Universidad Politécnica
de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

Grado en Ingeniería Informática

Trabajo Fin de Grado

Desarrollo de una aplicación web para la gestión de opiniones y gustos gastronómicos

Autor: Adrián Nogales Gil

Tutor(a): Alejandro Rodríguez González

Co-Tutor(a): Andrea Álvarez Pérez

Madrid, octubre de 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Ingeniería Informática
Título: Desarrollo de una aplicación web para la gestión de opiniones y gustos gastronómicos
Octubre de 2025

Autor: Adrián Nogales Gil

Tutor:
Alejandro Rodríguez González
Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software
ETSI Informáticos
Universidad Politécnica de Madrid

Co-Tutora:
Andrea Álvarez Pérez
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

Este proyecto se centra en el desarrollo de una aplicación web denominada "GastroLog" cuyo objetivo es una red social, diseñada para los apasionados de la gastronomía. La plataforma intenta establecer una comunidad, dónde los usuarios podrán explorar, calificar, y compartir sus aventuras en restaurantes de cualquier ciudad. La meta fundamental es simplificar la búsqueda de nuevas vivencias culinarias, mediante un sistema sustentado en la confianza y las conexiones sociales. En vez de limitarse a críticas anónimas, GastroLog dejará a sus usuarios seguir a críticos gastronómicos o amigos. Esto generará un flujo de actividades, funcionando como una fuente de recomendaciones digna de confianza. Para estimular la interacción, se incluirán perfiles de usuario, reseñas detalladas, y valoraciones. Toda la información relevante de usuarios, restaurantes y críticas se almacenará en una base de datos PostgreSQL. La arquitectura de la aplicación se fundamentará en un modelo de tres capas. El backend, construido en Java usando Spring Boot, proporcionará una API REST, facilitando la comunicación con el frontend. La interfaz, se construirá empleando tecnologías web, tales como React, CSS, y JavaScript, utilizando Bootstrap 5, para obtener un diseño intuitivo y adaptable a cualquier dispositivo. Esta aplicación intenta abordar la dificultad de hallar sugerencias culinarias personalizadas y fiables. Las plataformas actuales a menudo exhiben listados genéricos, pero, les falta un aspecto social que permita a los individuos descubrir sitios mediante vivencias compartidas con individuos que compartan sus predilecciones.

Abstract

This project focuses on the development of a web application called "GastroLog." It will be a social network designed for food enthusiasts. The platform aims to create a community where users can explore, rate, and share their restaurant adventures in any city. The main goal is to simplify the search for new culinary experiences through a system based on trust and social connections. Instead of relying on anonymous reviews, GastroLog will allow its users to follow food critics or friends. This will generate a stream of activity, functioning as a trustworthy source of recommendations. To encourage interaction, user profiles, detailed reviews, and ratings will be included. All relevant information about users, restaurants, and reviews will be stored in a PostgreSQL database. The app's architecture will be based on a three-tier model. The backend, built in Java using Spring Boot, will provide a REST API, facilitating communication with the frontend. The interface will be built using web technologies such as React, CSS, and JavaScript, using Bootstrap 5, to achieve an intuitive design adaptable to any device. This app seeks to address the difficulty of finding personalized and reliable culinary suggestions. Current platforms often display generic listings, but they lack a social aspect that allows individuals to discover places through shared experiences with people who have similar preferences.

Tabla de contenidos

1. Introducción

Mi enorme amor por la comida y probar platos diferentes, impulsó esta labor. Dándome cuenta, tras muchas idas y venidas, que hay una gran cantidad de cosas en internet sobre restaurantes, pero encontrar sugerencias de fiar y que peguen con tus gustos, es difícil. Tanto comentario anónimo genera desconfianza y la elección del lugar se vuelve un fastidio.

Aunque "TripAdvisor", "Google Maps" y "TheFork" son famosos y se usan bastante, básicamente son directorios con muchas valoraciones. Lo que importa es la cantidad de opiniones, dejando un poco de lado lo social y si a la gente le gustan las mismas cosas que a ti. Por eso, y viendo que no hay una herramienta que se centre en la confianza y en la comunidad para encontrar sitios buenos, nace este proyecto.

La idea principal de esto es crear y diseñar una aplicación web, "GastroLog", para hacer que encontrar un restaurante sea una experiencia más social.
La plataforma simplificará hallar lugares, mediante un sistema construido en una red de confianza, donde los usuarios pueden seguir a amigos o críticos gastronómicos de su agrado. La aplicación no sólo intenta mostrar datos útiles sobre restaurantes, sino también, habilitar un feed de actividad a medida. Así los usuarios miran reseñas y valoraciones de aquellos en quien confían, haciendo la decisión, ya sabes, más personal y certera.

Mediante esta plataforma, se aspira a mejorar la calidad y personalización al descubrir restaurantes, facilitando que los usuarios compartan sus vivencias de una forma más profunda y, en resumidas cuentas, levantar una comunidad confiable para los aficionados a la buena cocina.

1.1 Plan de Trabajo

En este apartado se establecen las metas estratégicas y las acciones requeridas para llevar a cabo el proyecto. Además, se presenta un cronograma detallado mediante un diagrama de Gantt, lo que permitirá monitorear el cumplimiento de plazos establecidos.

1.1.1 Lista de Objetivos

Objetivo1: Desarrollar un sistema de gestión de usuarios robusto y seguro, que incluya registro, autenticación mediante credenciales y gestión de perfiles.

Objetivo2: Implementar una funcionalidad completa para la gestión de restaurantes, permitiendo a los usuarios listarlos, buscarlos con filtros combinados y consultar su información detallada.

Objetivo3: Crear un sistema de reseñas y valoraciones que permita a los usuarios compartir sus opiniones y puntuar los restaurantes.

Objetivo4: Incorporar funcionalidades sociales, como la capacidad de seguir a otros usuarios y visualizar un feed con su actividad reciente.

Objetivo5: Construir una interfaz de usuario intuitiva y accesible, basándose en las maquetas proporcionadas y cumpliendo con los requisitos de usabilidad y accesibilidad.

1.1.2 Lista de Tareas

Se propone una división del trabajo en fases para una mejor organización y seguimiento.

Fase 1: Planificación y Configuración (1-2 Semanas)

- Definición de requisitos funcionales y no funcionales.
- Diseño de la arquitectura de la aplicación y el modelo de datos.
- Creación del plan de trabajo y el cronograma (Diagrama de Gantt).
- **T1.1:** Inicializar el proyecto de Spring Boot con las dependencias necesarias.
- **T1.2:** Crear y configurar el docker-compose.yml para levantar el servicio de PostgreSQL.
- **T1.3:** Estructurar los paquetes del proyecto backend (controller, Service, repository, model, etc.).
- **T1.4:** Preparar la estructura de archivos del proyecto frontend (css/, js/).

Fase 2: Desarrollo del Backend (5-6 Semanas)

- **T2.1 (Módulo de Usuarios):**
    - Crear las entidades Java (User, RelacionUsuario).
    - Implementar la lógica de registro y login con Spring Security (cifrado de contraseñas).
    - Crear los endpoints de la API para la gestión del perfil de usuario (editar, darse de baja, etc.).
- **T2.2 (Módulo de Restaurantes):**
    - Crear las entidades Java (Restaurante, Valoración y Foto).
    - Implementar los servicios y endpoints para buscar, filtrar y obtener detalles de restaurantes.
    - Implementar la lógica para publicar, ver y gestionar reseñas y valoraciones.
- **T2.3 (Módulo Social):**
    - Implementar la lógica y los endpoints para seguir/dejar de seguir a usuarios.

- Crear el endpoint que devuelve el feed de actividad de los usuarios seguidos.

Fase 3: Desarrollo del Frontend (5-6 Semanas)

- **T3.1 (Maquetación y Autenticación):**
  - Diseñar las pantallas de registro, login y la página principal con React
  - Crear el código JavaScript para interactuar con los endpoints de autenticación del backend.
- **T3.2 (Funcionalidad Principal):**
  - Desarrollar la vista de listado de restaurantes.
  - Implementar la lógica JavaScript para llamar a la API de búsqueda y filtros y renderizar los resultados.
  - Crear la vista de detalle del restaurante, incluyendo mapa y listado de reseñas.
- **T3.3 (Funcionalidad Social):**
  - Diseñar la página de perfil de usuario.
  - Implementar la lógica JavaScript para el seguimiento de usuarios y la visualización del feed de actividad.

Fase 4: Integración, Pruebas y Documentación (2-3 Semanas)

- **T4.1:** Realizar pruebas de integración completas, asegurando que el frontend y el backend se comunican correctamente.
- **T4.2:** Realizar pruebas de usabilidad y corregir errores visuales o de flujo.
- **T4.3:** Redactar la memoria final del TFG.
- **T4.4:** Preparar la presentación y defensa del proyecto.

1.1.3 Diagrama de Gantt

En la siguiente Ilustración se muestra el Diagrama de Gantt, en el que se muestra en % la duración de cada fase.

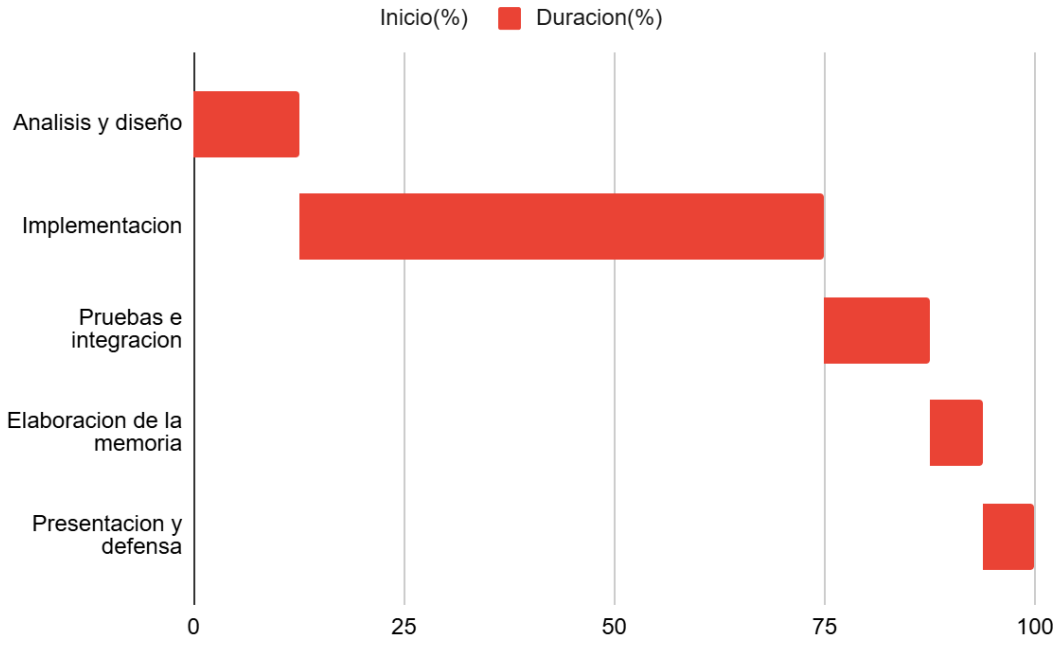


Ilustración 1 Diagrama de Gantt

2 Metodología y Desarrollo

En esta sección se muestra el método de trabajo que se ha empleado en este proyecto, seguidamente tenemos las tecnologías y herramientas utilizadas.

2.1 Metodología

Para la gestión y planificación del proyecto, se ha adoptado el modelo en cascada. Este es un modelo de desarrollo secuencial en el que el proceso se divide en fases sucesivas, donde cada una de las fases debe completarse antes de que comience la siguiente.

Se ha elegido este modelo por los siguientes puntos:

- **Claridad y simplicidad:** Ofrece una estructura clara y bien definida de los pasos a seguir.
- **Facilidad de gestión:** Al ser un proceso lineal, es sencillo de gestionar, documentar y controlar.
- **Requisitos definidos:** Resulta especialmente útil en proyectos de alcance acotado, como el presente TFG, donde los requisitos funcionales están claramente establecidos desde el inicio.

2.2 Tecnologías y Herramientas

En este apartado se describen las tecnologías, herramientas y lenguajes de programación que conforman la tecnología del proyecto, divididos según su rol en la arquitectura.

2.2.1 Tecnologías del Backend

Java y Spring Boot: El backend fue implementado utilizando Java, un lenguaje de programación. robusto y orientado a objetos. Sobre él, se utiliza el framework Spring Boot, que agiliza la creación de aplicaciones autocontenidas y listas para producción. Su elección permite aprovechar un ecosistema maduro que incluye módulos clave para este proyecto como Spring Security (para la autenticación) y Spring Data JPA (para la persistencia de datos).

PostgreSQL (Base de Datos): Como sistema de gestión de base de datos relacional se ha seleccionado PostgreSQL. Sus características más relevantes para el proyecto son:

- Es una solución de código abierto y de alto rendimiento.
- Ofrece una excelente integración con Spring Boot a través de Spring Data JPA.
- Cuenta con soporte para tipos de datos avanzados y herramientas de gestión gráfica como PgAdmin.

Docker: Para garantizar un entorno de desarrollo, pruebas y producción coherente y aislado, el proyecto utiliza Docker. Esta plataforma de contenedores se emplea específicamente para gestionar el servicio de la base de datos PostgreSQL. Mediante un archivo docker-compose.yml, se define y levanta un contenedor que ejecuta la base de datos con una configuración idéntica en cualquier máquina.

2.2.2 Tecnologías del Frontend

React: Es la biblioteca de JavaScript que sirve como núcleo del frontend. Se utiliza para construir la interfaz de usuario completa, gestionando su interactividad, dinamismo y estado.

JavaScript: La interactividad y el dinamismo de la aplicación se implementan con JavaScript. Este lenguaje ejecutado en el lado del cliente permite la manipulación y la comunicación asíncrona con el backend posibilitando la actualización de contenido sin necesidad de recargar la página.

CSS3: Para la estilización visual de los componentes de React, se emplea CSS3 (Cascading Style Sheets). Facilita un control detallado y preciso sobre la apariencia de la aplicación, incluyendo colores, tipografías y diseño, aplicándose directamente a los componentes para definir su aspecto.

2.2.3 Integración con Servicios Externos

OpenStreetMap (Overpass API): Proyecto colaborativo para crear un mapa del mundo libre y editable. Para consultar estos datos, la aplicación utiliza la Overpass API, una API de solo lectura optimizada para consumir partes específicas de la base de datos de OSM mediante consultas.

Uso en el Proyecto: Esta API es el núcleo del motor de búsqueda de restaurantes externos. Permite a los usuarios encontrar establecimientos que aún no existen en nuestra base de datos local.

- **Funcionalidad:** Se realizan consultas en lenguaje *Overpass QL* para filtrar nodos etiquetados como amenity=restaurant dentro de un área geográfica específica.
- **Filtros Implementados:** La integración permite filtrar dinámicamente por nombre del establecimiento (usando expresiones regulares para búsquedas "empieza por"), tipo de cocina (cuisine), dirección y ubicación geográfica.

Nominatim (Geocodificación): Motor de búsqueda de OpenStreetMap que permite realizar geocodificación (convertir direcciones o nombres de lugares en coordenadas geográficas).

Uso en el Proyecto: Actúa como paso previo indispensable para las consultas a la Overpass API.

- **Flujo de Trabajo:** Cuando un usuario busca restaurantes en una ciudad (ej. "Valladolid"), el sistema consulta primero a Nominatim para obtener las coordenadas exactas (latitud/longitud) y el identificador de área (osm_id) de dicha ciudad.

Cloudinary (Gestión de Medios en la Nube): Solución SaaS (*Software as a Service*) líder en la gestión integral de imágenes y videos en la nube. Ofrece almacenamiento seguro, manipulación de imágenes en tiempo real y una red de entrega de contenidos (CDN) optimizada.

Uso en el Proyecto: Se utiliza para externalizar el almacenamiento de todo el contenido multimedia generado por los usuarios, garantizando que la aplicación sea *stateless* (sin estado) y escalable.

- **Almacenamiento:** Las fotos de perfil de los usuarios y las imágenes adjuntas a las reseñas de restaurantes se suben directamente a los servidores de Cloudinary.
- **Base de Datos:** En nuestra base de datos local (PostgreSQL) únicamente almacenamos la URL pública segura (https) devuelta por Cloudinary, evitando así guardar archivos binarios pesados en nuestro propio sistema.

2.2.4 Seguridad del Sistema

Spring Security: Configurado para gestionar la autenticación y el control de acceso.

JWT (JSON Web Token): Cuando un usuario inicia sesión se genera un token que contiene la identidad de ese usuario y fecha de expiración. Se ha implementado JstRequestFilter que intercepta las peticiones HTTP antes de llegar a los controladores.

Gestión de contraseñas: Se ha implementado BCryptPasswordEncoder, sirve para hashear las contraseñas en la base de datos.

Protección de endpoints: Se ha implementado SecurityFilterChain para proteger ciertos endpoints.

Validación de datos: Se han utilizado DTOs para validar las entradas.

3 Análisis de Requisitos

A continuación, se definen las funcionalidades clave del sistema y los requisitos técnicos necesarios para su correcto desempeño. Se abordan tanto las tareas que el sistema debe ejecutar como las condiciones de seguridad, usabilidad y eficiencia que debe cumplir.

3.1 Requisitos Funcionales

ID del requerimiento	RF-01
Nombre del requerimiento	Crear usuario
Descripción	El Sistema crea un usuario
Prioridad	Alta

Tabla 1 Crear usuario

ID del requerimiento	RF-02
Nombre del requerimiento	Iniciar sesión
Descripción	El sistema debe permitir iniciar sesión una vez haya creado el usuario
Prioridad	Alta

Tabla 2 Iniciar sesión

ID del requerimiento	RF-03
Nombre del requerimiento	Editar perfil
Descripción	El Sistema debe permitir que un usuario edite los datos de su perfil
Prioridad	Alta

Tabla 3 Editar Perfil

ID del requerimiento	RF-04
Nombre del requerimiento	Listar restaurantes.

Descripción	La aplicación debe dispones de una pantalla en la que los usuarios puedan consultar un listado de restaurantes
Prioridad	Alta

Tabla 4 Listar Restaurantes

ID del requerimiento	RF-05
Nombre del requerimiento	Consultar restaurante
Descripción	La aplicación debe permitir a los usuarios buscar restaurantes por nombre, ciudad, tipo de cocina o puntuación media, todos estos se pueden combinar
Prioridad	Alta

Tabla 5 Consultar Restaurante

ID del requerimiento	RF-06
Nombre del requerimiento	Detalles restaurante
Descripción	La aplicación debe permitir a los usuarios acceder a la ficha de un restaurante con información detallada, incluyendo su ubicación en un mapa
Prioridad	Alta

Tabla 6 Detalles del restaurante

ID del requerimiento	RF-07
Nombre del requerimiento	Consultar reseñas
Descripción	La aplicación debe permitir a los usuarios consultar reseñas y valoración publicadas por otros usuarios en cada restaurante
Prioridad	Alta

Tabla 7 Consultar reseñas

ID del requerimiento	RF-08
Nombre del requerimiento	Publicar reseñas
Descripción	La aplicación debe permitir a los usuarios publicar sus reseñas sobre restaurantes
Prioridad	Alta

Tabla 8 Publicar reseña

ID del requerimiento	RF-09
Nombre del requerimiento	Valorar restaurante
Descripción	La aplicación debe permitir a los usuarios valorar un restaurante mediante una puntuación numérica de 1 a 5
Prioridad	Alta

Tabla 9 Valorar Restaurante

ID del requerimiento	RF-10
Nombre del requerimiento	Seguir usuario
Descripción	La aplicación debe permitir a los usuarios seguir a otros usuarios registrados.
Prioridad	Alta

Tabla 10 Seguir usuario

ID del requerimiento	RF-11
Nombre del requerimiento	Ver actividad usuarios seguidos
Descripción	La aplicación debe disponer de una pantalla en la que los usuarios puedan visualizar la actividad de los usuarios a los que siguen
Prioridad	Alta

Tabla 11 Ver actividad de usuarios seguidos

ID del requerimiento	RF-12
Nombre del requerimiento	Darse de baja
Descripción	La aplicación debe permitir eliminar la cuenta
Prioridad	Alta

Tabla 12 Darse de baja

ID del requerimiento	RF-13
Nombre del requerimiento	Añadir favoritos
Descripción	La aplicación debe permitir a los usuarios añadir restaurantes a favoritos
Prioridad	Alta

Tabla 13 Añadir a favoritos

ID del requerimiento	RF-14
Nombre del requerimiento	Eliminar favoritos
Descripción	La aplicación debe permitir a los usuarios eliminar de favoritos restaurantes
Prioridad	Alta

Tabla 14 Eliminar de favoritos

ID del requerimiento	RF-15
Nombre del requerimiento	Contraseña olvidada
Descripción	La aplicación debe permitir a los usuarios que cambie su contraseña si se le ha olvidado
Prioridad	Alta

Tabla 15 Contraseña olvidada

ID del requerimiento	RF-16
Nombre del requerimiento	Ver mis favoritos
Descripción	La aplicación debe permitir a los usuarios poder ver sus restaurantes favoritos
Prioridad	Alta

Tabla 16 Ver mis favoritos

ID del requerimiento	RF-17
Nombre del requerimiento	Ver seguidores y seguidos de otro usuario
Descripción	La aplicación debe permitir a los usuarios poder ver los seguidores y seguidos de otro usuario

Prioridad	Alta

Tabla 17 Ver seguidores y seguidos de otro usuario

ID del requerimiento	RF-18
Nombre del requerimiento	Ver a quien sigo y quien me sigue
Descripción	La aplicación debe permitir a los usuarios poder ver a quien siguen y quien le sigue
Prioridad	Alta

Tabla 18 Ver a quien sigo y quien me sigue

ID del requerimiento	RF-19
Nombre del requerimiento	Ver solicitudes pendientes
Descripción	La aplicación debe permitir a los usuarios ver las solicitudes de amistad pendientes
Prioridad	Alta

Tabla 19 Ver solicitudes pendientes

ID del requerimiento	RF-20
Nombre del requerimiento	Eliminar seguidor
Descripción	La aplicación debe permitir a los usuarios eliminar un seguidor suyo
Prioridad	Alta

Tabla 20 Eliminar seguidor

ID del requerimiento	RF-21
Nombre del requerimiento	Dejar de seguir
Descripción	La aplicación debe permitir a los usuarios dejar de seguir a otro usuario
Prioridad	Alta

Tabla 21 Dejar de seguir

ID del requerimiento	RF-22

Nombre del requerimiento	Eliminar una reseña
Descripción	La aplicación debe permitir a los usuarios eliminar una reseña suya
Prioridad	Alta

Tabla 22 Eliminar una reseña

ID del requerimiento	RF-23
Nombre del requerimiento	Cerrar Sesión
Descripción	La aplicación debe permitir a los usuarios cerrar la sesión actual
Prioridad	Alta

Tabla 23 Cerrar Sesión

3.2 Requisitos No Funcionales

ID del requerimiento	RNF-01
Nombre del requerimiento	Seguridad
Descripción	La aplicación debe garantizar la seguridad de los datos de los usuarios mediante autenticación segura y cifrado de contraseñas.
Prioridad	Alta

Tabla 24 Seguridad

ID del requerimiento	RNF-02
Nombre del requerimiento	Privacidad
Descripción	La aplicación debe garantizar la privacidad de los usuarios
Prioridad	Alta

Tabla 25 Privacidad

ID del requerimiento	RNF-03

Nombre del requerimiento	Rendimiento
Descripción	La aplicación debe garantizar un tiempo de respuesta aceptable, de manera que la mayoría de las pantallas carguen en menos de 2 segundos bajo condiciones normales
Prioridad	Alta

Tabla 26 Rendimiento

ID del requerimiento	RNF-04
Nombre del requerimiento	Escalabilidad
Descripción	La aplicación debe ser escalable, de forma que pueda soportar un incremento progresivo de usuarios, restaurantes y reseñas sin pérdida significativa de rendimiento.
Prioridad	Alta

Tabla 27 Escalabilidad

ID del requerimiento	RNF-05
Nombre del requerimiento	Usabilidad
Descripción	La aplicación debe ser usable, ofreciendo una interfaz clara e intuitiva que facilite la navegación y el acceso a las principales funcionalidades.
Prioridad	Alta

Tabla 28 Usabilidad

ID del requerimiento	RNF-06
Nombre del requerimiento	Accesibilidad
Descripción	La aplicación debe ser accesible desde distintos dispositivos y navegadores, adaptando su diseño a pantallas de ordenador, Tablet y móvil.
Prioridad	Alta

Tabla 29 Accesibilidad

ID del requerimiento	RNF-07
Nombre del requerimiento	Accesibilidad de imagen
Descripción	La aplicación debe cumplir con las pautas básicas de accesibilidad, proporcionando suficiente contraste de colores, texto alternativo en imágenes y navegación sencilla.
Prioridad	Media

Tabla 30 Accesibilidad de imagen

ID del requerimiento	RNF-08
Nombre del requerimiento	Disponibilidad
Descripción	La aplicación debe garantizar la disponibilidad de los datos mediante la realización de copias de seguridad periódicas.
Prioridad	Alta

Tabla 31 Disponibilidad

ID del requerimiento	RNF-09
Nombre del requerimiento	Integridad
Descripción	La aplicación debe garantizar la integridad de la información almacenada, evitando pérdidas o modificaciones no autorizadas de datos.
Prioridad	Alta

Tabla 32 Integridad

3.3 Casos de Uso

CU-01 - Registrarse

ID	CU-01
Requisitos Asociados	RF-01
Descripción	Permite a un usuario crear una cuenta en el sistema para poder autenticarse posteriormente y

	acceder a las funcionalidades reservadas.
Precondición	El usuario no debe tener una cuenta registrada en el sistema.
Flujo Normal	1. El actor accede a la página de Registro. 2. Introduce nombre, correo electrónico y contraseña. 3. El sistema valida que los datos cumplen los requisitos. 4. El sistema crea la cuenta y confirma el registro al usuario.
Flujos Alternativos	Tras el paso 3: 1. El sistema detecta que el correo ya está registrado. 2. Muestra el mensaje 'El correo ya existe'. 3. Ofrece volver a intentar desde el paso 2.
Postcondición	La cuenta de usuario queda registrada en el sistema. El usuario puede iniciar sesión con sus credenciales.

Tabla 33 Caso de Uso Registrarse

CU-02 - Iniciar sesión

ID	CU-02
Requisitos Asociados	RF-02
Descripción	Permite a un usuario autenticarse en el sistema introduciendo sus credenciales válidas.
Precondición	El usuario debe tener una cuenta registrada.
Flujo Normal	1. El usuario accede a la página de inicio de sesión. 2. Introduce correo electrónico y contraseña. 3. El sistema verifica las credenciales ingresadas 4. Si son válidas, se inicia la sesión y el usuario es redirigido a la pantalla principal.
Flujos Alternativos	Tras la verificación, el sistema identifica que las credenciales son correctas, se muestra el mensaje “Usuario o contraseña incorrectos”. Se permite al usuario volver a

	intentar el inicio de sesión desde el primer paso.
Postcondición	Sesión activa creada y usuario autenticado en el sistema.

Tabla 34 Caso de uso Iniciar sesión

CU-03 - Editar perfil

ID	CU-03
Requisitos Asociados	RF-03
Descripción	Permite a un usuario modificar la información de su perfil, como nombre, foto o biografía.
Precondición	El usuario tiene estar autenticado.
Flujo Normal	1. El usuario accede a la pantalla de Perfil. 2. Selecciona la opción de Editar perfil. 3. Modifica los datos deseados. 4. El sistema valida los cambios y los guarda.
Flujos Alternativos	Si los datos no cumplen los requisitos de validación, se muestra un mensaje de error y no se guardan los cambios, el nombre de usuario no puede estar vacío.
Postcondición	Los datos del perfil se actualizan correctamente en el sistema.

Tabla 35 Caso de Uso editar Perfil

CU-04 - Consultar lista de restaurantes

ID	CU-04
Requisitos Asociados	RF-04
Descripción	Permite a un usuario visualizar un listado de restaurantes disponibles en el sistema.
Precondición	Debe existir al menos un restaurante registrado en la base de datos.
Flujo Normal	1. El usuario accede a la pantalla de Restaurantes. 2. El sistema muestra un listado paginado de restaurantes con información básica (nombre, tipo de comida, puntuación).

Flujos Alternativos	Si no existen restaurantes registrados, se muestra un mensaje informando que no hay resultados.
Postcondición	El usuario visualiza correctamente el listado de restaurantes.

Tabla 36 Caso de Uso Listar restaurantes

CU-05 - Buscar restaurante

ID	CU-05
Requisitos Asociados	RF-05
Descripción	Permite a un usuario buscar restaurantes introduciendo un criterio (nombre, categoría, ciudad, rango de precio o puntuación media).
Precondición	Debe haber restaurantes registrados en el sistema.
Flujo Normal	1. El usuario introduce un término de búsqueda. 2. El sistema procesa la búsqueda y muestra resultados coincidentes.
Flujos Alternativos	Si no hay coincidencias, se muestra un mensaje indicando que no se encontraron resultados.
Postcondición	El usuario visualiza el listado de resultados coincidentes con su búsqueda.

Tabla 37 Caso de Uso Buscar Restaurante

CU-06 - Ver ficha de restaurante

ID	CU-06
Requisitos Asociados	RF-06
Descripción	Permite a un usuario acceder a la ficha detallada de un restaurante con datos, mapa y reseñas.
Precondición	El restaurante debe existir en el sistema.
Flujo Normal	1. El usuario selecciona un restaurante de la lista. 2. El sistema muestra la ficha del

	restaurante con información detallada, ubicación en el mapa y reseñas.
Flujos Alternativos	Si el restaurante no existe o fue eliminado, se muestra un mensaje de error.
Postcondición	El usuario visualiza la información detallada del restaurante seleccionado.

Tabla 38 Caso de Uso Ver ficha de un restaurante

CU-7 - Consultar reseñas

ID	CU-07
Requisitos Asociados	RF-07
Descripción	Permite a un usuario visualizar las reseñas publicadas por otros usuarios en la ficha de un restaurante.
Precondición	El restaurante debe existir y tener reseñas publicadas.
Flujo Normal	1. El usuario accede a la ficha del restaurante. 2. El sistema muestra la lista de reseñas paginadas.
Flujos Alternativos	Si no existen reseñas, se muestra un mensaje indicando que no hay comentarios disponibles.
Postcondición	El usuario visualiza correctamente las reseñas del restaurante.

Tabla 39 Caso de Uso Consultar reseña

CU-08 - Escribir reseña

ID	CU-08
Requisitos Asociados	RF-08
Descripción	Permite a un usuario autenticado publicar una reseña sobre un restaurante.

Precondición	El usuario debe estar autenticado y el restaurante debe existir.
Flujo Normal	1. El usuario accede a la ficha de un restaurante. 2. Selecciona la opción de escribir reseña. 3. Introduce el texto de la reseña y opcionalmente imágenes. 4. El sistema valida y publica la reseña.
Flujos Alternativos	Si el usuario no está autenticado, se solicita iniciar sesión. Si el contenido no cumple requisitos de validación, se rechaza la publicación, solo se puede escribir una reseña de ese restaurante por usuario.
Postcondición	La reseña queda registrada y asociada al usuario y restaurante.

Tabla 40 Caso de Uso Escribir reseña

CU-09 - Valorar restaurante

ID	CU-09
Requisitos Asociados	RF-09
Descripción	Permite a un usuario dar una puntuación numérica a un restaurante.
Precondición	El usuario debe estar autenticado y el restaurante debe existir.
Flujo Normal	1. El usuario accede a la ficha del restaurante. 2. Selecciona una puntuación (1 a 5 estrellas). 3. El sistema guarda la valoración y actualiza la media.
Flujos Alternativos	Si el usuario ya valoró el restaurante, la puntuación anterior se actualiza.
Postcondición	La puntuación del usuario queda registrada y la media actualizada.

Tabla 41 Caso de Uso Valorar restaurante

CU-10 - Seguir a usuario

ID	CU-10
Requisitos Asociados	RF-10
Descripción	Permite a un usuario autenticado seguir a otro usuario registrado en el sistema.
Precondición	El usuario debe estar autenticado y el usuario a seguir debe existir.
Flujo Normal	1. El usuario accede al perfil de otro usuario. 2. Selecciona la opción Seguir. 3. El sistema registra la relación de seguimiento.
Flujos Alternativos	Si ya sigue a ese usuario, la acción se convierte en Dejar de seguir.
Postcondición	El usuario seguido aparece en la lista de seguidos del actor.

Tabla 42 Caso de Uso Seguir usuario

CU-11 - Ver actividad de seguidos

ID	CU-11
Requisitos Asociados	RF-11
Descripción	Permite a un usuario autenticado visualizar las últimas actividades (reseñas y valoraciones) de los usuarios a los que sigue.
Precondición	El usuario debe estar autenticado y seguir al menos a un usuario.
Flujo Normal	1. El usuario accede a la pantalla de Actividad. 2. El sistema muestra las reseñas y valoraciones de los usuarios seguidos, ordenadas cronológicamente.

Flujos Alternativos	Si el usuario no sigue a nadie, se muestra un mensaje indicando que no hay actividad disponible.
Postcondición	El usuario visualiza el feed con la actividad de las personas que sigue.

Tabla 43 Caso de Uso Ver actividad de seguidos

CU-12 – Darse de baja

ID	CU-12
Requisitos Asociados	RF-12
Descripción	Permite a un usuario eliminar su propia cuenta
Precondición	El usuario debe tener cuenta
Flujo Normal	1. El usuario accede a su perfil 2. Pulsa las tres rayas 3. Pulsa la opción de eliminar cuenta 4. El sistema te dirige a la página de inicio
Flujos Alternativos	No aplica
Postcondición	El usuario no tiene acceso a su cuenta

Tabla 44 Caso de Uso Darse de baja

CU-13-Añadir a favoritos

ID	CU-13
Requisitos Asociados	RF-13
Descripción	Permite a un usuario añadir restaurantes a favoritos
Precondición	El usuario debe tener cuenta
Flujo Normal	1. El usuario accede a la ficha de un restaurante 2. Pulsa añadir favoritos
Flujos Alternativos	No aplica
Postcondición	El restaurante aparece en favoritos del usuario

Tabla 45 Caso de Uso Añadir a favoritos

CU-14 – Eliminar de favoritos

ID	CU-14
Requisitos Asociados	RF-14
Descripción	Permite a un usuario eliminar restaurantes de sus favoritos, y tener ese restaurante en sus favoritos
Precondición	El usuario debe tener cuenta
Flujo Normal	1. El usuario accede a la ficha de un restaurante 2. Pulsa eliminar de favoritos
Flujos Alternativos	No aplica
Postcondición	El restaurante ya no aparece en sus favoritos

Tabla 46 Caso de Uso Eliminar de favoritos

CU-15 – Contraseña olvidada

ID	CU-15
Requisitos Asociados	RF-15
Descripción	Permite a un usuario editar su contraseña si se le ha olvidado
Precondición	El usuario debe tener cuenta
Flujo Normal	1. El usuario pulsa el botón de contraseña olvidada 2. Accede a su correo
Flujos Alternativos	No aplica
Postcondición	El usuario inicia sesión con esa contraseña

Tabla 47 Caso de Uso Contraseña olvidada

CU-16 – Ver mis favoritos

ID	CU-16
Requisitos Asociados	RF-16
Descripción	Permite a un usuario ver sus restaurantes favoritos
Precondición	El usuario debe tener cuenta
Flujo Normal	1. El usuario pulsa el botón de mis favoritos
Flujos Alternativos	No aplica

Postcondición	El usuario entra en la nueva pantalla y ves sus restaurantes favoritos.

Tabla 48 Caso de Uso Ver mis favoritos

CU-17 – Ver seguidores y seguidos de otro usuario

ID	CU-17
Requisitos Asociados	RF-17
Descripción	Permite a un usuario ver los seguidores y seguidos de otro usuario
Precondición	El usuario debe seguir al usuario
Flujo Normal	1. El usuario solicita al otro usuario 2. El otro usuario lo acepta 3. El usuario ver los seguidores y seguidos de ese usuario
Flujos Alternativos	No aplica
Postcondición	No aplica

Tabla 49 Caso de Uso Ver seguidores y seguidos de otro usuario

CU-18 – Ver a quien sigo y quien me sigue

ID	CU-18
Requisitos Asociados	RF-18
Descripción	Permite a un usuario ver a quien sigue y quien le sigue
Precondición	El usuario debe tener cuenta
Flujo Normal	1. El usuario pulsa el botón de seguidores o seguidos 2. Accede a la lista de usuarios
Flujos Alternativos	No aplica
Postcondición	No aplica

Tabla 50 Caso de Uso Ver a quien sigo y quien me sigue

CU-19 – Ver solicitudes pendientes

ID	CU-19
Requisitos Asociados	RF-19

Descripción	Permite a un usuario ver las solicitudes de amistad pendientes
Precondición	El usuario debe tener cuenta
Flujo Normal	1. El usuario pulsa el botón de solicitudes 2. La aplicación permite aceptar o denegar esa solicitud
Flujos Alternativos	No aplica
Postcondición	El usuario acepta o rechaza la solicitud

Tabla 51 Caso de Uso Ver solicitudes pendientes

CU-20 – Eliminar seguidor

ID	CU-20
Requisitos Asociados	RF-20
Descripción	Permite a un usuario eliminar un seguidor
Precondición	El usuario debe tener cuenta y le tiene que seguir alguien
Flujo Normal	1. El usuario pulsa el botón de seguidores 2. Pulsa el botón de eliminar
Flujos Alternativos	No aplica
Postcondición	Ya no le tiene como seguidor

Tabla 52 Caso de Uso Eliminar seguidor

CU-21 – Dejar de seguir

ID	CU-21
Requisitos Asociados	RF-21
Descripción	Permite a un usuario dejar de seguir a un usuario
Precondición	El usuario debe tener cuenta y tiene que seguir a ese usuario
Flujo Normal	1. El usuario pulsa el botón del perfil del otro usuario 2. Le deja de seguir
Flujos Alternativos	No aplica
Postcondición	Ya no le sigue

Tabla 53 Caso de Uso Dejar de seguir

CU-22 – Eliminar una reseña

ID	CU-22
Requisitos Asociados	RF-22
Descripción	Permite a un usuario eliminar una reseña suya
Precondición	El usuario debe tener cuenta y haber hecho una valoración de un restaurante
Flujo Normal	1. El usuario accede a la reseña 2. Elimina la reseña
Flujos Alternativos	No aplica
Postcondición	Ya no le sale la reseña

Tabla 54 Caso de Uso Eliminar reseña

CU-23 – Cerrar sesión Usuario

ID	CU-23
Requisitos Asociados	RF-23
Descripción	Permite a un usuario cerrar sesión
Precondición	El usuario debe tener cuenta
Flujo Normal	1. El usuario pulsa el botón de cerrar sesión
Flujos Alternativos	Recarga la pagina
Postcondición	No aplica

Tabla 55 Caso de Uso Cerrar sesión usuario

3.4 Matriz de trazabilidad

Requisito Funcional	**Descripción breve**	**Casos de Uso Asociados**	**Observaciones**
RF-01	Registro de usuarios	**CU-01 Registrarse**	Obligatorio
RF-02	Autenticación (login)	**CU-02 Iniciar sesión**	Obligatorio
RF-03	Visualización/edición del perfil	**CU-03 Editar perfil**	Obligatorio

Requisito Funcional	Descripción breve	Casos de Uso Asociados	Observaciones
RF-04	Listado de restaurantes	**CU-04 Listar restaurantes**	Obligatorio
RF-05	Búsqueda de restaurantes	**CU-05 Buscar restaurante**	Obligatorio
RF-06	Ficha de restaurante con mapa	**CU-06 Ver detalles restaurante**	Obligatorio
RF-07	Consultar reseñas	**CU-07 Consultar reseña**	Obligatorio
RF-08	Publicar reseñas	**CU-08 Publicar reseña**	Obligatorio
RF-09	Valorar restaurante	**CU-09 Valorar restaurante**	Obligatorio
RF-10	Seguir usuarios	**CU-10 Seguir usuario**	Opcional (social)
RF-11	Ver actividad de seguidos	**CU-11 Ver actividad usuarios seguidos**	Opcional (social)
RF-12	Darse de baja	**CU-12 Darse de baja**	Obligatorio
RF-13	Añadir a favoritos	**CU-13 Añadir a favoritos**	Obligatorio
RF-14	Eliminar de favoritos	**CU-14 Eliminar de favoritos**	Obligatorio
RF-15	Contraseña olvidada	**CU-15 Contraseña olvidada**	Obligatorio
RF-16	Ver mis favoritos	**CU-16 Ver mis favoritos**	Obligatorio
RF-17	Ver seguidores y seguidos de otro usuario	**CU-17 Ver seguidores y seguidos de otro usuario**	Obligatorio
RF-18	Ver a quien sigo y quien me sigue	**CU-18 Ver a quien sigo y quien me sigue**	Obligatorio
RF-19	Ver solicitudes pendientes	**CU-19 Ver solicitudes pendientes**	Obligatorio
RF-20	Eliminar seguidor	**CU-20 Eliminar seguidor**	Obligatorio

Requisito Funcional	Descripción breve	Casos de Uso Asociados	Observaciones
RF-21	Dejar de seguir	**CU-21 Dejar de seguir**	Obligatorio
RF-22	Eliminar reseña	**CU-22 Eliminar reseña**	Obligatorio
RF-23	Cerrar Sesión	**CU-23 Cerrar sesión usuario**	Obligatorio

Tabla 56 Matriz de trazabilidad

4 Diseño

Aquí se detalla la planificación del diseño del proyecto. El objetivo es estructurar de forma precisa cómo funcionará el sistema, sirviendo como hoja de ruta fundamental para su posterior puesta en marcha.

4.1 Arquitectura Global

El sistema se ha desarrollado siguiendo una **arquitectura cliente-servidor de tres capas**, un patrón de diseño estándar que separa la aplicación en tres niveles lógicos y físicos: la capa de presentación, la capa de lógica de negocio y la capa de datos. Adicionalmente, el backend implementa el patrón

Modelo-Vista-Controlador (MVC) para organizar su lógica interna.

Las tres capas principales son:

- **Capa de Presentación (Frontend):** Es la interfaz de usuario con la que interactúa el cliente. Se encarga de mostrar los datos y capturar las acciones del usuario, comunicándose con el backend a través de una API REST.
- **Capa de Lógica (Backend):** Contiene la lógica de negocio de la aplicación. Aquí es donde el patrón MVC organiza el código:
    - **Controlador:** Recibe las peticiones HTTP del cliente, invoca a los servicios correspondientes y devuelve una respuesta.
    - **Servicio:** Orquesta la lógica de negocio principal de la aplicación.
    - **Modelo:** Define la estructura de los datos (Entidades JPA) con la que operan los servicios.
- **Capa de Datos (Backend):** Es la responsable de la persistencia y el acceso a los datos, gestionada a través de repositorios que interactúan con la base de datos PostgreSQL.

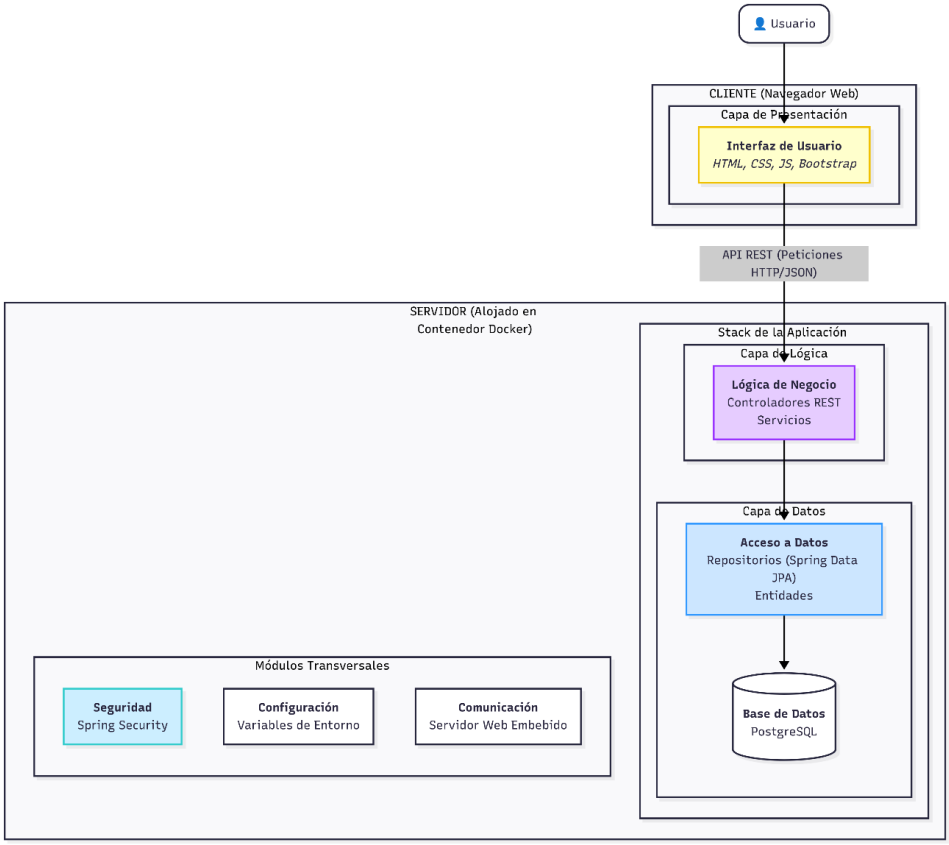


Ilustración 2 Arquitectura

El siguiente diagrama resume el flujo de una petición HTTP en esta arquitectura:

1. El **usuario** interactúa con la **Capa de Presentación (Frontend)** en su navegador, desencadenando una petición HTTP.
2. El **Frontend (JavaScript)** envía la petición a la URL correspondiente de la **API REST** del backend.
3. El **Controlador (Backend)** recibe la petición, interpreta la solicitud y llama al método apropiado en la **capa de Servicios**.
4. El **Servicio** ejecuta la lógica de negocio necesaria y solicita los datos a la **Capa de Datos** a través de un **Repositorio (JPA)**.
5. El **Repositorio**, mediante el ORM (Hibernate), traduce la llamada a una consulta SQL, la ejecuta en la base de datos **PostgreSQL** y devuelve los resultados como objetos del **Modelo** (Entidades).
6. Los datos regresan a través del Servicio hasta el Controlador.
7. Finalmente, el **Controlador** empaqueta los datos en formato **JSON** y los envía de vuelta al cliente en una respuesta HTTP, que el **Frontend** utiliza para actualizar la vista y mostrar la información al usuario.

4.2 Modelo de Datos

En este apartado se presentan las distintas Entidades y el diagrama correspondiente.

4.2.1 Entidades

Campo	**Tipo**	**Restricción**
id	bigint	Primary Key
nombreUsuario	varchar(255)	Unique, Not Null
email	varchar(255)	Unique, Not Null
pwd	varchar(255)	Not Null
fechaBaja	timestamp	Nullable
fotoPerfilUrl	Varchar(255)	Nullable
resetToken	Varchar(25)	Nullable
resetTokenExpiry	timeStamp	Nullable
verificado	boolean	Not Null, Default:false
tokenVerificacion	Varchar(255)	Unique
isPrivate	boolean	Default:false

Tabla 57 Usuario

Campo	**Tipo**	**Restricción**
id	bigint	Primary Key
nombre	varchar(255)	Not Null
ciudad	varchar(255)	Not Null
dirección	varchar(512)	Not Null
tipoCocina	varchar(255)	Nullable
osmId	bigint	Unique
lat	float	Not Null
lon	float	Not Null

Tabla 58 Restaurante

Campo	Tipo	Restricción
id	bigint	Primary Key
comentario	text	Not Null
Puntuación	int	Not Null
fechaCreacion	timestamp	Not Null
		Unique(usuario_id, restaurante_id)

Tabla 59 Valoración

Campo	Tipo	Restricción
Id	bigint	Primary Key
estado	varchar(50)	Not Null
		Primary Key(seguidor_id, seguido_id)

Tabla 60 RelacionUsuario

Campo	Tipo	Restricción
Id	bigint	Primary Key
fechaAgregado	timestamp	Not Null
		Primary Key(ususrio_id, restaurante_id)

Tabla 61 Favorito

Campo	Tipo	Restricción
Id	bigint	Primary Key
url	Varchar(512)	Not Null
		Primary Key(usuario_id, restaurante_id)

Tabla 62 Foto

Explicación de las relaciones entre tablas:

Usuario ↔ Restaurante (a través de Valoración)

N Usuario — M Restaurante

Cada Usuario puede escribir múltiples Valoraciones, y cada Restaurante puede recibir múltiples Valoraciones.

La tabla Valoración es la tabla intermedia que une a un Usuario con un Restaurante. Almacena los datos de esa unión (la puntuación, el comentario) y

usa una clave primaria compuesta para asegurar que un usuario solo pueda valorar un restaurante una sola vez.

Valoración ↔ Foto

1 Valoración — N Foto

Cada Valoración (una reseña específica) puede tener múltiples Fotos adjuntas. Cada Foto individual pertenece a una sola Valoración**.**

Usuario ↔ Restaurante (a través de Favorito)

N Usuario — M Restaurante

Cada Usuario puede marcar múltiples Restaurantes como favoritos, y cada Restaurante puede ser un favorito para múltiples Usuarios.

La tabla Favorito es la tabla intermedia que simplemente registra esta relación (quién marcó qué).

Usuario ↔ Usuario (a través de RelacionUsuario)

N Usuario — M Usuario

Esta es una relación de "seguir". Un Usuario puede seguir a múltiples Usuarios, y un Usuario puede ser seguido por múltiples Usuarios.

La tabla RelacionUsuario es la tabla intermedia que gestiona esto. Tiene dos columnas que apuntan a Usuario: seguidor_id (quién hace la acción) y seguido_id (quién la recibe).

4.2.2 Modelo Entidad Relación

En la siguiente Ilustración podemos ver el modelo empleado en el TFG.

Encontramos las distintas entidades explicadas anteriormente con sus atributos, también se encuentran las relaciones de cada entidad. Un resumen del flujo de datos sería el siguiente:

1. Un usuario se registra.
2. Busca un restaurante.
3. Crea una valoración de ese restaurante y tiene la opción de poner una foto.
4. Si le gusta ese restaurante u otro, lo añade a Favoritos.
5. Si quiere ver lo que comen otros le solicita (RelacionUsuario).

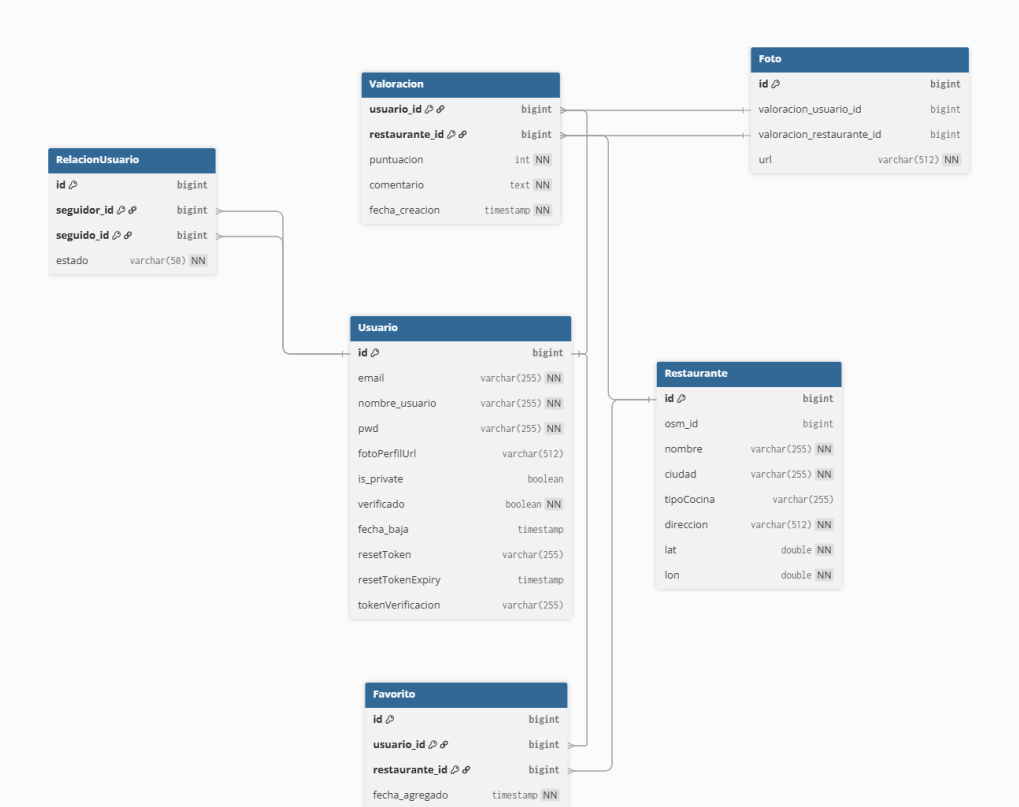


Ilustración 3 Modelo ERD

4.3 Vistas

En relación con las vistas, es lo equivalente a los Endpoints definidos en el "Controller".

Acciones Publicas: Registrar Usuario, Iniciar sesión, Verificar una nueva cuenta, Solicitar restablecimiento de contraseña, Restablecer la contraseña (con token), Buscar Restaurantes (en base de datos local), obtener restaurante por ID, buscar restaurantes (API externa).

Gestión perfil: Obtener mi perfil completo, Actualizar foto de perfil, Actualizar nombre de usuario, Actualizar email, Cambiar contraseña, Darse de baja, Cerrar sesión.

Relaciones Sociales: Enviar Solicitud de seguimiento, Aceptar solicitud, Rechazar solicitud, Eliminar Seguidor, Ver solicitudes pendientes, Ver mis seguidores, Ver a quien sigo, Ver seguidores de otro usuario, Ver a quien sigue otro usuario, Dejar de seguir, Buscar usuarios, Ver perfil de otro usuario.

Valoraciones y Favoritos: Crear Valoración, Añadir restaurante a favoritos, Eliminar restaurante de favoritos, Obtener mi lista de favoritos, Obtener el feed de reseñas de usuarios seguidos, Eliminar valoración.

Administración: Obtener todos los usuarios, Obtener un usuario por ID, Crear un restaurante manualmente, sincronizar desde la API externa, Obtener todas las valoraciones.

Documentación Swagger, abrir en swagger editor.[1]

4.4 Templates

El diseño de la interfaz busca definir la arquitectura de navegación mediante una serie de prototipos o plantillas, especificando su estructura y los elementos de interacción disponibles para el usuario. Asimismo, se detallan los casos de uso derivados de dichas interacciones. Como referencia visual, se incluye un esquema de navegación; el primero de ellos ilustra el flujo de trabajo completo de un perfil administrador, desde el inicio de sesión hasta el cierre de esta.

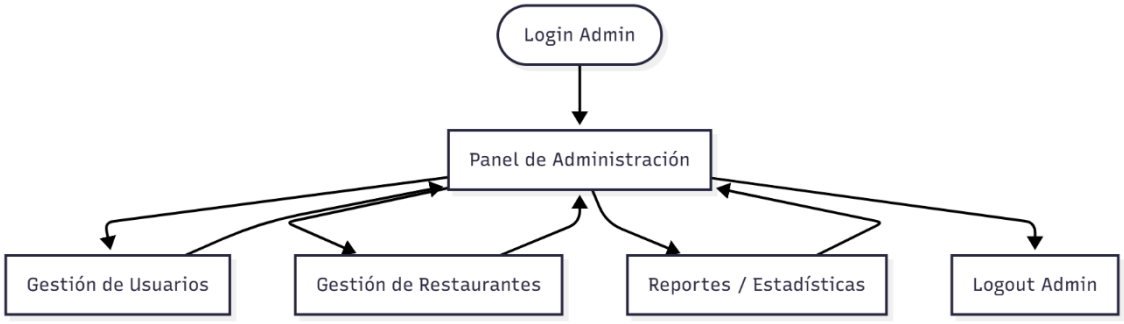


Ilustración 4 Flujo Administrador

Una vez validado el acceso, el usuario puede optar por los distintos recorridos dentro de la aplicación que se ilustran a continuación.

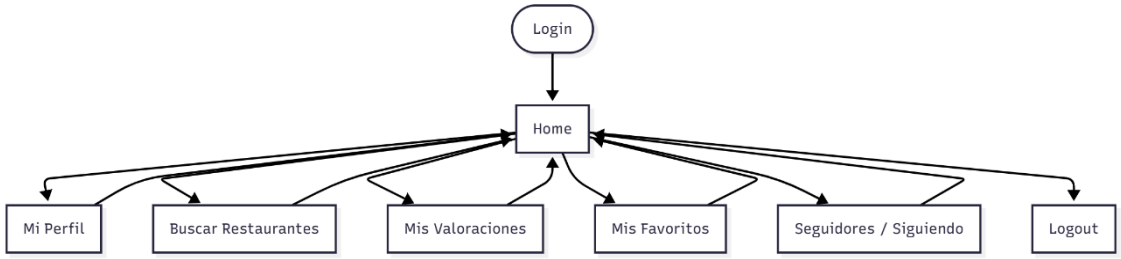


Ilustración 5 Flujo Usuario

[1] https://raw.githubusercontent.com/adriannogy/TFG/refs/heads/master/swagger.yaml

4.4.1 Templates Usuario

El usuario una vez dentro del sistema se encontrará con la página principal. Encontramos una imagen principal en la que se verán las reseñas publicadas por usuarios a los que sigue. A su vez debajo tendrá la opción de buscar un restaurante, seguir a otros usuarios o entrar en su perfil.

Ilustración 6 Flujo Página Principal

En la siguiente ilustración vemos unos parámetros de búsqueda, es decir, la opción de buscar restaurante, los parámetros que vemos son un prototipo, es decir, que pueden variar.

Ilustración 7 Filtro de Búsqueda

Seguidamente tenemos el restaurante que ha buscado el usuario, podemos ver los filtros de los que se compone el Restaurante, una serie de reseñas de usuarios, y opciones de añadir o eliminar de favoritos, además de un mapa para ver donde se encuentra.

Ilustración 8 Plantilla de Restaurante

A continuación, encontramos el perfil del usuario, donde tenemos la opción de arriba a la izquierda en la que un usuario pude hacer acciones dentro de su perfil como editar. Encontramos sus propias reseñas, las solicitudes de amistad y los favoritos que tiene.

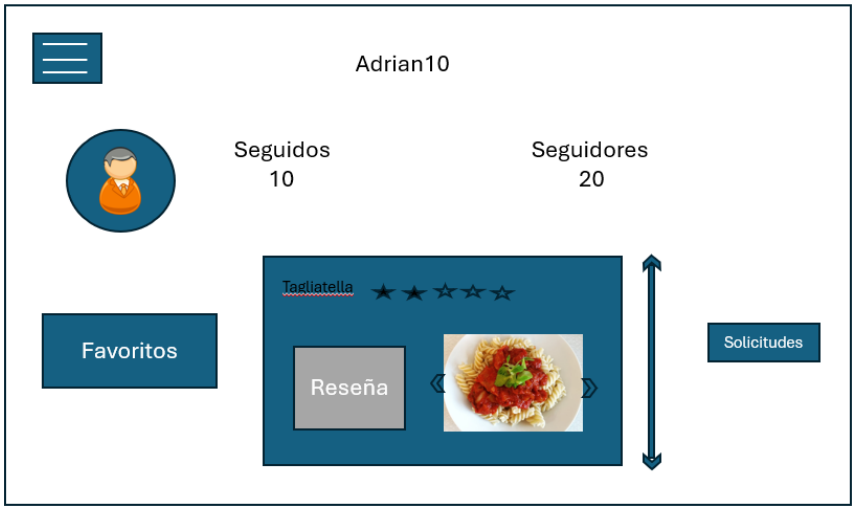


Ilustración 9 Perfil Usuario

Por último, encontramos la pantalla por la cual se puede hacer una reseña, valorando el restaurante, escribiendo la propia reseña y añadiendo fotos relacionadas.

Ilustración 10 Hacer Reseña

5 Desarrollo

En este apartado veremos el resultado de la aplicación web con sus distintas funcionalidades.

También encontramos el enlace directo al código fuente.

https://github.com/adriannogy/TFG

En la siguiente ilustración tenemos el inicio de la aplicación dando la bienvenida.

Ilustración 11 Pantalla de inicio

Seguidamente tenemos la opción de iniciar sesión (cuenta ya registrada) o la opción de registrase.

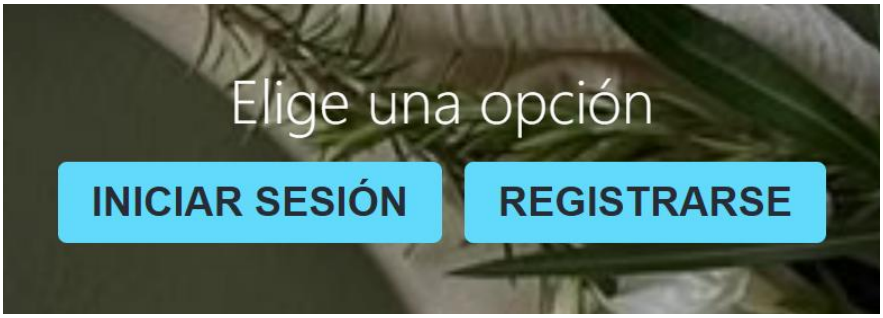


Ilustración 12 Opciones de aplicación

En la opción de registro, tenemos que rellenar los siguientes campos, confirmando la contraseña por temas de seguridad, después de esto, accederemos al correo donde nos vendrá un enlace para verificar la cuenta, si no se accede a ese link, la cuenta no esta verificada y no se podrá iniciar sesión.

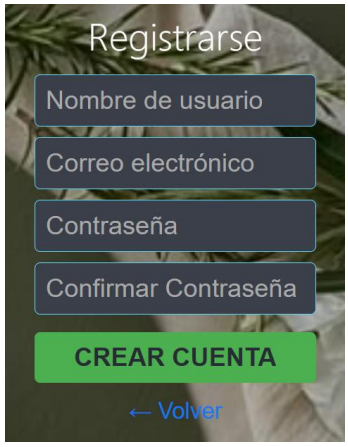


Ilustración 13 Campos de registro

En la opción de inicio de sesión hay que rellenar esos campos, si esos campos se rellenan adecuadamente, se podrá acceder a la aplicación en sí.

Ilustración 14 Campos de inicio de sesión

Además de iniciar sesión tenemos la opción de recuperar contraseña si no nos acordamos, se envía un enlace al correo donde se puede cambiar.

Ilustración 15 Opción recuperar contraseña

En esta pantalla tenemos los campos para rellenar la contraseña nueva, a esta pantalla se accede a través del enlace del correo.

Ilustración 16 Campos restablecer contraseña

Una vez iniciado sesión, tenemos nuestro perfil, donde podemos ver las distintas funcionalidades de la aplicación, como ver los favoritos, solicitudes reseñas, etc.

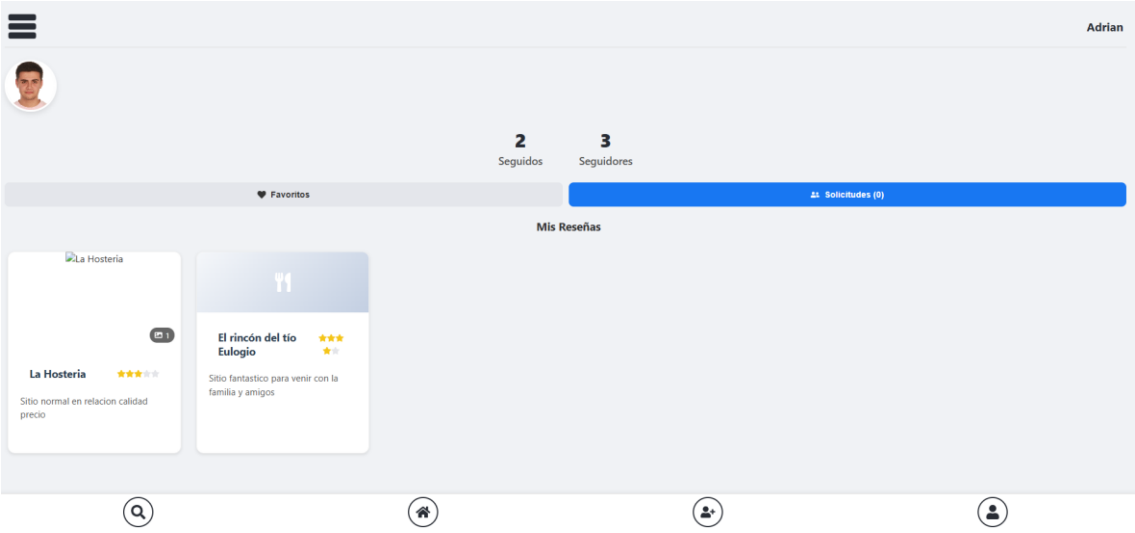


Ilustración 17 Perfil del usuario

En la siguiente imagen, tenemos las opciones del perfil, editar perfil que lo veremos a continuación, darse de baja, en la que, al pulsarlo en la base de datos, se genera la fecha de darse de baja, que al ser distinta de NULL, no podemos acceder a la aplicación. Por último, tenemos la opción de cerrar sesión que directamente sale de la aplicación.

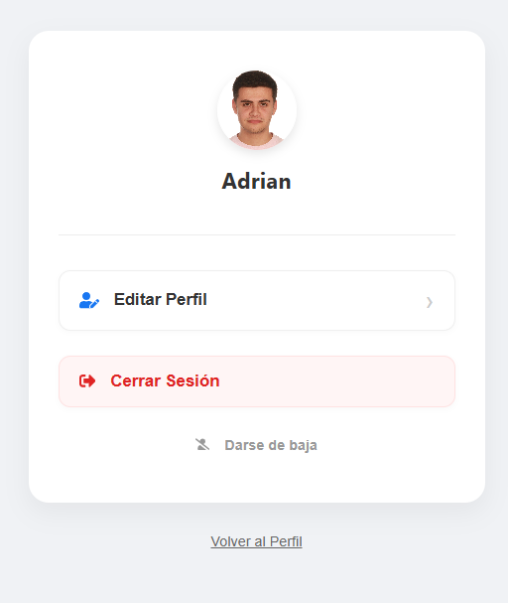


Ilustración 18 Opciones del menú

Dentro de la opción de editar perfil, tenemos las distintas funciones, la opción de editar imagen, tienes que acceder a tus archivos para editarla, después tenemos las opciones de cambiarte el nombre, editar el correo (no puede ser uno ya existente en la aplicación) y por ultimo la contraseña en la que tienes que poner la antigua y repetir la nueva por seguridad.

Ilustración 19 Editar perfil

Fuera de las opciones de editar perfil, tenemos el botón de ver mis restaurantes favoritos, como podemos ver a la derecha, sale en rojo la opción de guardar, si pulsamos de nuevo, se quita de favoritos, además de poder acceder a la ficha del restaurante desde aquí.

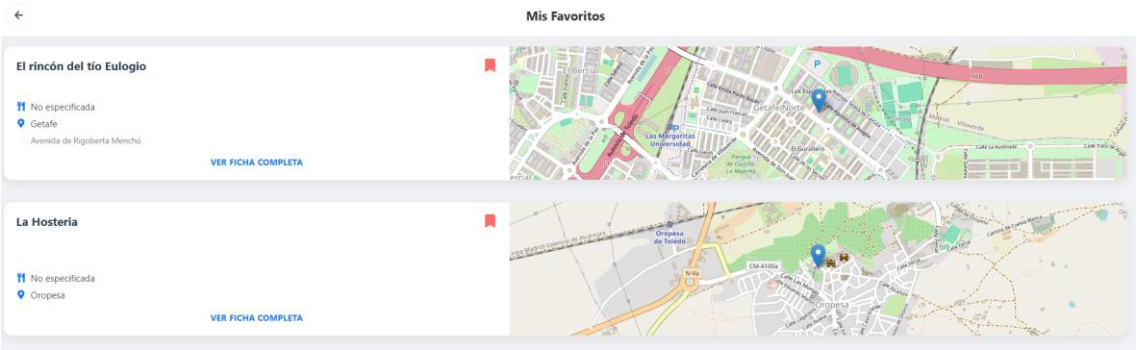


Ilustración 20 Favoritos

Otro botón que tenemos es el de ver las solicitudes de seguimiento, en este caso, no hay, pero si hubiese, tendríamos la opción de aceptarla o denegarla.

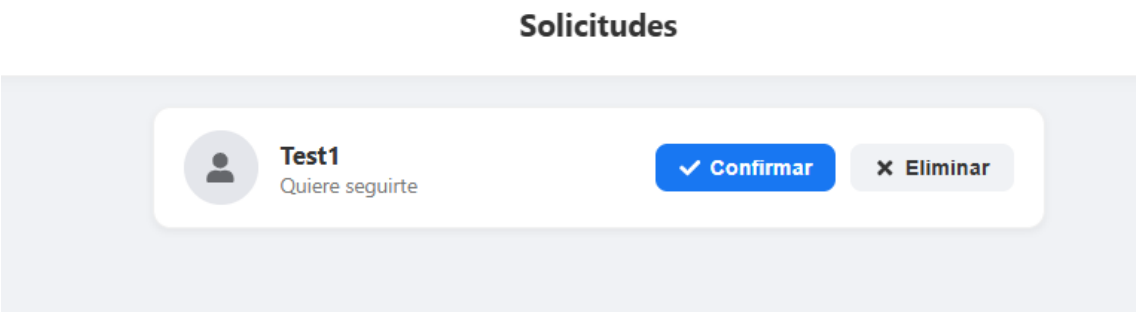


Ilustración 21 Solicitudes de seguimiento

Por último, podemos ver nuestras reseñas realizadas, donde están el número de valoración sobre 5 que le hemos dado, la descripción que ponemos y las imágenes (opcionales), además de poder borrar la reseña.

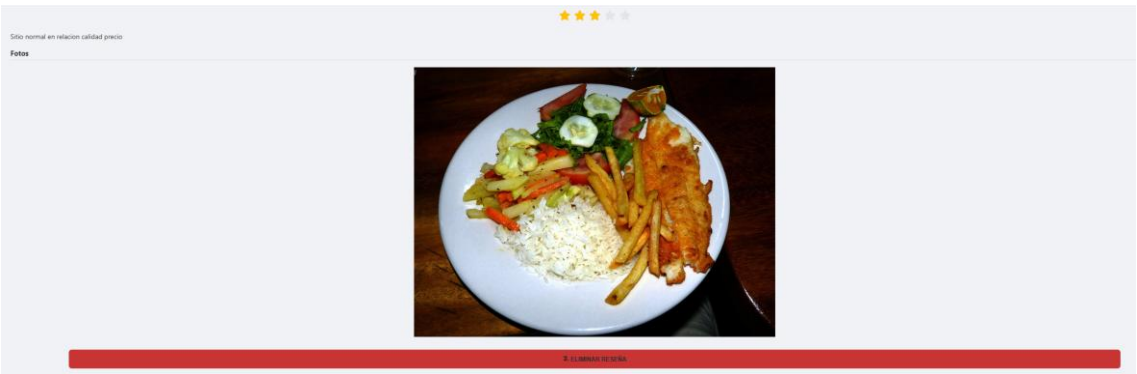


Ilustración 22 Reseña de un restaurante

En la siguiente ilustración, está la opción de buscar usuarios, segundo botón empezando por la derecha, donde ponemos el nombre o letras que queramos y salen los usuarios que coinciden.

Ilustración 23 Búsqueda de usuarios

Si pulsamos en el perfil del usuario, en este caso ya le seguimos, podemos ver su perfil completo.

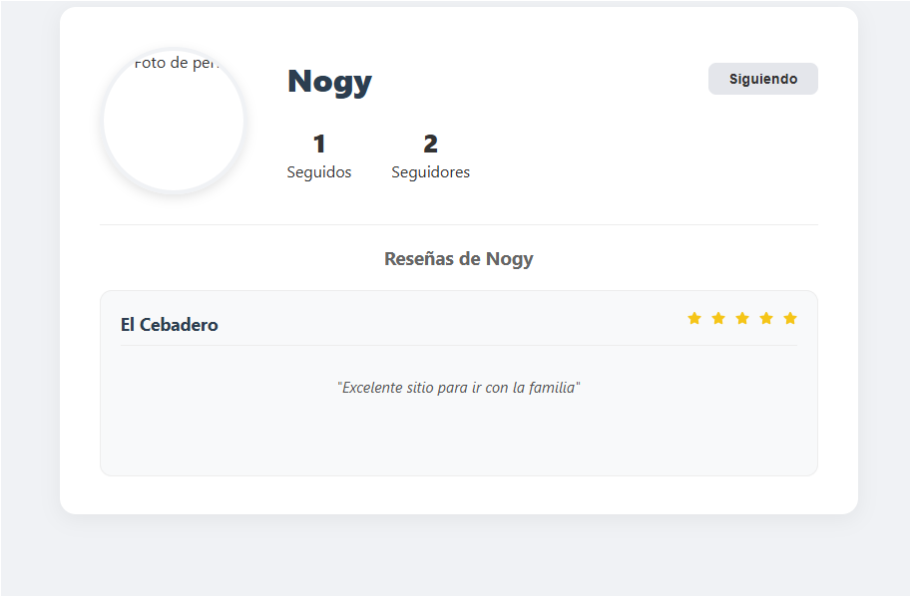


Ilustración 24 Perfil de otro usuario

Dentro de nuestro perfil podemos ver a la gente que seguimos.

Ilustración 25 Ver seguidos

Además de ver mis seguidores.

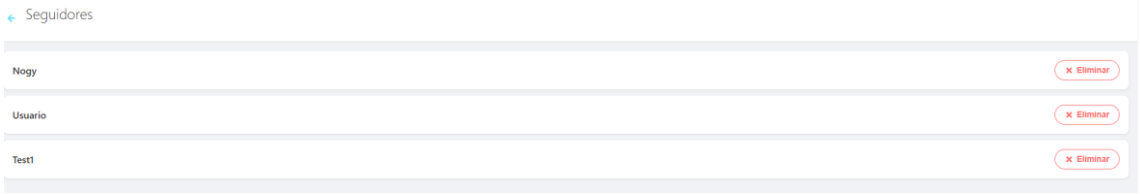


Ilustración 26 Ver seguidores

A continuación, tenemos la opción de búsqueda de restaurantes con filtros, en este caso queremos buscar los restaurantes de comida mediterránea que hay en Alcobendas.

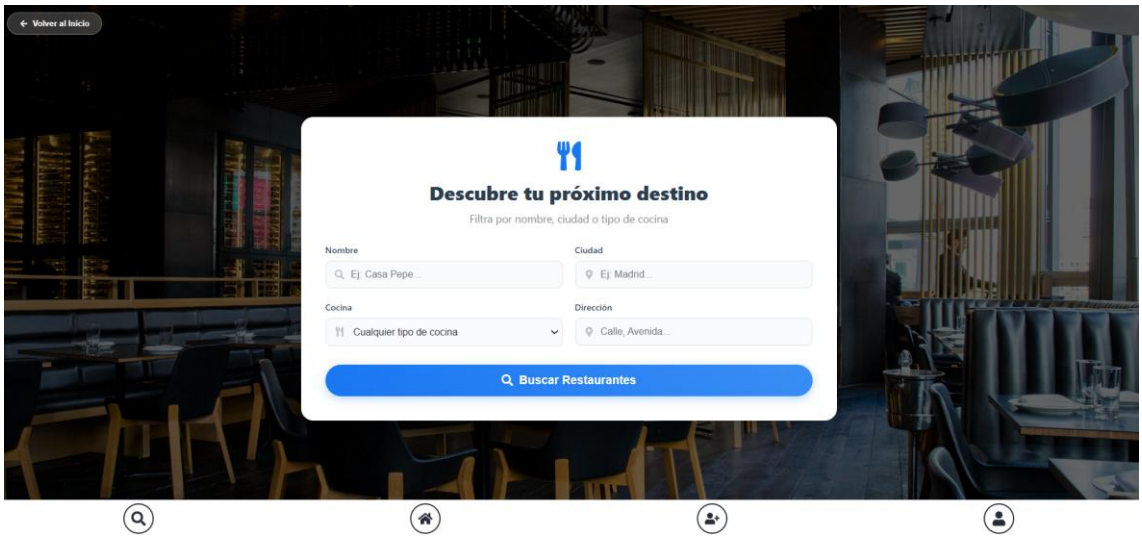


Ilustración 27 Búsqueda de restaurantes

El resultado es el siguiente, desde aquí tenemos la opción de añadirlo a favoritos y de ver el feed del restaurante.

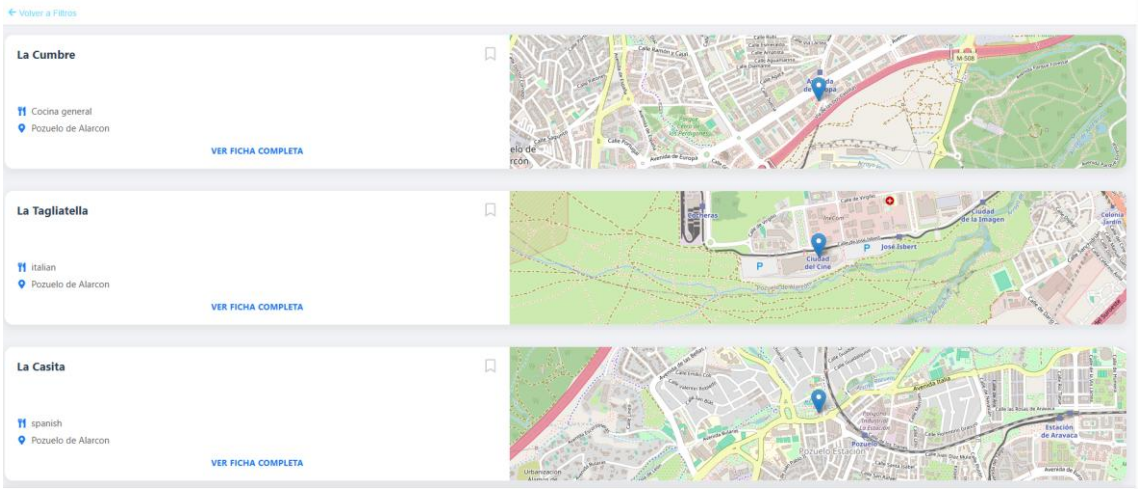


Ilustración 28 Resultado de la búsqueda

Dentro del perfil del restaurante, tenemos el tipo de cocina, la ubicación y la dirección, además de tener un mapa interactivo en el que podemos ver exactamente donde esta. Por último, tenemos la opción de hacer una reseña.

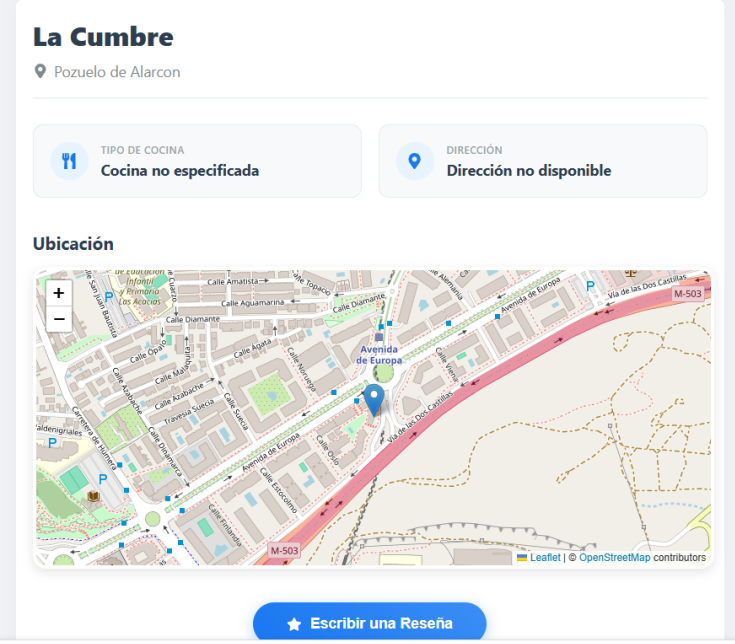


Ilustración 29 Perfil de un restaurante

En la opción de hacer la reseña, vemos el numero de estrellas que le podemos poner, el comentario sobre el restaurante y si queremos subir alguna foto.

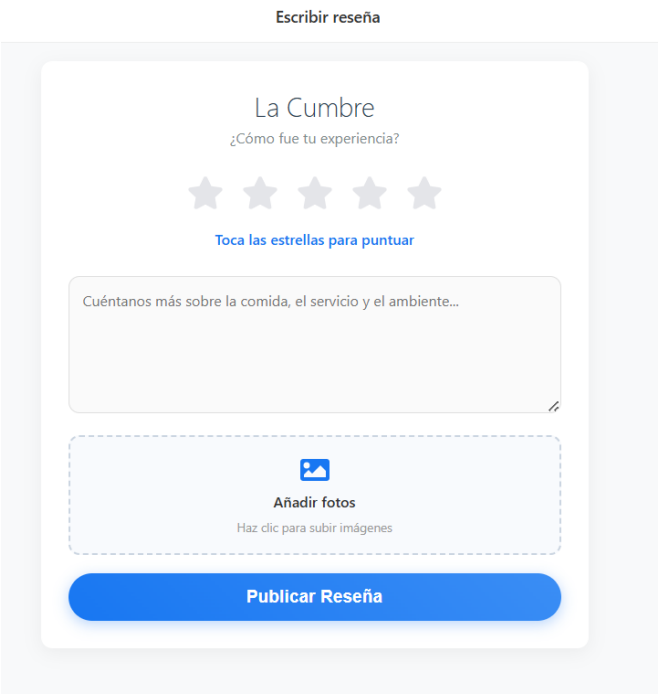


Ilustración 30 Hacer reseña

Para finalizar, tenemos el botón de la casa, en el que vemos las reseñas de otros usuarios a los que seguimos.

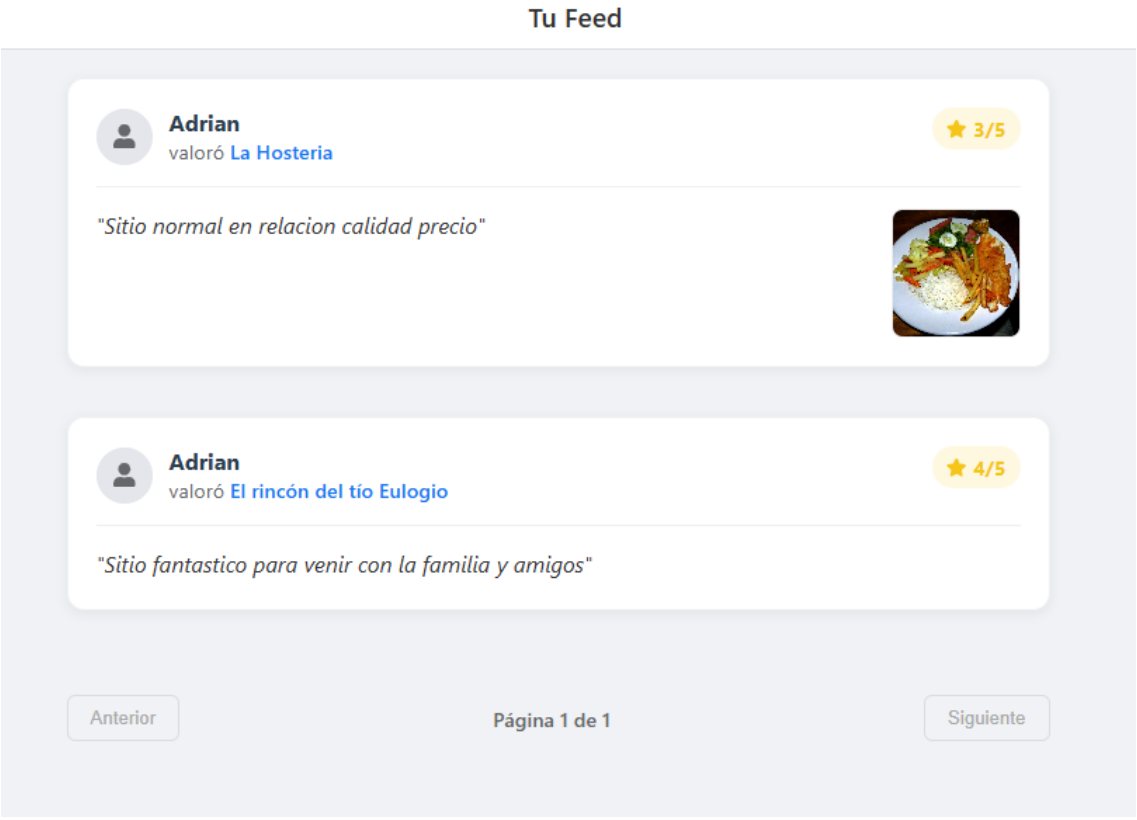
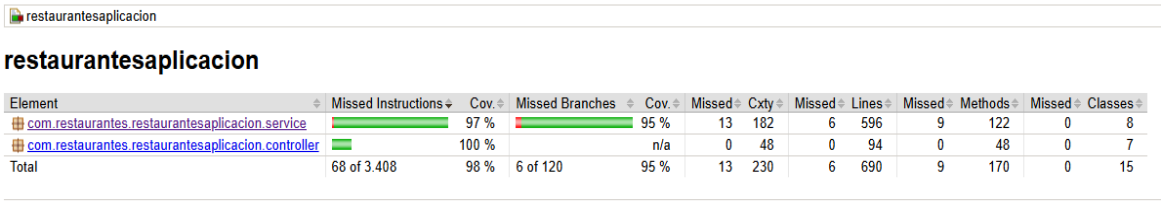


Ilustración 31 Pantalla principal

6 Validación y pruebas

Con el objetivo de garantizar la integridad y el correcto desempeño de la plataforma, se ejecutó un plan de pruebas exhaustivo sobre cada funcionalidad implementada. El desglose de estas evaluaciones se presenta a continuación en la tabla técnica. Además de esto, se usó la extensión de jacoco para ver el porcentaje de cobertura que se realiza, en la siguiente imagen se ve el porcentaje, se han realizado pruebas en la capa de servicios y controladores ya que son las únicas capas donde se puede probar la calidad del código, el archivo que muestra está en la capa de backend, target, site, cuyo nombre es "index.html". En relación con la tabla mencionada anteriormente, se muestran las pruebas mas relevantes ya que el total de pruebas realizadas es muy grande


restaurantesaplicacion

restaurantesaplicacion

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.restaurantes.restaurantesaplicacion.service		97 %		95 %	13	182	6	596	9	122	0	8
com.restaurantes.restaurantesaplicacion.controller		100 %		n/a	0	48	0	94	0	48	0	7
Total	68 of 3.408	98 %	6 of 120	95 %	13	230	6	690	9	170	0	15

Ilustración 32 Imagen de jacoco

Número	Prueba	Resultado esperado	Resultado
1	Registro Correcto	Usuario registrado y añadido a la base de datos	Ok
2	Registro ya existente	Retorna un error 409	Ok
3	Contraseña olvidada	Retorna un 200, cambio de contraseña en la base de datos	Ok
4	Contraseña olvidada email no existe	Retorna un 404 ya que el email no existe	Ok
5	Inicio de sesión	Cambio de pantalla en la aplicación	Ok
6	Inicio de sesión incorrecto	Error por contraseña o correo	Ok
7	Obtener restaurantes favoritos	Cambio de pantalla donde se encuentran los restaurantes favoritos	Ok
8	Añadir restaurante a favoritos	Mensaje de restaurante añadido	Ok

		a favoritos, añadido en la base de datos	
9	Eliminar Restaurante de favoritos	Mensaje de restaurante eliminado de favoritos, añadido en la base de datos	Ok
10	Solicitud de seguimiento	Mensaje de solicitud enviada, añadido en la base de datos como pendiente	Ok
11	Aceptar o rechazar solicitud	Se añade en la base de datos si esta aceptado o denegado	Ok
12	Ver mis seguidores o seguidos	Cambio de pantalla donde salen los usuarios	Ok
13	Ver seguidores o seguidos de otro usuario	Cambio de pantalla donde salen los usuarios	Ok
14	Ver solicitudes pendientes	Cambio de pantalla donde salen todas las solicitudes	Ok
15	Eliminar Seguidor	Modificación de la base de datos, donde ya no aparece como seguidor	Ok
16	Dejar de seguir	Modificación de la base de datos, donde ya no sigues a ese usuario	Ok
17	Buscar Restaurantes con filtros	Cambio de pantalla donde salen los restaurantes con esos filtros	Ok
18	Buscar Restaurantes no exitoso	Error 404	Ok
19	Darse de baja	Cuenta dada de baja, cambio en la base de datos (fecha distinta de null)	Ok
20	Actualizar perfil	Modificación de datos o imagen, cambio en la base de datos	Ok

21	Actualizar perfil email no valido	Error ya que el email ya esta en la base de datos	Ok
22	Buscar usuarios	Cambio de pantalla donde salen los usuarios buscados	Ok
23	Ver Perfil de otro usuario	Cambio de pantalla al perfil del usuario	Ok
24	Crear valoración	Valoración creada y añadida a la base de datos	Ok
25	Crear valoración duplicada	Retorna un 409 ya que ya existe una de ese restaurante realizada por ese usuario	Ok
26	Ver valoraciones	Cambio de pantalla a la valoración buscada	Ok
27	Eliminar valoración	Cambio en la base de datos donde la valoración ya no aparece	Ok
28	Foto guardada en la nube	url de la foto en la nube, se guarda en la base de datos	Ok
29	Foto subida incorrectamente	Retorna una excepción	Ok
30	Envío de mensaje de bienvenida	Mensaje enviado al correo	Ok
31	Envío de mensaje para verificar el email	Correo con una url para verificarlo	Ok
32	Agregar favorito cuando ya es	El restaurante no se guarda ya que ya lo esta	Ok
33	Buscar Restaurantes, ciudad errónea	Lanza una excepción de conflicto	Ok
34	Solicitud de seguimiento a ti mismo	Lanza una BadRequest	Ok
35	Ver seguidores de otro usuario no seguido	No puede ver los seguidores ya que no le sigue	Ok
36	Registrar usuario dado de baja	Lo realiza correctamente, la	Ok

		fecha de baja se pone null	
37	Iniciar sesión sin haber verificado	Lanza una excepción y no deja entrar en la web	Ok
38	Iniciar sesión dado de baja	Lanza una excepción y no deja continuar	Ok
39	Actualizar nombre ya existente	Lanza una excepción y no deja poner ese nombre	Ok
40	Actualizar email por el mismo	No ocurre nada, se actualiza igualmente	Ok
41	Actualizar imagen archivo vacío	Lanza una excepción ya que no se puede	Ok
42	Confirmar contraseña incorrecta	Lanza una excepción ya que no es la misma	Ok

Tabla 63 Pruebas Unitarias

Durante la fase de ejecución de pruebas, se identificaron diversas incidencias que fueron solventadas de manera iterativa. Tras este proceso de depuración, se ha verificado que la plataforma web alcanza un estado óptimo de estabilidad, operando satisfactoriamente bajo los parámetros previstos.

7 Conclusiones

La ejecución de este proyecto me ha permitido profundizar en el ciclo de vida completo de un producto software a escala reducida. He podido experimentar cada etapa de manera integral: desde la planificación estratégica de las tareas hasta la selección del marco metodológico, habiendo optado en este caso por un modelo en cascada para guiar el desarrollo, a parte de las herramientas que me han ayudado a realizarlo como Visual Studio Code o React, además de darle mucha importancia a la seguridad de una aplicación que es lo más importante al tratarse de datos como correos y contraseñas de usuarios.

Una vez finalizado el proyecto, concluyo que cumple su objetivo que es facilitar a las personas a conocer de primera mano el mundo gastronómico basándonos en otros usuarios que solemos conocer y no gente aleatoria. Contando con un sistema robusto de seguridad y un sistema muy intuitivo.

8 Mejoras y plan de futuro

En relación con mejoras en la aplicación encontramos:

- Dotar de roles y permisos para garantizar una mayor eficacia.
- En lo que se refiere al sistema de búsqueda de restaurantes, contar con un sistema actualizado día a día ya que surgen nuevos restaurantes y desaparecen.
- Incrementar la seguridad, concretamente en el login, es decir, si se introduce la contraseña un numero de veces bloquear la web en ese dispositivo

En la parte de plan de futuro, me gustaría implementar un chat en vivo entre usuarios y que puedan comentar reseñas de otros a los que siguen.

9 Informe de impacto en la sociedad

La aplicación va dirigida a cualquier tipo de personas ya que se centra en la gastronomía.

El objetivo principal es ayudar a conocer los restaurantes del lugar que quieras, así como una fuente de reseñas fiables, que hoy en día pueden llegar a no serlo.

10 Bibliografía

[1] «Tripadvisor,» [En línea]. Available: https://www.tripadvisor.es/ [Último acceso: 19 10 2025].

[2] «TheFork,» [En línea]. Available: https://www.thefork.es/ [Último acceso: 19 10 2025].

[3] «Spring Boot,» [En línea]. Available: https://spring.io/projects/spring-boot [Último acceso: 1 11 2025].

[4] «PostgreSQL,» [En línea]. Available: https://www.postgresql.org/ [Último acceso: 5 11 2025].

[5] «React,» [En línea]. Available: https://es.react.dev/ [Último acceso: 30 10 2025].

[6] «OpenStreetMap,» [En línea]. Available: https://www.openstreetmap.org/#map=6/40.01/-2.49 [Ultimo acceso: 25 11 2025]

[7] «Leaflet,» [En línea]. Available: https://leafletjs.com/ [Ultimo Acceso: 26 11 2025]

[8] «Cloudinary,» [En línea]. Available: https://cloudinary.com/ [Ultimo acceso: 15 11 2025]

[9] «Docker,» [En línea]. Available: https://www.docker.com/ [Ultimo acceso: 20 10 2025]

[10] «Jacoco,» [En línea]. Available: https://www.jacoco.org/jacoco/ [Ultimo acceso: 12 12 2025]

[11] «JWT,» [En línea]. Available: https://www.jwt.io/ [Ultimo acceso: 12 11 2025]

[12] «Swagger,» [En línea]. Available: https://swagger.io/ [Ultimo acceso: 2 12 2025]

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Tue Jan 20 23:07:10 CET 2026
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)