Tecnologías utilizadas (punto 5.3)

**Laravel**

Laravel es un framework de desarrollo web basado en PHP que permite construir aplicaciones modernas de forma estructurada y eficiente. Utiliza el patrón de arquitectura Modelo-Vista-Controlador (MVC), lo que facilita la organización del código separando la lógica de negocio, la interfaz y los datos. Laravel proporciona herramientas integradas como el sistema de rutas, migraciones de base de datos, autenticación, validación de formularios y plantillas Blade, lo que reduce y facilita significativamente el desarrollo del proyecto, a la vez que es sencillo de aprender.  
Hemos utilizado esta tecnología debido a su popularidad en el entorno backend y a su capacidad para manejar fácilmente tareas complejas como la gestión de usuarios, seguridad, y acceso a bases de datos, además de estar basado en PHP, lenguaje con el que estamos muy familiarizados tras haber trabajado con él durante todo el curso. Además, su sintaxis clara favorece el mantenimiento y escalabilidad del proyecto.

**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.

**Figma**

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.

**Git**

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**

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.

**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 mediante Eloquent ORM facilita el acceso a los datos de forma sencilla y segura.

**MySQL Workbench**

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.

**Bootstrap**

Bootstrap es un framework front-end 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.

Marco práctico (punto 6)

Vista de inicio (sin personalizar):

Diagrama

El contenido generado por IA puede ser incorrecto.

Vista de registro:

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Vista de login:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Vista de recordar contraseña:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Vista de listado de recetas:

Interfaz de usuario gráfica, Texto, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

Vista de creación de receta:

Patrón de fondo

El contenido generado por IA puede ser incorrecto.

Vista de edición de receta:

Interfaz de usuario gráfica, Aplicación, Patrón de fondo

El contenido generado por IA puede ser incorrecto.