAILLE

10 de junio de 2025

WeCook – Red Social de recetas

Desarrollo de Aplicaciones Web

**Índice de contenidos**

[1. Resumen 3](#_Toc199243479)

[2. Abstract 3](#_Toc199243480)

[3. Introducción 4](#_Toc199243481)

[4. Objetivos 5](#_Toc199243482)

[5. Alcance del proyecto 6](#_Toc199243483)

[5.1. Requisitos funcionales y no funcionales 6](#_Toc199243484)

[Requisitos funcionales 6](#_Toc199243485)

[Requisitos no funcionales 7](#_Toc199243486)

[5.2.Prototipado 8](#_Toc199243487)

[Wireframing 8](#_Toc199243488)

[Bocetado de vistas 9](#_Toc199243489)

[5.3. Tecnologías utilizadas 18](#_Toc199243490)

[Laravel 18](#_Toc199243491)

[Blade 18](#_Toc199243492)

[JavaScript 19](#_Toc199243493)

[Figma 21](#_Toc199243494)

[Miro 21](#_Toc199243495)

[Git 20](#_Toc199243496)

[GitHub 20](#_Toc199243497)

[Visual Studio Code 20](#_Toc199243498)

[MySQL 19](#_Toc199243499)

[MySQL Workbench 19](#_Toc199243500)

[Bootstrap 19](#_Toc199243501)

[5.4. Diagrama de Entidad-Relación 22](#_Toc199243502)

[5.5. Diagrama de Casos de Usos 24](#_Toc199243503)

[6. Marco práctico 25](#_Toc199243504)

[Estructura 25](#_Toc199243505)

[Controladores 29](#_Toc199243506)

[Modelos 29](#_Toc199243507)

[Rutas 29](#_Toc199243508)

[Vistas 29](#_Toc199243509)

[Base de datos 30](#_Toc199243510)

[Vista de inicio 31](#_Toc199243511)

[Vista de recordar contraseña: 32](#_Toc199243512)

[Cabecera común para todas las vistas 32](#_Toc199243513)

[Pie común para todas las vistas 32](#_Toc199243514)

[Vista de listado de recetas: 33](#_Toc199243515)

[Vista de detalle de receta 33](#_Toc199243516)

[Vista de creación/edición de receta: 34](#_Toc199243517)

[Vista de perfil de usuario 34](#_Toc199243518)

[Vista de seguidores/seguidos del usuario 35](#_Toc199243519)

[Modal de seguidores/seguidos 35](#_Toc199243520)

[Modal recetas guardadas 36](#_Toc199243521)

[Modal de editar receta 36](#_Toc199243522)

[Vista de administrador de recetas 36](#_Toc199243523)

[Vista de administrador de usuarios 37](#_Toc199243524)

[7. Conclusiones 38](#_Toc199243525)

[8. Bibliografía 39](#_Toc199243526)

[9. Webgrafía 39](#_Toc199243527)

# Resumen

Este proyecto consiste en el desarrollo de una aplicación web, concretamente una red social para compartir recetas de cocina. El foco principal ha sido construir una plataforma donde los usuarios puedan compartir recetas y mensajes e interactuar con otros usuarios de forma cómoda, sencilla y segura.

Para ello, hemos empleado Laravel en su versión 12, un framework de PHP moderno y robusto, que ha facilitado la implementación de funcionalidades en nuestra aplicación, como la autenticación de usuarios, manejo de rutas, controladores, relación entre modelos, validación de formulario y un largo etcétera de estándares para aplicaciones web.

A través de este proyecto hemos podido aprender la estructura de frameworks como Laravel, así como la forma de trabajar con él de forma eficiente. Además, hemos podido aplicar los conocimientos aprendidos a lo largo del grado, desde el manejo y estructuración de base de datos, hasta el diseño visual de la web, pasando por la creación de las diferentes funcionalidades a través de la programación.

# Abstract

This project involves the development of a web application, specifically a social network for sharing cooking recipes. The main focus was to build a platform where users can share recipes and messages and interact with other users in a convenient, simple, and secure way.

To achieve this, we used Laravel version 12, a modern and robust PHP framework, which facilitated the implementation of features in our application, such as user authentication, route management, controllers, model relationships, form validation, and a long list of web application standards.

Through this project, we were able to learn the structure of frameworks like Laravel, as well as how to work with it efficiently. In addition, we were able to apply the knowledge acquired throughout the degree, from database management and structuring to the visual design of the website, including the creation of different functionalities through programming.

# Introducción

En este trabajo se propone el diseño y desarrollo de una red social centrada en compartir recetas de cocina empleando tecnologías de desarrollo web como Laravel, JavaScript y MySQL. El auge y relevancia de las redes sociales en la actualidad nos ha llevado a crear una red social dedicada a la gastronomía, más concretamente sobre recetas, pues es un tema que a los integrantes del grupo de trabajo nos interesa. Esta plataforma busca ofrecer un espacio donde los usuarios puedan intercambiar sus mejores recetas, fotos y consejos, fomentando una experiencia interactiva y colaborativa.

El desarrollo de esta aplicación se sustenta en el uso de Laravel, un framework de PHP que gestiona el backend y la lógica de negocio, además de facilitar plantillas Blade para manejar el frontend de forma sencilla junto a JavaScript. Por otro lado, MySQL es la base de daos elegida para almacenar la información de los usuarios, recetas y comentarios, así como todas sus relaciones.

Todo esto se ha planteado con la finalidad de ofrecer una solución práctica, escalable y sencilla para los amantes de la gastronomía o cualquier persona que quiera aprender a cocinar o necesite de recursos culinarios para el día a día, además de incluir funciones que fomentan la interacción social.

# Objetivos

* Desarrollar una aplicación web funcional que permita a los usuarios registrarse, iniciar sesión y gestionar su cuenta de manera segura y sencilla, utilizando Laravel como framework.
* Implementar un sistema para la publicación de recetas de cocina, en el que los usuarios puedan crear, editar y eliminar sus propias recetas incluyendo título, descripción, ingredientes, pasos de preparación y una imagen ilustrativa.
* Diseñar una interfaz web intuitiva, atractiva y responsive, garantizando una experiencia de usuario óptima en dispositivos móviles y de escritorio.
* Definir y gestionar una base de datos relacional en MySQL, destinada a almacenar de forma eficiente y segura la información de usuarios, recetas y demás entidades necesarias para el funcionamiento de la aplicación.
* Incorporar un sistema de búsqueda y filtrado de recetas, que permita a los usuarios encontrar contenidos relevantes según nombre, ingredientes o categoría.
* Aplicar mecanismos de seguridad en el desarrollo de la aplicación, tales como validación de formularios, protección contra ataques comunes (como inyecciones SQL) y almacenamiento seguro de contraseñas.
* Aplicar buenas prácticas de desarrollo web tanto en el frontend como en el backend, incluyendo el uso de control de versiones con Git, documentación del código y separación de responsabilidades entre componentes.
* Realizar pruebas funcionales y de usuario, con el objetivo de verificar el correcto funcionamiento de todas las funcionalidades implementadas y asegurar una buena experiencia de uso.

# Alcance del proyecto

## 5.1. Requisitos funcionales y no funcionales

### Requisitos funcionales

Los requisitos funcionales describen las funcionalidades que el sistema debe ofrecer al usuario final. Para esta aplicación web de recetas de cocina, se han identificado los siguientes:

* Registro de usuarios: La aplicación debe permitir a nuevos usuarios registrarse mediante un formulario con nombre, correo electrónico y contraseña.
* Inicio de sesión y cierre de sesión: Los usuarios registrados deben poder autenticarse para acceder a funcionalidades personalizadas, así como cerrar sesión de forma segura.
* Gestión de perfil de usuario: Cada usuario debe poder visualizar y editar la información de su perfil, incluyendo su nombre, foto (opcional) y contraseña.
* Seguir usuarios: cada usuario tiene la opción de seguir a otros usuarios y viceversa, para poder ver las recetas de estos en su página de inicio de forma prioritaria.
* Publicación de recetas: Los usuarios autenticados podrán crear recetas introduciendo título, descripción, ingredientes, pasos de preparación e imagen.
* Edición y eliminación de recetas: Los usuarios podrán editar o eliminar únicamente las recetas que hayan creado.
* Visualización de recetas: Cualquier visitante podrá visualizar todas las recetas disponibles, incluyendo su contenido completo y la información del autor.
* Buscador de recetas: La aplicación debe ofrecer un sistema de búsqueda por palabras clave (título, ingredientes, etc.).
* Filtrado de recetas: El usuario podrá filtrar recetas por categorías predefinidas (por ejemplo: entrantes, platos principales, postres...).
* Sistema de comentarios: Los usuarios autenticados podrán comentar en las recetas de otros usuarios.
* Marcado de recetas favoritas: Los usuarios podrán marcar recetas como favoritas para consultarlas posteriormente.

### Requisitos no funcionales

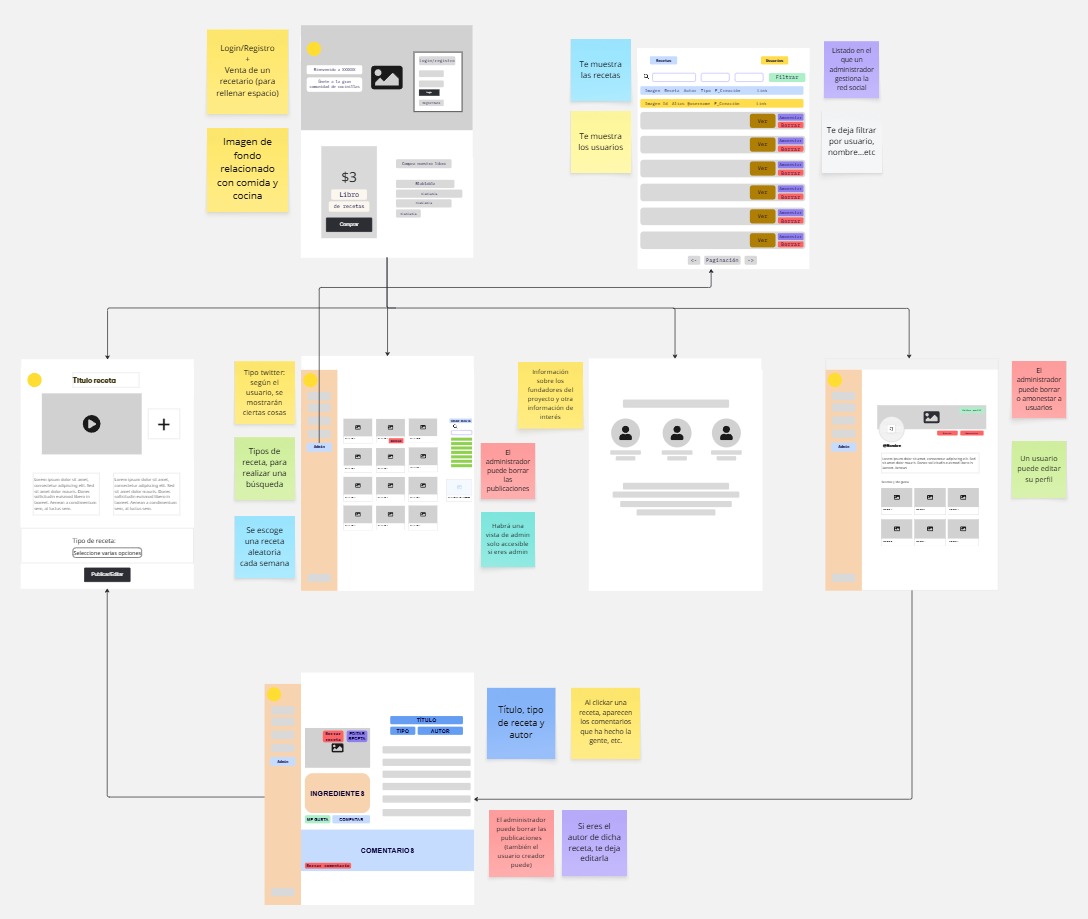
Los requisitos no funcionales definen características de calidad y restricciones técnicas que debe cumplir el sistema:

* Seguridad: Las contraseñas de los usuarios deben almacenarse cifradas. Se deben validar las entradas del usuario para evitar ataques como inyección SQL. Se debe implementar protección contra ataques CSRF en los formularios.
* Usabilidad: La interfaz debe ser intuitiva, fácil de usar y accesible para usuarios sin conocimientos técnicos. Se debe garantizar una experiencia fluida tanto en dispositivos móviles como en ordenadores.
* Rendimiento: Las páginas deben cargarse en un tiempo razonable (<2 segundos en condiciones normales). El sistema debe ser capaz de manejar múltiples usuarios y recetas sin disminuir significativamente su rendimiento.
* Compatibilidad: La aplicación debe ser compatible con los navegadores web más utilizados (Chrome, Firefox, Edge, Safari).
* Mantenibilidad: El código debe estar estructurado de forma clara y modular, utilizando buenas prácticas de programación para facilitar su mantenimiento y ampliación.
* Escalabilidad: La arquitectura basada en frontend y backend separados debe permitir futuras ampliaciones (por ejemplo, integración con APIs, app móvil, etc.).
* Portabilidad: El proyecto debe poder instalarse y desplegarse fácilmente en otros entornos utilizando herramientas comunes como Docker o entornos LAMP/WAMP.

## 5.2. Prototipado

### **Wireframing**

<https://miro.com/welcomeonboard/eGptdE1iVEdpZmcvM1M1RlJwTERXUmZmTkZjN3FxRGgrTEFoc0JKblNkekZ2eE8xYWR6UXpFbm9vWEEwR1pLcFd6WE8rSUNDRnZiTGFpakZEN244ZUgyTlVIZVhZNGNGdXJzbXJFNmFqNW5FSmJxY1lhQWFjNE9JNEl3SSs4SHhBd044SHFHaVlWYWk0d3NxeHNmeG9BPT0hdjE=?share_link_id=690549974316>



### Bocetado de vistas

* Landing page:

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

* About us:

Gráfico

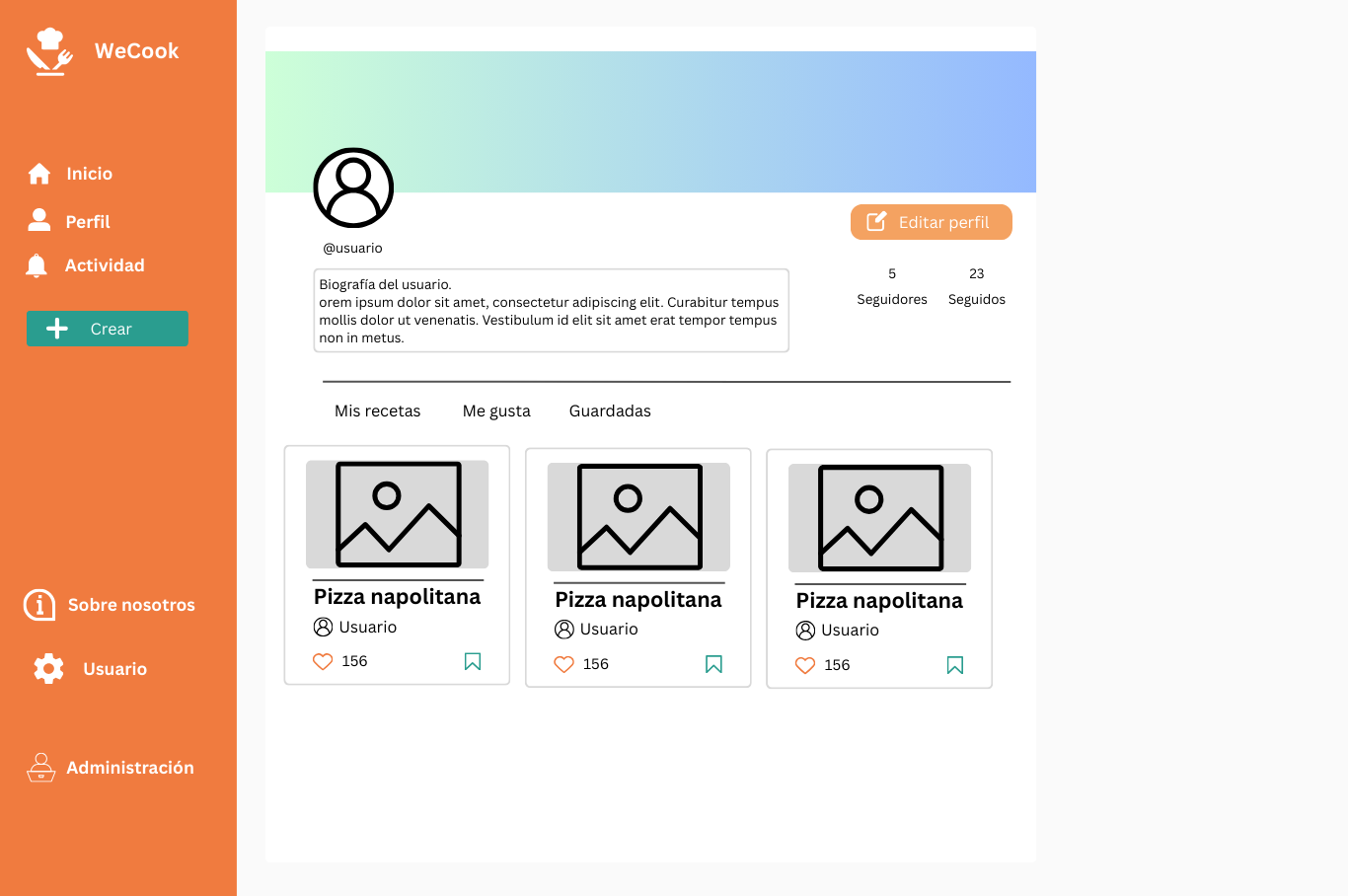
El contenido generado por IA puede ser incorrecto.

* Publicar y editar receta:

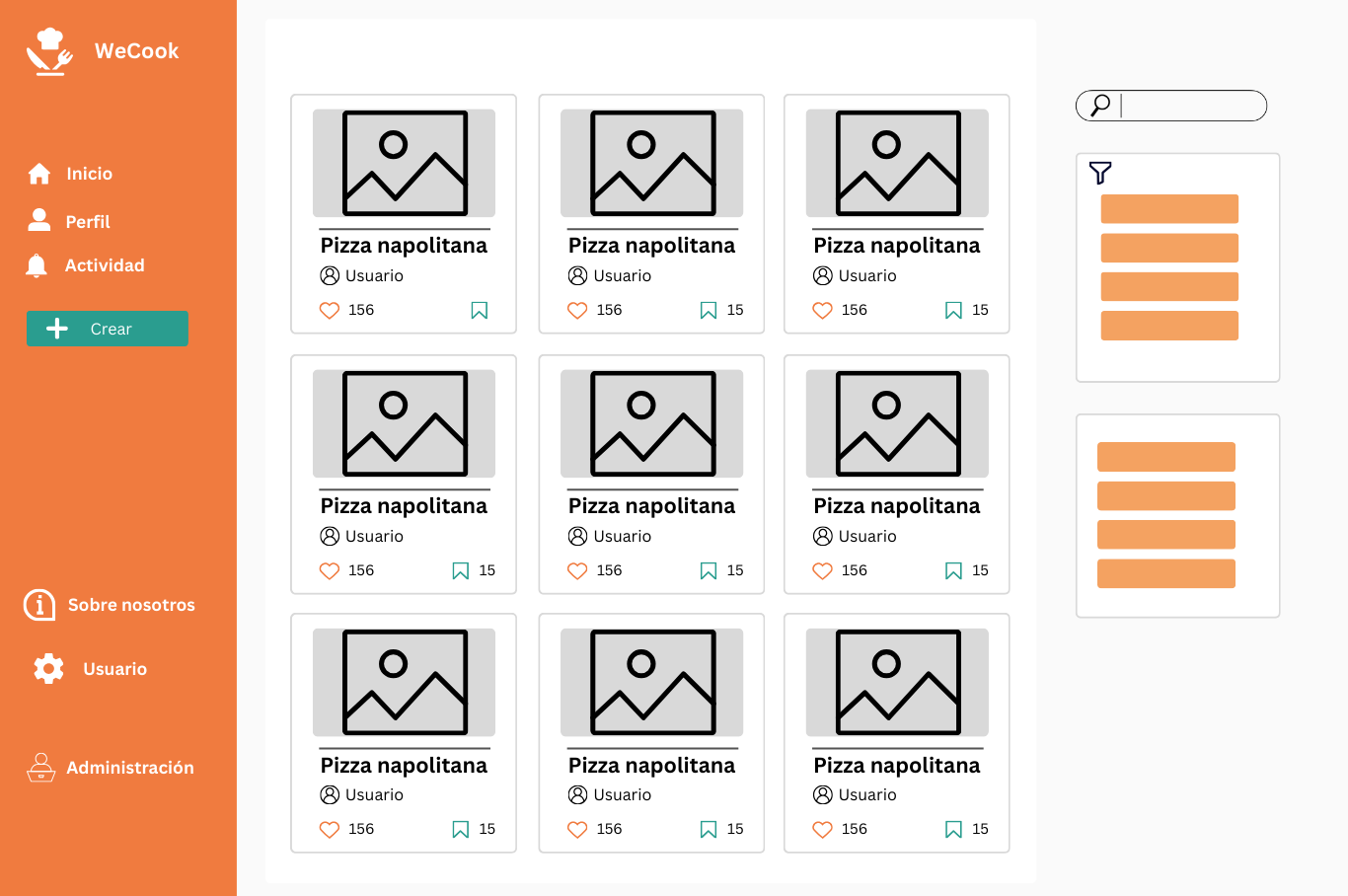
Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

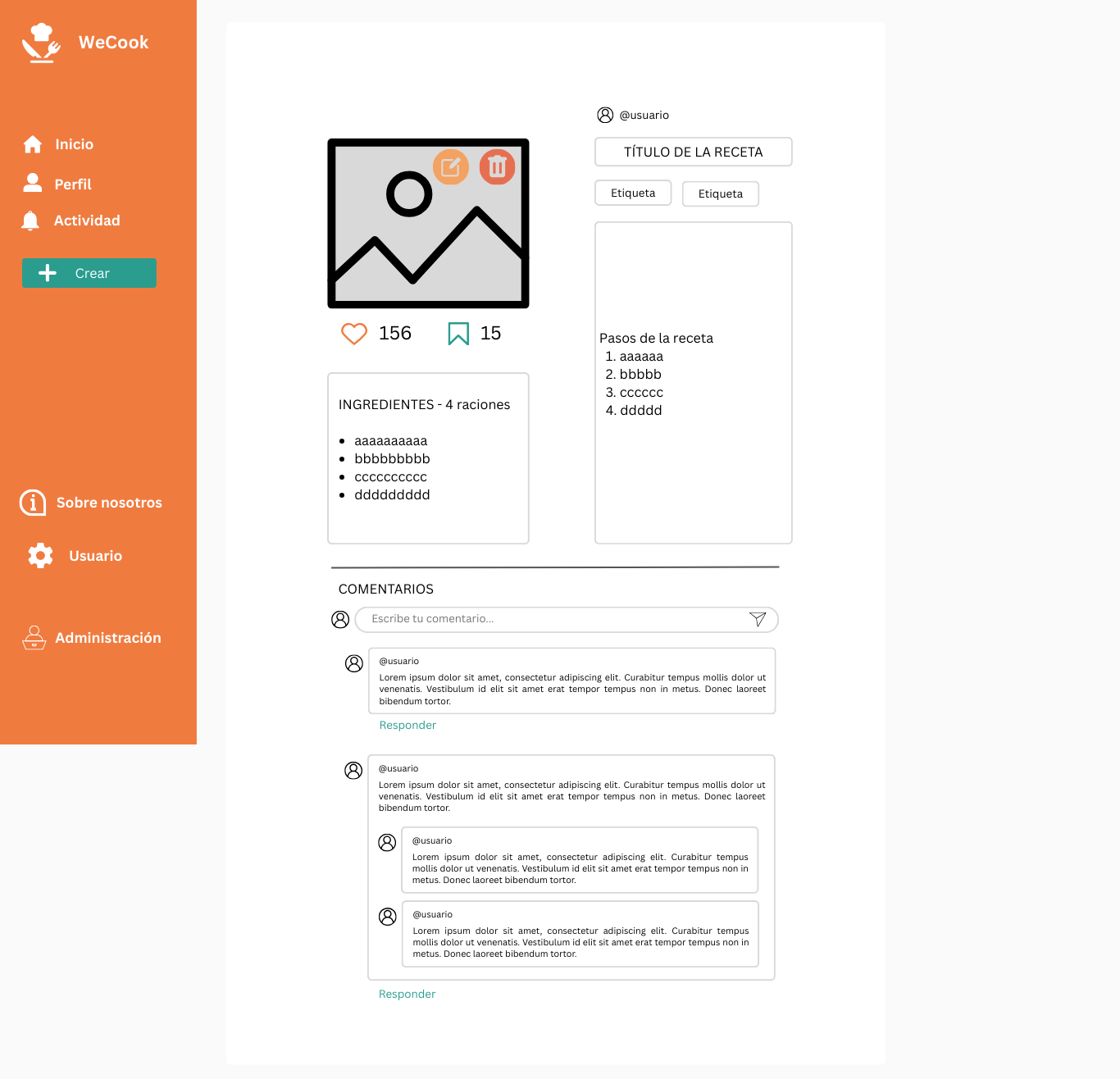
* Perfil de usuario:



* Página principal:



* Página de receta:



* Página de administrador:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Vistas en versión móvil

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Texto, Carta

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

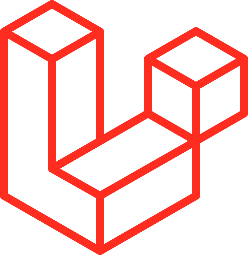
El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.

## 5.3. Tecnologías utilizadas

### Laravel

Laravel es un [framework](https://es.wikipedia.org/wiki/Framework" \o "Framework) de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) para desarrollar aplicaciones y servicios web con [PHP](https://es.wikipedia.org/wiki/PHP). Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "[código espagueti](https://es.wikipedia.org/wiki/C%C3%B3digo_espagueti)" empleando el patrón de arquitectura Modelo-Vista-Controlador (MVC), que facilita la organización del código separando la lógica de negocio, la interfaz y los datos.

Laravel ofrece utilidades que simplifican la creación de [aplicaciones web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web), agilizando significativamente este proceso y generando un código organizado y manejable. Gracias a esta característica, la popularidad de Laravel ha experimentado un rápido crecimiento en los últimos tiempos, atrayendo a numerosos desarrolladores que lo eligen como su framework preferido para alcanzar una eficiente metodología de desarrollo.

Hemos utilizado esta tecnología debido a su popularidad y capacidad de manejar de forma sencilla algunas tareas complejas como la gestión de usuarios, seguridad y conexión a bases de datos. Además, está basado en PHP, lenguaje de programación con el cuál estamos familiarizados y nos ha facilitado tanto el proceso de aprendizaje como de desarrollo.

### Blade

Blade es el motor de plantillas que viene integrado en Laravel y se utiliza para construir la parte visual de la aplicación. Permite escribir vistas de forma sencilla y estructurada, combinando HTML con una sintaxis específica que facilita la inclusión de lógica directamente en las plantillas, como bucles, condicionales y estructuras heredadas.

Una de las principales ventajas de Blade es que permite reutilizar componentes de la interfaz mediante plantillas y secciones, lo que promueve una mejor organización del código frontend. También permite el uso de directivas como @if, @foreach, @include o @yield, que simplifican mucho la creación de páginas dinámicas sin necesidad de mezclar directamente PHP con HTML.

Hemos elegido Blade como sistema de plantillas porque se integra perfectamente con Laravel y proporciona una forma clara, limpia y mantenible de construir las vistas. Su uso nos ha permitido separar correctamente la lógica del backend y la presentación del frontend, favoreciendo así el desarrollo ágil y ordenado de la aplicación.

### JavaScript

JavaScript es un lenguaje de programación interpretado, ampliamente utilizado en el desarrollo web para añadir interactividad y dinamismo a las páginas. Se ejecuta del lado del cliente, lo que permite modificar el contenido del HTML y CSS en tiempo real sin necesidad de recargar la página.

Esta tecnología es fundamental para nuestro proyecto ya que permite crear funcionalidades interactivas como validación de formularios, menús desplegables, animaciones y comunicación asincrónica con el servidor mediante AJAX. Gracias a su versatilidad y compatibilidad con todos los navegadores, JavaScript es esencial en cualquier aplicación web moderna.

### Bootstrap

Icono

El contenido generado por IA puede ser incorrecto.Bootstrap es un framework frontend que proporciona estilos CSS y componentes JavaScript listos para usar. Fue creado por Twitter y está enfocado en el diseño responsive, permitiendo que las aplicaciones se adapten automáticamente a diferentes tamaños de pantalla. Hemos utilizado Bootstrap para construir una interfaz visual atractiva y funcional sin necesidad de escribir mucho CSS personalizado. Gracias a sus componentes predefinidos como formularios, botones, menús y grillas, pudimos acelerar el desarrollo de la parte visual de nuestra aplicación y garantizar una buena experiencia en dispositivos móviles.

### MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto. Permite almacenar, consultar y gestionar grandes cantidades de datos estructurados de manera eficiente mediante el uso de lenguaje SQL.

Hemos utilizado MySQL como base de datos principal del proyecto, ya que permite definir relaciones entre entidades como usuarios, recetas o comentarios. Su compatibilidad con Laravel facilita el acceso a los datos de forma sencilla y segura.

### MySQL Workbench

Icono

El contenido generado por IA puede ser incorrecto.MySQL Workbench es una herramienta visual que permite modelar, administrar y consultar bases de datos MySQL de manera gráfica. Incluye funciones para diseñar esquemas, realizar consultas SQL, gestionar usuarios y analizar el rendimiento.

La hemos utilizado para diseñar y mantener la estructura de nuestra base de datos, visualizar relaciones entre tablas y ejecutar consultas de prueba, lo cual ha sido de gran ayuda en la fase de desarrollo y prueba del proyecto.

### Git

Icono

El contenido generado por IA puede ser incorrecto.Git es un sistema de control de versiones distribuido que permite gestionar el historial de cambios en el código fuente de un proyecto. Facilita la colaboración entre varios desarrolladores al permitir trabajar en ramas independientes y luego fusionar cambios de manera fácil y segura.

En nuestro TFG, Git ha sido fundamental para trabajar en equipo sin conflictos, mantener un historial de versiones y poder recuperar estados anteriores del proyecto en caso de errores. Su integración con plataformas como GitHub lo convierte en una herramienta indispensable en cualquier entorno de desarrollo profesional.

### GitHub

GitHub es una plataforma web para alojar repositorios de Git. Proporciona funcionalidades colaborativas y gestión de versiones, además de facilitar el trabajo en equipo y la revisión de código.

Hemos utilizado GitHub para centralizar nuestro proyecto, compartir avances, y realizar control de versiones de forma remota. Su interfaz amigable y sus herramientas de colaboración han permitido una organización clara y efectiva del flujo de trabajo durante el desarrollo de la aplicación.

### Visual Studio Code

Icono

El contenido generado por IA puede ser incorrecto.Visual Studio Code (VS Code) es un editor de código fuente ligero, potente y multiplataforma desarrollado por Microsoft. Incluye herramientas de depuración, resaltado de sintaxis, autocompletado y un amplio ecosistema de extensiones.

En nuestro proyecto, VS Code ha sido nuestro entorno de desarrollo principal, facilitando la escritura de código en Laravel, JavaScript y otros lenguajes gracias a su integración con Git, terminal integrada y compatibilidad con diversas extensiones específicas para cada tecnología utilizada.

### Figma

Gráfico

El contenido generado por IA puede ser incorrecto.Figma es una herramienta de diseño gráfico basada en la nube, especialmente orientada al diseño de interfaces de usuario (UI) y experiencia de usuario (UX). Permite trabajar de forma colaborativa en tiempo real, lo que facilita la creación de prototipos interactivos, wireframes y maquetas de alta fidelidad antes del desarrollo.

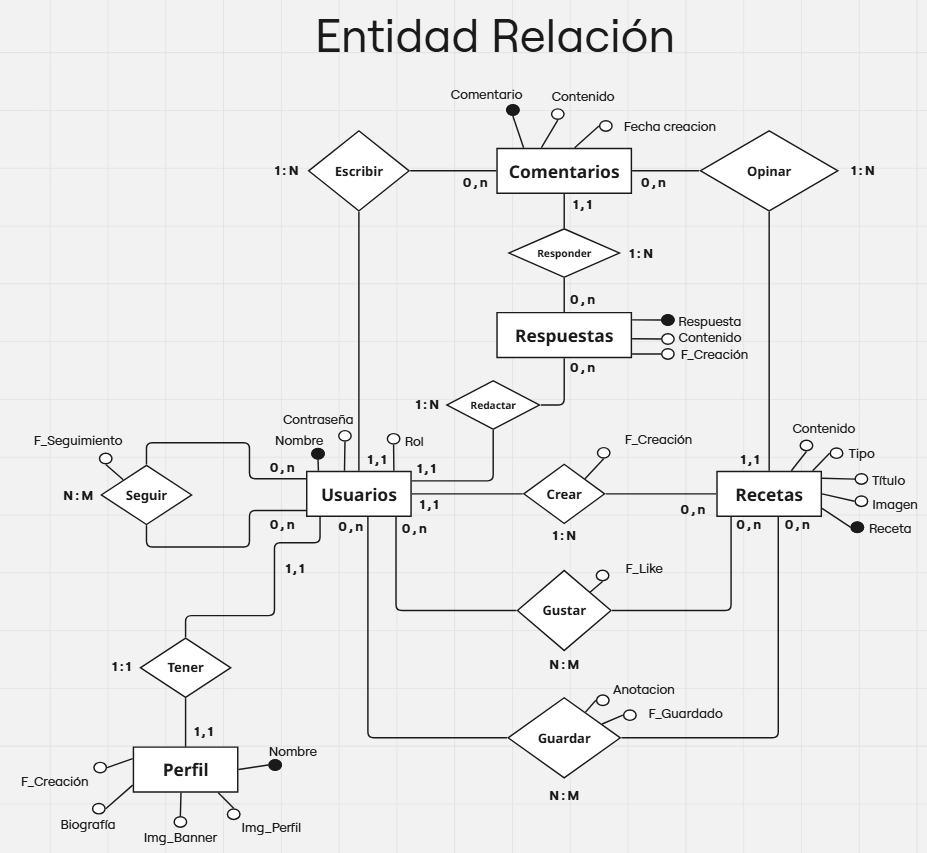
Hemos utilizado Figma para diseñar visualmente las pantallas de nuestra aplicación web, asegurándonos de que la experiencia de usuario sea intuitiva y atractiva. Gracias a sus funcionalidades de colaboración y su enfoque centrado en el diseño responsive, ha sido clave para definir la estructura visual del proyecto.

### Miro

Miro es una plataforma colaborativa de pizarras digitales en línea diseñada para facilitar la planificación, organización y trabajo en equipo de forma visual. Permite a varios usuarios interactuar en tiempo real sobre una misma pizarra, añadiendo notas, diagramas, mapas mentales, esquemas y otros elementos visuales.

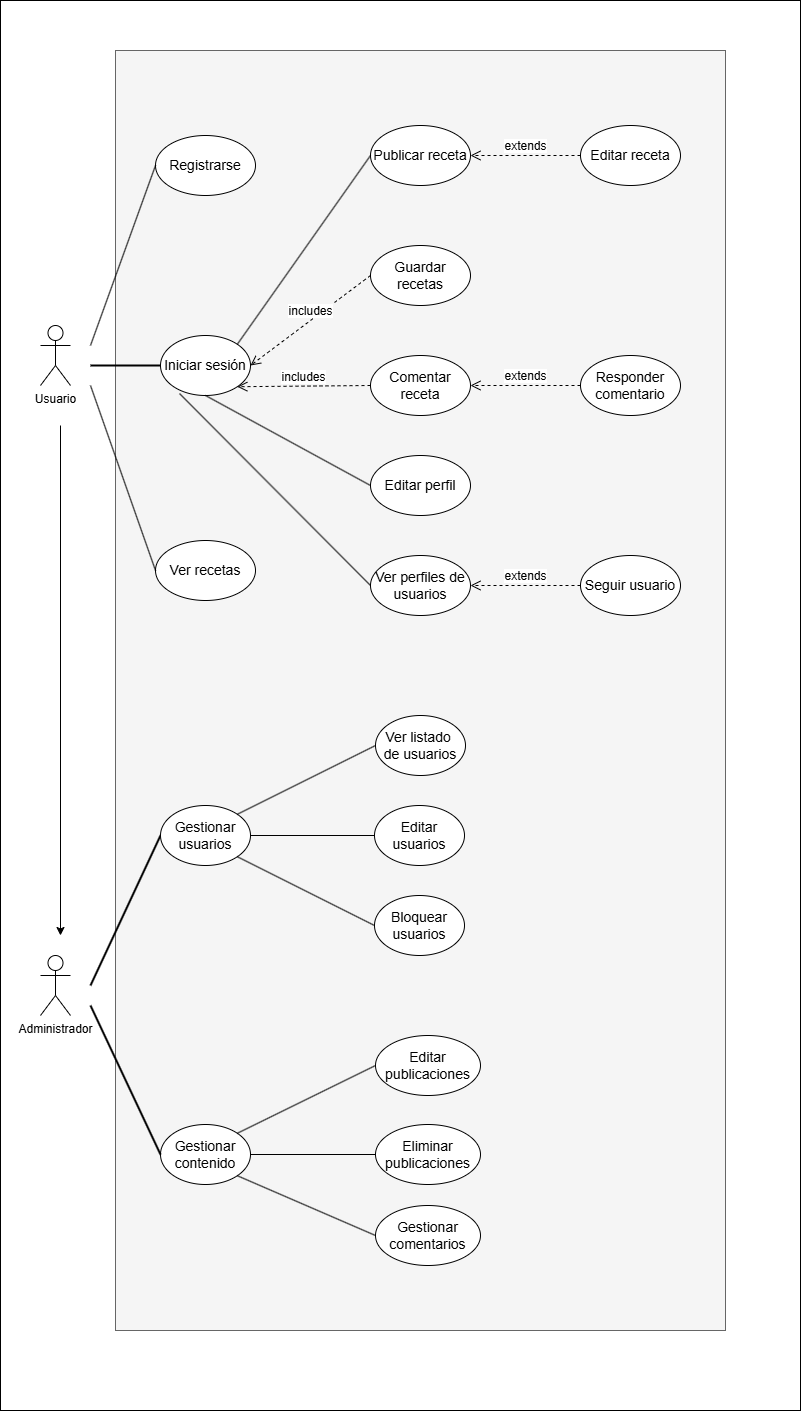
En nuestro TFG hemos utilizado Miro principalmente para planificar el proyecto, organizar ideas y definir estructuras de la aplicación como diagramas de flujo o esquemas de navegación. Su interfaz intuitiva y sus múltiples plantillas nos han permitido coordinar mejor el trabajo en grupo, clarificar tareas y tener una visión global del desarrollo.

## 5.4. Diagrama de Entidad-Relación





## 5.5. Diagrama de Casos de Usos



# Marco práctico

## Estructura

La estructura de carpetas y ficheros de este proyecto es la que trae Laravel en su versión 12 por defecto, como se muestra y se explica a continuación:

Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.

* Raíz del Proyecto (/proyecto\_recetas)

Aquí se encuentran las carpetas y otros archivos importantes como:

* .env → Configuración del entorno (base de datos, mail, claves API...).
* artisan → Script de consola para ejecutar comandos Laravel.
* composer.json → Gestión de dependencias PHP.
* package.json → Dependencias y scripts del frontend.
* app/

Contiene el núcleo de la aplicación (la lógica de negocio). Aquí se agrupan:

* Http/Controllers/ → Controladores de la lógica web (patrón MVC).
* Models/ → Modelos que representan entidades y gestionan los datos de la base de datos.
* Http/Middleware/ → Filtros que se ejecutan al procesar una petición HTTP.
* Providers/ → Servicios que configuran e inyectan funcionalidades en la aplicación.
* bootstrap/

Contiene el archivo app.php, que se ejecuta al iniciar Laravel. Se usa para cargar y configurar el framework. También incluye la carpeta cache/ para el almacenamiento de archivos cacheados que mejoran el rendimiento.

* config/

Contiene archivos .php de configuración para todos los componentes de Laravel: base de datos (database.php), correo (mail.php), sesiones (session.php), etc. Permite modificar fácilmente el comportamiento de la aplicación.

* database/

Contiene todo lo relacionado con la base de datos:

* migrations/ → Archivos para crear/modificar la estructura de la base de datos.
* seeders/ → Archivos que insertan datos de prueba o iniciales.
* factories/ → Generadores de datos de prueba para testing.
* public/

Carpeta accesible públicamente desde el navegador. Contiene:

* index.php → Punto de entrada a la aplicación.
* Archivos públicos como imágenes, hojas de estilo (CSS), scripts (JS), etc.
* resources/

Contiene los archivos "front-end" de la aplicación:

* views/ → Archivos Blade (.blade.php) que definen la interfaz del usuario.
* lang/ → Archivos de traducción para aplicaciones multilenguaje.
* css/y js/ → Archivos fuente que serán compilados para el frontend.
* routes/

Define las rutas de la aplicación. Laravel organiza las rutas por contexto:

* web.php → Rutas para la web (HTML, vistas Blade).
* api.php → Rutas para APIs REST.
* console.php → Comandos de consola (Artisan).
* channels.php → Rutas para canales de broadcasting (WebSockets).
* storage/

Almacena archivos generados o utilizados por la aplicación:

* logs/ → Archivos de registro de errores.
* app/ → Archivos del usuario (como subidas).
* framework/ → Archivos temporales de cache, sesiones, vistas compiladas, etc.

Esta carpeta no debe ser accesible desde el navegador, excepto el subdirectorio storage/app/public si se enlaza correctamente ejecutando el comando php artisan storage:link.

* tests/

Contiene los tests automatizados:

* Feature/ → Pruebas de alto nivel de funcionalidades completas.
* Unit/ → Pruebas unitarias de clases específicas.

Laravel usa PHPUnit por defecto, pero puede integrarse con Pest u otros frameworks.

## Landing page con inicio de sesión y registro de usuario

Interfaz de usuario gráfica, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

La pantalla principal, donde se sitúa la ruta raíz, es la landing page. En ella se encuentra el inicio de sesión y el registro de usuario. Aquí se define la ruta que lleva a esta página:

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Además, controla si el usuario ya está autenticado (es decir, ya ha iniciado sesión) y en tal caso, lo redirige a la página de listado de recetas a través de una llamada a uno de los controladores (la página predeterminada para los usuarios que ya iniciaron sesión). De esta forma se evitan problemas como cuando una sesión está iniciada, no pueda volver a iniciarse teniendo así múltiples sesiones simultáneas iguales abiertas.

Por otro lado, tenemos control de inicio erróneo de sesión. En caso de que un usuario trate de iniciar con un usuario inexistente o unas credenciales erróneas, se le redirige directamente a la pestaña de registro con un mensaje de error:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

En cuanto al registro de usuario, controla también con advertencias si el usuario que intentas registrar ya existe o si la repetición de contraseña es incorrecta:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Si el usuario consigue registrarse correctamente, se le redirige a la vista de listado de recetas (ruta predeterminada para usuarios autenticados).

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Todo esto lo manejan los controladores de autentificación por defecto que trae Laravel, facilitando mucho el trabajo. En uno de los controladores, hemos creado una función que crea al usuario de forma distinta a como viene por defecto, pues para cada usuario queremos crear un perfil al que se le asocia, que pueda editar a su gusto. En esta imagen se ve la función del controlador que crear el perfil del usuario según se registra.

Texto

El contenido generado por IA puede ser incorrecto.

## Recordar contraseña:

Si el usuario necesita recuperar su contraseña, puede acceder al enlace que se sitúa justo debajo de la sección de inicio de sesión, que le llevará a esta vista:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Esto enviará un mensaje a su correo electrónico como el siguiente (y muestra un aviso de que se envió correctamente, como se ve en la anterior imagen en el recuadro verde):

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Pulsar el botón del mensaje, le llevará a otra vista, donde podrá cambiar su contraseña por una nueva y de esta forma poder iniciar sesión con ella:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

El envío de correos electrónicos se gestiona desde los controladores por defecto de Laravel, que se encuentran dentro de la carpeta /controllers/auth. Por ejemplo: este controlador se encarga de enviar el enlace de reinicio de contraseña al usuario.

Texto

El contenido generado por IA puede ser incorrecto.

Maneja el correo específico que requiere la actualización de contraseña y así controla la seguridad, pues no permite acceder desde enlaces diferentes al cambio de contraseña de un correo concreto. Si se trata de poner otro correo, se mostrará un aviso y no se realizará la acción:

Imagen que contiene Aplicación

El contenido generado por IA puede ser incorrecto.

Además, una vez realizado el cambio de contraseña, el enlace pierde la efectividad y, en lugar de redirigir nuevamente a la vista de restablecer contraseña, lleva al usuario a la página principal.

## Verificación de usuario

Cuando un usuario se registra, se le envía un correo electrónico con el cual se ha registrado, para realizar una autenticación de seguridad de que se registró con un correo correcto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Esto es manejado por el siguiente controlador:

Texto

El contenido generado por IA puede ser incorrecto.

Este controlador también se genera por defecto por Laravel, aunque modificamos la ruta por defecto a la que el enlace del correo electrónico redirige tras autenticar correctamente al usuario (en nuestro caso, decidimos redirigir al listado de recetas).

## Barra de navegación

Para todas las vistas de los usuarios autenticados (es decir, aquellos que han iniciado sesión), se muestra una barra de navegación (la cual se puede minimizar y ampliar), con diversos enlaces al resto de vistas y funcionalidades de la aplicación.

Tanto la barra de navegación como las diferentes vistas se renderizan en el siguiente archivo app.blade.php:

Texto

El contenido generado por IA puede ser incorrecto.

## Listado de recetas:

Esta es la vista principal predeterminada para los usuarios con sesión iniciada. Aquí se muestran las recetas publicadas por los usuarios. Permite visualizar las recetas en detalle pulsando sobre ellas, además de dar me gusta y guardar recetas con los botones y acceder al perfil del usuario creador de cada receta pulsando sonre su nombre.

Por otro lado, se pueden realizar búsquedas de recetas por filtros y búsqueda de usuarios.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Todo esto se gestiona a través de las funciones del controlador específico para manejar las recetas:

Texto

El contenido generado por IA puede ser incorrecto.

Esta función concreta del controlador lista las recetas mediante AJAX, lo cual permite la actualización dinámica de las mismas sin necesidad de recargar la página, permitiendo una navegación mucho más cómoda y fluida para el usuario.

Las recetas que se muestran dependen de los usuarios a los que sigues y de la popularidad de las recetas en base a la cantidad de “Me gusta” que reciben. De esta forma, este simple algoritmo muestra mayor diversidad de recetas que si simplemente se muestran por fecha de creación.

## Detalle de receta

Esta vista muestra la información concreta de la receta, así como todos los comentarios y respuestas de los usuarios.

Texto

El contenido generado por IA puede ser incorrecto.

Esta función del controlador se encarga de manejar esta vista en función de cada receta (según su ID). En función del usuario que visualice la receta, tendrá diferentes opciones disponibles. Si es el usuario creador, se le mostrarán botones para editar y eliminar la receta, pero no podrá dar me gusta ni guardar su propia receta. Los usuarios con sesión iniciada podrán dar me gusta, guardar la receta y comentar. Por último, los usuarios invitados (es decir, que no han iniciado sesión) tan solo podrán visualizar dicha receta, sin poder realizar ninguna otra acción.

En este fragmento de código se muestran las funciones del controlador que permitan al propietario de la receta editar a esta misma:

Texto

El contenido generado por IA puede ser incorrecto.

Una de ellas redirige al modal de edición con los datos de dicha receta escritos desde el inicio para editarlos. La otra, se encarga de persistir la información en la base de datos (realizando las comprobaciones y validaciones pertinentes) y redirige de nuevo al listado tras la actualización.

Por otro lado, se muestra también la función que permite al usuario eliminar su propia receta:

Texto

El contenido generado por IA puede ser incorrecto.

Realiza las comprobaciones necesarias para impedir que otros usuarios eliminen recetas que no son suyas y elimina la receta de la base de datos si la acción se realiza con éxito.

Además, se muestran avisos con doble verificación para eliminar las recetas para mayor seguridad e impedir el borrado de información de forma indeseada:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

También se muestra otra alerta avisando si la acción de borrado se ha realizado correctamente:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

En cuanto a las funcionalidades de gustar y guardar las recetas, se manejan desde las siguientes funciones del mismo controlador:

Texto

El contenido generado por IA puede ser incorrecto.

Comprueba si el usuario ha iniciado sesión para permitirle realizar la acción y en caso afirmativo, verifica si la receta ya está guardada o no. En caso de no estarlo aún, realiza el guardado de la receta y lo persiste en base de datos haciendo uso del modelo.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Esta misma situación se replica de manera similar, pero para eliminar las recetas de los guardados del usuario.

Por otro lado, tenemos el mismo funcionamiento para las recetas gustadas:

Texto

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Cabe destacar que en ambos casos (para guardar o dar me gusta a recetas), las acciones se realizan a través de AJAX para facilitar la navegación, pues no es necesario que la página se recargue completamente, si no que solo se actualizan aquellas partes de la vista que se ven afectadas por estas acciones.

Como último apartado a mencionar sobre esta vista del detalle de recetas, mencionar el apartado de los comentarios y respuestas. Este apartado se maneja desde controladores diferentes, como se muestra a continuación:

Texto

El contenido generado por IA puede ser incorrecto.

Este fragmento de código se encarga de realizar las validaciones y persistencia en base de datos que corresponde a los comentarios de una receta. Primero comprueba si el usuario se encuentra con sesión iniciada y, tras esto, permite guardar el comentario.

Algo muy similar ocurre para las respuestas a los comentarios:

Texto

El contenido generado por IA puede ser incorrecto.

## Crear receta

Abc

Interfaz de usuario gráfica, Aplicación, PowerPoint

El contenido generado por IA puede ser incorrecto.

## Vista de perfil de usuario

Abc

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Vista de seguidores/seguidos del usuario

Abc

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

## Editar perfil

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Vista de administrador de usuarios/recetas

Abc

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Vista about

Abc

Diagrama

El contenido generado por IA puede ser incorrecto.

# Conclusiones

La realización de este proyecto ha supuesto un gran reto y una excelente oportunidad de aprendizaje. Laravel, aunque completamente nuevo para nosotros al comienzo del desarrollo, ha demostrado ser una herramienta muy útil, bien documentada y con una comunidad activa que facilita el aprendizaje.

Gracias a su estructura basada en el patrón MVC (el cual ya conocíamos) y sus numerosas funcionalidades integradas, hemos podido comprender mejor cómo se construye una aplicación web profesional desde cero, siendo además un trabajo más cómodo de realizar que con PHP simple.

También he mejorado mis conocimientos de bases de datos relacionales (MySQL), seguridad en aplicaciones web y diseño de interfaces. En definitiva, este proyecto nos ha permitido aplicar conocimientos teóricos adquiridos a lo largo del grado y desarrollar competencias clave para nuestro futuro profesional en el ámbito del desarrollo web.

# Bibliografía

No hay bibliografía consultada.

# Webgrafía

Bootstrap. (s. f.). <https://getbootstrap.com/>  
Laravel. (s. f.). <https://laravel.com/docs/12.x>  
MySQL Documentation. (s. f.). <https://dev.mysql.com/doc/>  
Stack Overflow. (s. f.). <https://stackoverflow.com/>  
W3Schools. (s. f.). <https://www.w3schools.com/>  
Composer. (s. f.). https://getcomposer.org/doc/  
Laravel Breeze. (s. f.). <https://laravel.com/docs/12.x/starter-kits#breeze>  
Font Awesome. (s. f.). https://fontawesome.com/docs  
Google Fonts. (s. f.). <https://fonts.google.com/>