



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una aplicación web para la gestión de recetas de cocina

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Moreira Barrionuevo, Roberth Paul

Tutor: Martí Campoy, Antonio

Segundo tutor: Rodríguez Ballester, Francisco

Curso 2020-2021

Resum

La següent memòria detalla el desenvolupament d'una aplicació web, l'objectiu de la qual és proveir un lloc on compartir i gestionar receptes de cuina i discutir sobre temes culinaris.

L'aplicació es troba estructurada en dues parts, el desenvolupament del back-end emprant el framework de desenvolupament Spring, amb el qual es construirà un API REST de consultes HTTP. D'altra banda el desenvolupament del front-end, la interfície d'usuari, que es construirà amb el framework de desenvolupament web Angular emprant el llenguatge de programació web TypeScript.

Al llarg de la memòria es detallaran les fases de desenvolupament emprades per a construir l'aplicació web, l'especificació de requisits, l'anàlisi, la implementació i proves, a més d'altres aspectes relacionats amb la resolució de problemes i com se ha arribat a la solució proposada.

Paraules clau: Framework, Spring, Angular, back-end, front-end

Resumen

La siguiente memoria detalla el desarrollo de una aplicación web, cuyo objetivo es proveer un sitio donde compartir y gestionar recetas de cocina y discutir acerca de temas culinarios.

La aplicación se encuentra estructurada en dos partes, el desarrollo del *back-end* empleando el *framework* de desarrollo *Spring*, con el cual se construirá un API REST de consultas HTTP. Por otro lado el desarrollo del *front-end*, la interfaz de usuario, que se construirá con el *framework* de desarrollo web Angular empleando el lenguaje de programación web TypeScript.

A lo largo de la memoria se van a detallar las fases de desarrollo empleadas para construir la aplicación web, la especificación de requisitos, el análisis, la implementación y pruebas, además de otros aspectos relacionados con la resolución de problemas y como se ha llegado a la solución propuesta.

Palabras clave: Framework, Spring, Angular, back-end, front-end

Abstract

The following report details the development of a web application, whose objective is to provide a site to share and manage cooking recipes and discuss about culinary topics. The application is structured in two parts, the back-end development using the Spring development framework, with which a REST API for HTTP queries will be built.

On the other hand, the front-end development, the user interface, which will be built with the Angular web development framework using the TypeScript web programming language.

Throughout the report we will detail the development phases used to build the web application, the requirements specification, analysis, implementation and testing, as well as other aspects related to problem solving and how the proposed solution has been reached.

Key words: Framework, Spring, Angular, back-end, front-end

Índice general

Índice general	V
Índice de figuras	VII
Índice de tablas	VIII
<hr/>	
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Plan de trabajo	2
1.4 Estructura de la memoria	2
2 Metodología	5
3 Especificación de requisitos	7
3.1 Introducción	7
3.1.1 Propósito	7
3.1.2 Ámbito del sistema	7
3.1.3 Definiciones, Acrónimos y abreviaturas	8
3.2 Descripción general	8
3.2.1 Perspectiva del producto	8
3.2.2 Funciones del producto	8
3.2.3 Características de los Usuarios	8
3.2.4 Restricciones	9
3.2.5 Suposiciones y Dependencias	9
3.3 Requisitos Específicos	10
3.3.1 Interfaces Externas	10
3.4 Funciones	10
3.4.1 Definición de roles	11
3.4.2 Definición de funciones	11
3.5 Requisitos de rendimiento	14
3.6 Restricciones de diseño	14
3.7 Atributos de sistema	14
4 Análisis	15
4.1 Diseño estructural	15
4.1.1 Diagrama de contexto	15
4.1.2 Diagrama de clases	16
4.2 Diagramas de casos de uso	18
4.2.1 Modelado por actores	33
5 Diseño	35
5.1 Diseño Arquitectónico: modelo, vista, controlador	35
5.2 Diseño de la interfaz gráfica de la aplicación web	36
5.2.1 Representación de los prototipos	36
6 Implementación	47
6.1 Back-End	47
6.1.1 Tecnologías	47

6.1.2	Elección de las tecnologías	47
6.1.3	Implementación del Back-End	48
6.2	Front-End	51
6.2.1	Tecnologías	51
6.2.2	Elección de Angular	51
6.2.3	Implementación del <i>front-end</i>	52
6.3	Herramientas	55
7	Pruebas	57
7.1	Registro	57
7.2	Inicio de sesión	57
7.2.1	Fallo en el inicio de sesión	58
7.3	Cerrar sesión	59
7.4	Alta de una receta	60
7.5	Editar una receta	62
7.6	Borrar una receta	62
7.7	Visualizar una receta	63
7.8	Comentar una receta	64
7.9	Compartir una receta	65
7.10	Alta de un recetario	66
7.11	Borrar un recetario	67
7.12	Añadir una receta al recetario	67
7.13	Búsqueda de una receta	68
7.14	Alta de una tema en el foro	68
7.15	Responder un tema en el foro	70
7.16	Cambiar contraseña	71
7.17	Dar de baja una cuenta de usuario	72
7.18	Pruebas para el usuario anónimo	72
7.18.1	Página principal	72
7.18.2	Visualización de la receta y comentarios	72
7.18.3	Foro y respuestas	74
8	Presupuesto	75
9	Conclusiones	77
9.1	Trabajo futuro	77
9.2	Valoración	78
	Bibliografía	79

Índice de figuras

2.1	Metodología	5
4.1	Diagrama de contexto	16
4.2	Diagrama de clases	17
4.3	Diagrama de casos de uso para el rol Usuario	33
4.4	Diagrama de casos de uso para el rol Anónimo	34
5.1	Patrón arquitectónico MVC	35
5.2	Logotipo de JustInMind	36
5.3	Alta de un usuario	36
5.4	Alta de un usuario	37
5.5	Cerrar sesión	37
5.6	Búsqueda receta	38
5.7	Baja de una cuenta de usuario	38
5.8	Dar de alta una receta	39
5.9	Editar una receta	39
5.10	Borrar una receta	40
5.11	Comentar una receta	41
5.12	Representación de una receta	42
5.13	Compartir una receta vía enlace	43
5.14	Crear un recetario	43
5.15	Guardar una receta en el recetario	44
5.16	Dar de alta un tema en el foro	44
5.17	Cerrar un tema en el foro	45
5.18	Comentar en el foro	45
5.19	Cambiar contraseña	46
6.1	Descripción del diseño del servidor	48
6.2	Estructura general del <i>back-end</i>	49
6.3	Creación de un método personalizado en JPA.	50
6.4	Configuración de seguridad	51
6.5	Configuración de rutas accesibles por cualquier cliente	51
6.6	Componentes Bootstrap importados al proyecto web	53
6.7	Estructura de la página web	53
6.8	Estructura de los componentes	54
6.9	Modelos definidos en Angular	54
6.10	Carpeta raíz de la aplicación	55
7.1	Registro	57
7.2	Inicio de sesión	58
7.3	Fallo en el inicio de sesión	58
7.4	Perfil de usuario	59
7.5	Cerrar sesión	59
7.6	Alta de una receta	60

7.7	Receta añadida a la página de gestión	61
7.8	Editar receta	62
7.9	Borrar receta	62
7.10	Visualizar receta	63
7.11	Comentar receta	64
7.12	Comentario en una receta	64
7.13	Botón para compartir una receta.	65
7.14	Enlace generado para compartir la receta	65
7.15	Alta recetario	66
7.16	Recetario añadido	66
7.17	Eliminar recetario	67
7.18	Añadir una receta al recetario	67
7.19	Receta añadida añadida al recetario	67
7.20	Búsqueda	68
7.21	Filtro	68
7.22	Foro	68
7.23	Añadir tema al foro	69
7.24	Tema añadido al foro	69
7.25	Tema de discusión	70
7.26	Formulario de respuesta a un tema	70
7.27	Respuesta subida al foro	70
7.28	Botón para cambiar la contraseña	71
7.29	Error en el cambio de la contraseña	71
7.30	Baja de la aplicación	72
7.31	Página principal de la aplicación	73
7.32	Cebecera de a receta visualizada por un usuario anónimo	73
7.33	Comentarios restringidos al usuario anónimo	73
7.34	Visualización del foro siendo un usuario anónimo	74
7.35	Visualización del los temas siendo un usuario anónimo	74

Índice de tablas

1.1	Tabla de distribución de trabajo	2
3.1	Inicio de sesión	11
3.2	Registro de un usuario	11
3.3	Búsqueda de una receta	12
3.4	Dar de alta una receta	12
3.5	Tabla 05: Editar una receta	12
3.6	Comentar una receta	12
3.7	Compartir una receta	12
3.8	Dar de baja una cuenta	12
3.9	Crear un recetario	12
3.10	Añadir una recetar al recetario	13
3.11	Borrar un recetario	13
3.12	Dar de alta un tema de discusión en el foro	13

3.13	Comentar en el foro.	13
3.14	Cerrar un tema de discusión.	13
3.15	Cambio de contraseña.	13
4.1	Caso de uso 1: Registro	19
4.2	Caso de uso 2: Iniciar de sesión	20
4.3	Caso de uso 3 : Cerrar sesión	20
4.4	Caso de uso 04: Búsqueda de una receta	21
4.5	Caso de uso 5: Dar de baja una cuenta de usuario	22
4.6	Caso de uso 6: Dar de alta una receta	23
4.7	Caso de uso 7: Editar una receta	24
4.8	Caso de uso 8: Borrar una receta	25
4.9	Caso de uso 9: Comentar una receta	26
4.10	Caso de uso 10: Compartir una receta	26
4.11	Caso de uso 11: Crear un recetario	27
4.12	Caso de uso 12: Guardar una receta en el recetario	28
4.13	Caso de uso 13: Dar de alta un tema en el foro	29
4.14	Caso de uso 14: Cerrar un tema en el foro	30
4.15	Caso de uso 15: Responder un tema del el foro	31
4.16	Caso de uso 16: Comentar en el foro	32
8.1	Tabla de costes	76

CAPÍTULO 1

Introducción

Mi proyecto tiene como objetivo diseñar y desarrollar una aplicación web que gestione recetas de cocina que los usuarios registren en la aplicación web, donde se podrá clasificar la receta según diferentes etiquetas como “baja en calorías”, “baja en azúcar” o “sin gluten”. Cualquier persona relacionada con la cocina podrá compartir sus propias recetas, desde un cocinero experto a una persona que simplemente quiere compartir una receta familiar.

Para poder construir esta aplicación web haré uso del *framework Spring*¹ para implementar el *back-end* de la aplicación, que se encargará de la gestión de las funciones “núcleo” y por otro lado, el *front-end* de la aplicación será construido en *Angular*², un *framework* de desarrollo web que ofrece la capacidad de crear una *single page application*, así los cambios son instantáneos y hace que la web sea más rápida.

A lo largo de los siguientes puntos se detalla el desarrollo de la aplicación web, desde la especificación de requisitos, pasando por la elaboración de casos de uso hasta la implementación, validación y finalizando con despliegue.

1.1 Motivación

Normalmente se pueden encontrar muchas páginas web donde consultar recetas de cocina desde recetas de chinas pasando la tailandesa y vietnamita, hasta la cocina peruana y tradicional occidental, pero estas recetas son subidas y editadas por los propios editores de las páginas web, nunca por el propio cocinero.

Seguramente existan cientos y cientos de recetas de cocina desconocidas por muchos que son propias de una familia, región o país que aun no han tenido la oportunidad de ser compartidas y cocinadas por los usuarios más curiosos, por ello del desarrollo de esta aplicación web, una herramienta con la que se podrá compartir recetas que los diferentes usuarios suban a la plataforma.

1.2 Objetivos

Principalmente la aplicación web tendrá como objetivo la gestión y publicación de recetas, con la oportunidad de poder ser puntuadas, guardadas y filtradas por distintos criterios sujetos a la propia receta.

Otros objetivos a tener en cuenta son:

¹<https://spring.io/>

²<https://angular.io/>

1. Un sistema de autenticación y registro de usuarios.
2. Portal de gestión de recetas se puede seleccionar una receta para editar o borrar.
3. Un sistema de gestión de recetas, donde los usuario añaden recetas a una colección, que puede tener varios objetivos.
4. Un foro de discusión en el cual se pueden discutir diferentes tipos de temas, relacionados con las recetas o con la comida en general.

1.3 Plan de trabajo

En esta sección se detalla el plan de trabajo previsto para este trabajo de fin de grado, basada en una jornada de seis horas a lo largo de 11 semanas, en total 330 horas dedicadas a la elaboración del proyecto.

Tarea	Duración	Semanas										
		1	2	3	4	5	6	7	8	9	10	11
Requisitos	13 días	x	x	x								
Análisis	3 días			x	x							
Diseño	4 días				x							
Implementación	20 días					x	x	x	x			
Pruebas	10 días									x	x	
Memoria	5 días											x

Tabla 1.1: Tabla de distribución de trabajo

1.4 Estructura de la memoria

Este documento se estructura de la siguiente manera:

- **Capítulo 1:** en este capítulo se introduce el trabajo realizado, dando un contexto, la motivación, objetivos y plan de trabajo.
- **Capítulo 2:** se detalla la metodología empleada para el desarrollo de la aplicación.
- **Capítulo 3:** se especifican los requisitos de la aplicación en base al estándar IEEE 830.
- **Capítulo 4:** se realiza un análisis de la especificación de requisitos especificando los casos de uso de la aplicación.
- **Capítulo 5:** apartado de diseño de la aplicación, donde se detallan los prototipos y el modelo arquitectónico.
- **Capítulo 6:** se indican cuales son las tecnologías empleadas para el desarrollo de la aplicación, haciendo hincapié en aspectos mas específicos como la estructura de la aplicación y la seguridad.
- **Capítulo 7:** sección donde se realizan una serie de pruebas para verificar las funciones de la aplicación web.

- **Capítulo 8:** descripción de un presupuesto para el desarrollo de la aplicación.
- **Capítulo 9:** conclusiones, futuras mejoras y valoración final.

CAPÍTULO 2

Metodología

El tipo de metodología del que se hace uso para el desarrollo del proyecto web es el modelo cascada, un modelo lineal de diseño secuencial. Es un enfoque metodológico que ordena rigurosamente las etapas del proceso de desarrollo software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

Al final de cada etapa esta pasa por un proceso de revisión que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase, la figura 2.1 representa esta metodología.

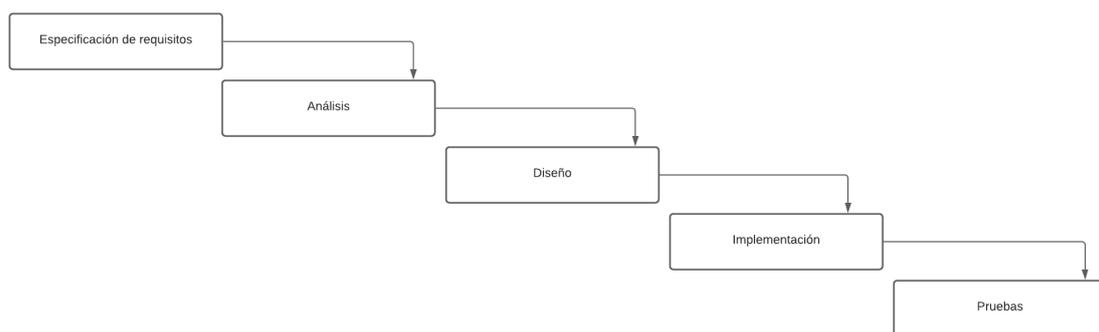


Figura 2.1: Metodología

Se detallan a continuación las fases de desarrollo en las cuales se divide el proyecto web:

- **Especificación de requisitos:** fase del proceso de desarrollo software donde se obtiene la descripción completa del comportamiento del software a desarrollar. Mediante diferentes reuniones con los clientes, se plantean los requerimientos que debe cumplir el sitio web.
- **Análisis:** proceso en el cual se analizan las ideas extraídas de la especificación de requisitos para definir los modelos, las funcionalidades y estructuras del sistema que se quiere construir.
- **Diseño:** fase en que consiste en el diseño de los componentes del sistema que dan respuesta a las funcionalidades descritas en la etapa de análisis.
- **Implementación:** parte en la cual se programan las funcionalidades y requerimientos descritos en las etapas anteriores empleando las tecnologías y herramientas escogidas.

- **Pruebas:** fase de *testing* en la cual se pone a prueba la implementación de las funcionalidades desarrolladas para validar y asegurar su correcto funcionamiento.

CAPÍTULO 3

Especificación de requisitos

La fase de especificación de requisitos consiste en definir de forma clara y concisa los requerimientos funcionales y no funcionales de los cuales se compone la aplicación web. Esta fase es de vital importancia, ya que definir los requerimientos muy precisamente ayudará a que no se tenga que realizar cambios en las fases más avanzadas del proyecto, retrasando de manera significativa el desarrollo del proyecto.

3.1 Introducción

A lo largo de este capítulo se detalla la especificación de requisitos del proyecto web según el documento de "Especificación de requisitos según el estándar de IEEE 830"[1], donde se definen en varias secciones la descripción general del proyecto y los requisitos más específicos, tanto funcionales como de rendimiento.

3.1.1. Propósito

La especificación de requisitos tiene como propósito la correcta definición de los requerimientos software de los que se compone la aplicación. Esto es clave para recoger de manera clara cuales son los elementos de los que se compone el proyecto software, atendiendo fundamentalmente a las necesidades del cliente o los usuarios.

Una de las premisas fundamentales dentro de la especificación de requisitos es evitar retrasos o errores en las fases más avanzadas del desarrollo, ya que la no definición clara de los requisitos puede llevar a contratiempos que pueden afectar a la calidad del **software** o a retrasos en los tiempos de entrega.

La presente definición de requisitos va dirigida al usuario lector de esta memoria, para que tenga una idea clara de la estructura de la aplicación web que se va a desarrollar.

3.1.2. Ámbito del sistema

El proyecto web tiene como finalidad ser un sitio web donde compartir fácilmente recetas de cocina y discutir de muchos aspectos relacionados con la comida, tomando el nombre de "FoodCompanion". Este sitio web tiene como objetivo ser un lugar donde poder compartir recetas, de manera fácil, rápida y sencilla. Además de un sitio donde discutir diferentes temas en los foros y comentar las recetas subidas por los usuarios.

Des esta misma forma, el sitio web no permitirá el uso de los foros o del sistema de comentarios en como lugar donde discutir de temas no relacionados con las receta o la

cocina en general. No es bienvenido cualquier tema fuera del ámbito culinario o cualquier abuso por parte de los usuarios.

3.1.3. Definiciones, Acrónimos y abreviaturas

- **HTML:** *HyperText Markup Language*.
- **CSS:** *Cascading Style Sheets* o “Hojas de estilo en cascada”.
- **MySQL:** Sistema de gestión de bases de datos relacional.
- **Hosting:** espacio donde se almacenan ficheros accesibles mediante la web.
- **IEE:** *Institute of Electrical and Electronics Engineers* o “Instituto de Ingenieros Eléctricos y Electrónicos”.
- **ERS:** especificación de requerimientos de software..

3.2 Descripción general

En esta sección se describen todos aquellos factores que afectan al producto, pero no se describen sus requisitos si no su contexto.

3.2.1. Perspectiva del producto

De forma general la aplicación web que se pretende desarrollar tiene como base la implementación de un sistema que permita el alojamiento de recetas de cocina, así mismo de su consulta para poder ser elaboradas, un sistema de comentarios sencillo y un foro de opinión para discutir de todo aspecto relacionado a al ámbito culinario.

Al ser de una aplicación web, se asume que el usuario tendrá nociones básicas de navegación web y edición de textos, para poder colgar sus recetas en el sistema y poder navegar por la aplicación. Se sabe que no todos los usuarios van a tener el mismo nivel de desarrollo tecnológico, por ello se tiene como premisa el desarrollo de una interfaz lo mas sencilla y clara posible, para facilitar la tarea a los usuarios menos expertos.

3.2.2. Funciones del producto

A grandes rasgos las funciones principales de la aplicación web a implementar son en el almacenamiento de diferentes recetas de cocina, su administración para poder ser visualizadas, editadas o borradas, un sistema de comentarios en cada receta, la definición de colecciones de recetas o recetarios y un foro de discusión.

Por otro lado, también debe poseer funciones mas específicas como el alta o baja de una cuenta de usuario, un buscador de recetas y una página principal para visualizar recetas sugeridas por el sistema.

3.2.3. Características de los Usuarios

La aplicación tiene como usuario objetivo un usuario capaz de seguir una serie de instrucciones para poder cocinar las recetas que se publicarán en el sitio web, por ello se tiene en cuenta a un usuario con un nivel educativo medio, capaz de leer y manejar diferentes tipos de herramientas culinarias.

En cambio si se pasa al nivel puramente técnico basado en el uso de nuevas tecnologías, como la web, se pueden encontrar todo tipo de usuarios, pero en general se tiene como base a un usuario capaz de entender el uso de un navegador, acceder a la web y dar de alta una cuenta de usuario o directamente buscar una receta y cocinarla, por ello se piensa en un usuario con nivel tecnológico medio, capaz de usar un teléfono móvil o un ordenador para acceder a páginas web e interactuar con sus contenidos.

3.2.4. Restricciones

La aplicación web "FoodCompanion" tiene como objetivo ser desarrollada en *Spring* para la sección del *back-end*, encargada del procesamiento de recetas y los usuarios; y por otro lado *Angular* para la sección de *front-end*, la interfaz de usuario.

Esencialmente, ambas herramientas van a aportar lo necesario para crear el esqueleto de la aplicación y gracias a otros *frameworks* de desarrollo como *Bootstrap*¹ para el diseño web, *MySQL*² para el manejo de los datos y por último la herramienta *PostMan*³ para el *testing* de la implementación del *Back-end* aportan lo necesario para completar el desarrollo de la aplicación.

En cuanto a los lenguajes de programación necesarios para la implantación de lo mencionado anteriormente, se emplea *Java* para la programación de las clases y relaciones necesarias en *Spring*⁴, *TypeScript*, *HTML* y *CSS* para el desarrollo web del *front-end* y por último *MySQL* para la creación y administración de las tablas en la base de datos.

Pasando a hablar de las limitaciones *hardware* que la aplicación posee, se tienen en cuenta las limitaciones impuestas por el *hosting* de la página web, en este caso, el proyecto web al no tener fase de despliegue, únicamente se tienen en cuenta las limitaciones impuestas por la máquina donde se va a desarrollar el proyecto.

Por último teniendo en cuenta al usuario definido en el apartado 3.2.3, se decide tener en cuenta unos requisitos de habilidad básicos: el saber acceder a la web, navegar y poseer nociones básicas en cuanto al manejo y gestión de cuentas son los requisitos básicos para la utilización correcta de la aplicación web.

3.2.5. Suposiciones y Dependencias

Al ser un proyecto orientado al desarrollo de una aplicación web para un trabajo de fin de grado, este trabajo está dispuesto para ser desarrollado por una persona a lo largo de la programación establecida en el calendario de trabajo, descrito en el apartado 1.3, donde algún desvío puede presuponer el retraso de la entrega del trabajo.

Pasando a los sistemas *software* necesarios para el correcto funcionamiento de la aplicación, se tiene en consideración un *hosting* que provea con lo necesario para desplegar la aplicación, por ello se presupone la instalación de *JavaScript* y *NodeJs* para el *front-end* además de *Java*⁵ y *MySQL*⁶ para el *back-end* de la aplicación.

Gracias a ser tecnologías compatibles en muchos sistemas operativos, para el desarrollo se puede emplear cualquier sistema operativo dentro del ámbito de Linux, Windows o Mac.

¹<https://getbootstrap.com/>

²<https://www.mysql.com/>

³<https://www.postman.com/>

⁴<https://spring.io/>

⁵<https://www.java.com/es/>

⁶<https://bigdata-analytics.es/sql/>

3.3 Requisitos Específicos

A lo largo de esta sección se detallan los requisitos específicos de la aplicación web. Estos requisitos están descritos a un nivel de detalle lo suficientemente claro para permitir al desarrollador diseñar el sistema software, realizar pruebas para verificar el sistema implementado y así poder satisfacer las necesidades del cliente.

En esta sección se describen los comportamientos externos al sistema, perceptibles por parte de los usuarios y otros sistemas, poniendo cierto énfasis en las siguientes características :

- **Corrección:** El requisito debe reflejar una necesidad real a implementar en el sistema.
- **No ambiguo:** Cada requisito tiene una sola interpretación.
- **Completo:** El requisito satisface de manera completa las necesidades del cliente.
- **Consistentes:** Los requisitos no pueden ser contradictorios.
- **Clasificados:** Cada requisito es identificable.
- **Verificable:** Es verificable si existe un proceso finito y no costoso capaz de demostrar que el sistema cumple con el requisito.
- **Modificable:** Si se pueden realizar cambios de forma fácil, completa y consistente, sin afectar al sistema.
- **Trazable:** Es trazable si se conoce el origen y se facilita la referencia de cada requisito.

3.3.1. Interfaces Externas

A continuación se definen los requisitos que afectan a la interfaz del usuario o interfaz con otros sistemas.

- Una barra de búsqueda lo suficientemente grande e intuitiva para el usuario, donde se especifica un criterio de búsqueda, además de sugerir el ítem que mas se adecue a la búsqueda.
- Un botón de acceso directo al inicio.
- Un botón de acceso directo a la página de administración del usuario.
- Un botón de acceso directo al foros.
- Una vez autenticado en el sistema, un botón de acceso a la pantalla de gestión de recetas y perfil de usuario, además de una botón para cerrar sesión.

3.4 Funciones

En esta sección se definen las funciones del sistema, las cuales según el estándar IEEE 803 pueden estar organizadas de diferentes formas. En este documento se ha decidido dividir las funciones según el tipo de usuario. Cada usuario posee un conjunto de diferentes requisitos relacionados con cada tarea o función a los que están sujetos.

3.4.1. Definición de roles

A continuación se definen los roles a tener en cuenta en el sistema:

1. **Usuario:** tiene la capacidad de subir recetas a la plataforma web, así mismo de poder editarlas, borrarlas, comentar otras recetas, crear colecciones de recetas y participar en los temas añadidos a los foro.
2. **Anónimo:** tiene restringido el uso del sistema de comentarios y el foro, puede crear una cuenta de “usuario”, iniciar sesión, visualizar las recetas y el foro.

3.4.2. Definición de funciones

Como se ha mencionado en el apartado anterior, en el siguiente punto se definen las funciones adheridas a cada usuario. Para la correcta estructuración de este apartado se emplearán una serie de tablas donde se especifica:

- El nombre de la función.
- El rol
- Su grado de importancia.
- La definición de la función.

A continuación se definen las funciones principales del sistema:

Nombre de la función:	Inicio de sesión
Rol:	Usuario
Importancia:	Alta
Definición de la función:	El sistema permitirá al usuario identificarse con un nombre de usuario y una contraseña, para poder acceder a su página principal, siendo esta el panel de administración.

Tabla 3.1: Inicio de sesión

Nombre de la función:	Registro de un usuario
Rol:	Anónimo
Importancia:	Alta
Definición de la función:	El usuario será capaz de dar de alta a un usuario en la aplicación a través de un formulario de alta.

Tabla 3.2: Registro de un usuario

Nombre de la función:	Búsqueda de una receta
Rol:	Usuario, anónimo
Importancia:	Alta
Definición de la función:	El usuario será capaz de buscar una receta según una palabra o serie de palabras.

Tabla 3.3: Búsqueda de una receta

Nombre de la función:	Dar de alta una receta
Rol:	Usuario
Importancia:	Alta
Definición de la función:	El sistema permitirá al usuario dar de alta una receta.

Tabla 3.4: Dar de alta una receta

Nombre de la función:	Editar una receta
Rol:	Usuario
Importancia:	Alta
Definición de la función:	El usuario será capaz de poder editar un receta, solo el usuario que añadió la receta la puede modificar.

Tabla 3.5: Tabla 05: Editar una receta

Nombre de la función:	Comentar una receta
Rol:	Usuario
Importancia:	Media
Definición de la función:	El usuario será capaz de publicar comentarios en las recetas.

Tabla 3.6: Comentar una receta

Nombre de la función:	Compartir una receta
Rol:	Usuario, anónimo
Importancia:	Media
Definición de la función:	El usuario podrá compartir una receta mediante un enlace.

Tabla 3.7: Compartir una receta

Nombre de la función:	Dar de baja un cuenta
Rol:	Usuario
Importancia:	Alta
Definición de la función:	El sistema permitirá al usuario darse de la aplicación.

Tabla 3.8: Dar de baja una cuenta

Nombre de la función:	Crear un recetario
Rol:	Usuario
Importancia:	Media
Definición de la función:	El usuario será capaz de crear un recetario para guardar recetas.

Tabla 3.9: Crear un recetario

Nombre de la función:	Añadir una receta al recetario
Rol:	Usuario
Importancia:	Media
Definición de la función:	El usuario será capaz de añadir una receta un recetario dado, creado por el usuario.

Tabla 3.10: Añadir una receta al recetario

Nombre de la función:	Borrar un recetario
Rol:	Usuario
Importancia:	Media
Definición de la función:	El usuario será capaz de borrar una recetario seleccionado, creado por el usuario.

Tabla 3.11: Borrar un recetario

Nombre de la función:	Dar de alta un tema de discusión en el foro
Rol:	Usuario
Importancia:	Media
Definición de la función:	El sistema permitirá al usuario dar de alta un tema de discusión en el foro.

Tabla 3.12: Dar de alta un tema de discusión en el foro

Nombre de la función:	Comentar en el foro
Rol:	Usuario
Importancia:	Media
Definición de la función:	Los usuarios podrán comentar los tema de discusión añadidos al foro

Tabla 3.13: Comentar en el foro.

Nombre de la función:	Cerrar un tema de discusión
Rol:	Usuario
Importancia:	Media
Definición de la función:	Los usuarios podrán cerrar los temas añadidos por ellos al foro de discusión.

Tabla 3.14: Cerrar un tema de discusión.

Nombre de la función:	Cambio de contraseña
Rol:	Usuario
Importancia:	Alta
Definición de la función:	El usuario será capaz de cambiar su contraseña.

Tabla 3.15: Cambio de contraseña.

3.5 Requisitos de rendimiento

Esta sección describe los requisitos relacionados con las propiedades del sistema en cuanto a la carga que este debe soportar.

Al tratarse de una aplicación web donde se tiene como objetivo compartir recetas, y dado que no hay ninguna restricción del lugar o país desde del que el usuario accede, el número de usuario puede ser elevado.

A continuación se especifican en una lista las características de rendimiento del sistema:

- El sistema debe ser capaz de dar soporte a usuarios a nivel internacional, a la orden de cientos de usuarios de forma simultanea.
- Se esperan almacenar del orden de miles de recetas.

3.6 Restricciones de diseño

El diseño es una parte esencial de la aplicación, parte que de detallará mas adelante en el apartado de **Diseño** de este documento donde gracias a los prototipos, se observa una primera aproximación del diseño de la aplicación web. En este apartado se detallan las restricciones principales del diseño de la aplicación, siendo estas:

- El sistema tiene como base color verde y tonos claros.
- La letra de la aplicación web debe ser lo suficientemente clara y las secciones deben estar bien definidas.
- Botones grandes y con un diseño suave, sin bordes puntiagudos.

3.7 Atributos de sistema

La siguiente sección presenta los atributos relacionados con la fiabilidad, portabilidad y seguridad de la aplicación. Se especifican los mecanismos de seguridad utilizados por el sistema y las tareas restringidas a cada usuario.

Al tratarse de una aplicación web donde no es primordial los tiempos de ejecución o la propia disponibilidad del sistema, como en un sistema bancario o una aplicación para el manejo de sistemas de navegación aérea, se tendrán en cuenta características basadas en la seguridad de las cuentas de usuarios, la integridad y concurrencia de los datos.

En la siguiente lista se detallan las cuestiones relacionadas con la fiabilidad y la mantenibilidad del sistema:

- El sistema al no ser de tipo crítico, su mantenibilidad debe asegurarse a nivel medio, siendo posible que la aplicación pueda estar inoperable por mantenimiento o por caída una cantidad de tiempo suficiente para satisfacer las necesidades de mantenibilidad, sin priorizar el tiempo.
- El sistema debe asegurar el manejo fiable de los datos personales de los usuarios, asegurando que su acceso únicamente por el usuario en cuestión.
- El sistema debe asegurar el acceso concurrente a los datos, asegurando que cualquier cambio en las recetas, sea visto por los usuarios lo antes posible.

CAPÍTULO 4

Análisis

A lo largo de este apartado se describe de forma detallada el diseño estructural de la aplicación, las diferentes funcionalidades requeridas, entidades, casos de uso y el diseño de la base de datos mediante diferentes tipos de diagramas que ayudan a entender la estructura y organización de la aplicación web.

4.1 Diseño estructural

Para la descripción del diseño estructural de la aplicación se define en primer lugar un diagrama de contexto, donde se representa de forma general y a grandes rasgos el contexto de la aplicación.

En segundo lugar se detalla la estructura del sistema, especificando las características, restricciones y las relaciones necesarias en el diagrama de clases, donde también se especificarán los métodos necesarios para el correcto desarrollo de las funciones anteriormente descritas.

Se usará para la construcción del diagrama estructural y del diagrama de clases el lenguaje de modelado de sistemas UML¹ ("*Unified Modeling Language*"), un estándar de lenguaje para especificar, visualizar, construir y documentar artefactos y sistemas software.

4.1.1. Diagrama de contexto

Los diagramas de contexto representan visualmente el alcance del software al mostrar el sistema software y sus interacciones con las personas y otros actores, en el caso de la aplicación web, los actores "Usuario" y "anónimo".

Como se puede observar en el diagrama de contexto representado por la figura 4.1, se presentan los diferentes usuarios definidos en la especificación de requisitos, que proporcionarían una entrada al sistema, encargado de administrar a los usuarios, la recetas y el foro, y la aplicación web se comunicará con el usuario mediante avisos, que notifican al usuario si la operación a realizar se ha ejecutado de manera correcta.

¹<https://www.uml.org/>



Figura 4.1: Diagrama de contexto

4.1.2. Diagrama de clases

El diagrama de clases representado en la figura 4.2 muestra la estructura estática del sistema modelado, las relaciones que existen entre las distintas clases y objetos del sistema. Los diagramas de clases son de gran utilidad para tener una idea clara acerca de la abstracción de un dominio y para formalizar el análisis de los conceptos relacionados, además de documentar una solución de diseño y la estructura del sistema que se va a implementar en términos de clases, sus objetos y métodos.

Para el desarrollo del proyecto, se tendrán en cuenta las siguientes clases y relaciones:

- **Usuario:** Id, email, name y password.
- **Role:** Id y rol.
- **Recetarios:** Id, title y description.
- **Receta:** Id, title, description, numComensales, imageUrl, type, dificultad, tags, ingredientes y body .
- **Comentario:** Id, body, user y date.
- **Post:** Id, name, body y userName.
- **Replies:** Id, body y userName.
- **Forum:** Id, name y description.

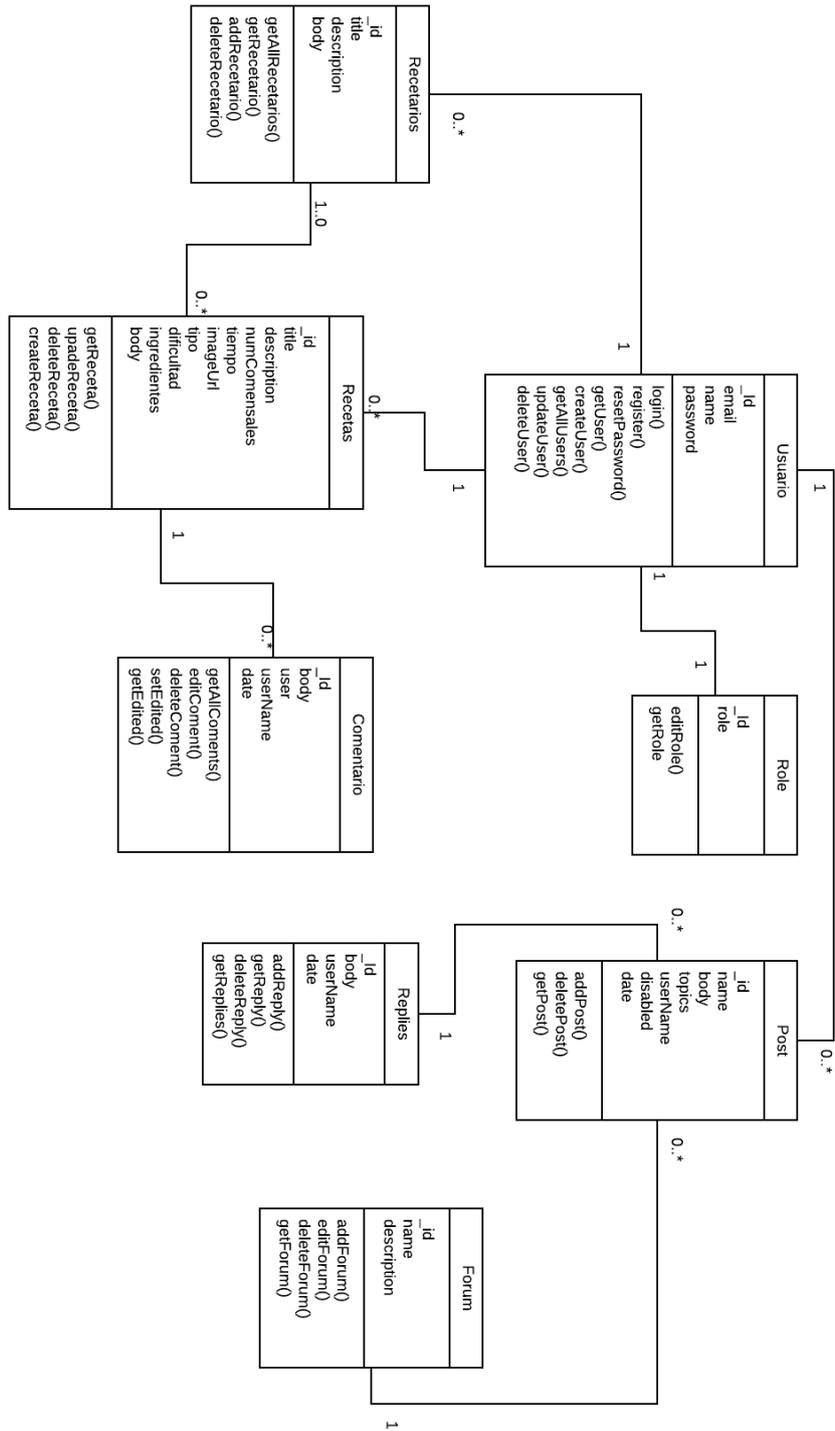


Figura 4.2: Diagrama de clases

4.2 Diagramas de casos de uso

Los diagramas de casos de uso describen las interacciones entre el usuario y el sistema software, modelan la funcionalidad del sistema usando un sistema de actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

Más concretamente, un caso de uso es una parte de un diagrama que describe un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta el sistema para producir un resultado observable de valor para el actor. Describe que hace el sistema, pero no cómo lo hace.

Los casos de uso son muy útiles para capturar el comportamiento deseado del sistema si tener que especificar como se implementa este comportamiento, además de dar soporte como medio de comprensión del sistema a los usuarios y desarrolladores.

Pero principalmente los diagramas de caso de sirven para ayudar a validar la arquitectura del sistema y verificar el sistema en el transcurso del desarrollo del mismo, siendo una pieza esencial para el correcto desarrollo de la aplicación web.

Para poder definir de forma correcta cada caso de uso se emplean una tabla que se dividen en los siguientes apartados[6]:

- **Descripción:** refleja uno o varios requisitos funcionales del sistema o una parte de un requisito.
- **Actores:** usuarios relacionados con el caso de uso.
- **Pre-condiciones:** condiciones que deben cumplirse para que se de el caso de uso.
- **Escenarios:** se incluye toda la descripción de todos los flujos de eventos dentro del caso de uso.
- **Pos-condiciones:** condiciones que se cumplen posteriormente al caso de uso.

A continuación se definen los casos de uso que contempla el sistema:

Caso de uso 1: Registro	
Descripción:	Alta (Usuario)
Actor:	Anónimo
Pre-condición:	El usuario no tiene cuenta en el sistema.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la página con el formulario de registro. 2. El usuario cumplimenta el formulario de registro. 3. Pulsa el botón de registro 4. El usuario es dado de alta en la base de datos. 5. El usuario el redirigido a su página personal. <p>Flujo de eventos excepcional 1 :</p> <ol style="list-style-type: none"> 1. El usuario cancela el registro. <p>Flujo de eventos excepcional 2 :</p> <ol style="list-style-type: none"> 1. Se produce un error en el sistema que impide el registro del usuario, las causas son: usuario o e-mail ya introducido en el sistema.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ■ El usuario es registrado en el sistema y posee una cuenta de tipo usuario. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ■ El usuario no puede terminar el proceso de registro, es notificado del problema y es redirigido a la página principal de la aplicación.

Tabla 4.1: Caso de uso 1: Registro

Caso de uso 2: Iniciar sesión	
Descripción:	Iniciar sesión en el sistema
Actor:	usuario
Pre-condición:	El usuario no ha iniciado sesión en el sistema
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de sesión. 2. El usuario cumplimenta los campos de usuario y contraseña. 3. El usuario selecciona el botón de 'Iniciar sesión'. 4. Es redirigido a su página principal. <p>Flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ■ El usuario cancela el inicio de sesión. <p>Flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ■ Se produce un error en el sistema que impide el inicio de sesión del usuario, causado por claves erróneas o una mala comunicación con el servidor.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ■ El usuario inicia sesión en el sistema. <p>Según el flujo de eventos excepcional 1 y 2 :</p> <ul style="list-style-type: none"> ■ El sistema no autentica al usuario e indica que las credenciales son erróneas.

Tabla 4.2: Caso de uso 2: Iniciar de sesión

Caso de uso 3 : Cerrar sesión	
Descripción:	Cerrar sesión en el sistema
Actor:	Usuario
Pre-condición:	El usuario tiene una sesión abierta en el sistema
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario selecciona el botón de "Cerrar sesión" y es llevado a la página principal de la aplicación.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ■ El usuario cierra sesión en el sistema.

Tabla 4.3: Caso de uso 3 : Cerrar sesión

Caso de uso 4: Búsqueda de una receta	
Descripción:	Buscar una receta en la aplicación
Actor:	Usuario, anónimo
Pre-condición:	El usuario desea buscar una receta en la aplicación
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario selecciona la barra de búsqueda. 2. Teclea los criterios de búsqueda. 3. Selecciona una receta sugerida por el motor de búsqueda. <p>Flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ■ El usuario cancela la búsqueda. <p>Flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ■ No se encuentran recetas según los criterios de búsqueda. <p>Flujo de eventos excepcional 3 :</p> <ul style="list-style-type: none"> ■ Se produce un error en el sistema que impide la búsqueda de las recetas.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ■ Se le redirige a la receta seleccionada. <p>Según el flujo de eventos excepcional 2 y 3 :</p> <ul style="list-style-type: none"> ■ El usuario se mantiene en la página donde selecciono la barra de búsqueda.

Tabla 4.4: Caso de uso 04: Búsqueda de una receta

Caso de uso 5: Dar de baja una cuenta de usuario	
Descripción:	Un usuario desea dar de baja una cuenta de tipo Usuario
Actor:	Usuario
Pre-condición:	El usuario debe tener una cuenta dada de alta en el sistema.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a su página de perfil de usuario. 2. Selecciona el botón de “Darse de baja”. 3. Confirma el proceso de baja. 4. El usuario se da de baja de la aplicación. <p>Flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ■ El usuario cancela el proceso de baja. <p>Flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ■ Se produce un error en el sistema que impide la baja de la aplicación.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ■ El sistema borra la cuenta de usuario del sistema junto con sus recetas. <p>Según el flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ■ Se notifica el error y vuelve a la página de perfil.

Tabla 4.5: Caso de uso 5: Dar de baja una cuenta de usuario

Caso de uso6: Dar de alta una receta	
Descripción:	El usuario da de alta una receta en el sistema.
Actor:	Administrador, usuario
Pre-condición:	<ol style="list-style-type: none"> 1. El usuario esta dado de alta en el sistema. 2. El usuario ha iniciado sesión en el sistema.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la página de gestión de recetas. 2. Selecciona el botón de alta de receta. 3. Accede a la página con el formulario de alta de receta. 4. Cumplimenta el formulario de alta de receta. 5. Se valida el formulario 6. Se añade la receta a su lista de recetas. 7. Se notifica al usuario con el alta de la receta. <p>Flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ▪ El usuario cancela el alta de la receta. <p>Flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ No se valida de forma correcta el formulario de la receta. <p>Flujo de eventos excepcional 3 :</p> <ul style="list-style-type: none"> ▪ Se produce un error en el sistema.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ La receta se añade exitosamente al sistema. <p>Según el flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ▪ El usuario es redirigido a la página de gestión de recetas. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ Se notifica el error y puede corregir el formulario de alta. <p>Según el flujo de eventos excepcional 3 :</p> <ul style="list-style-type: none"> ▪ Se notifica el error el redirigido a la página de gestión de recetas.

Tabla 4.6: Caso de uso 6: Dar de alta una receta

Caso de uso 7: Editar una receta	
Descripción:	El usuario edita una receta.
Actor:	Administrador, usuario
Pre-condición:	<ol style="list-style-type: none"> 1. El usuario esta dado de alta en el sistema. 2. El usuario ha iniciado sesión en el sistema. 3. El usuario tiene recetas dadas de alta en el sistema. 4. Deben haber recetas dadas de alta en el sistema.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la página de gestión de recetas. 2. Selecciona el botón edición de receta. 3. Selecciona la receta que desea editar. 4. Accede a la página con el formulario para editar la receta con los datos cumplimentados. 5. Edita la receta. 6. Se valida el formulario 7. Se notifica al usuario que los cambios han sido guardados de forma correcta. <p>Flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ▪ El usuario cancela el proceso de editar la receta. <p>Flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ No se valida de forma correcta el formulario de la receta. <p>Flujo de eventos excepcional 3 :</p> <ul style="list-style-type: none"> ▪ Se produce un error en el sistema.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ La receta es editada de forma correcta. <p>Según el flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ▪ El usuario es redirigido a la página de gestión de recetas. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ Se notifica el error y puede corregir el formulario de alta. <p>Según el flujo de eventos excepcional 3 :</p> <ul style="list-style-type: none"> ▪ Se notifica el error el redirigido a la página de gestión de recetas.

Tabla 4.7: Caso de uso 7: Editar una receta

Caso de uso 8: Borrar una receta	
Descripción:	El usuario quiere borrar una receta
Actor:	Administrador, usuario
Pre-condición:	<ol style="list-style-type: none"> 1. El usuario ha de tener una cuenta dada de alta en el sistema. 2. El usuario ha de tener una sesión iniciada en el sistema. 3. Deben haber recetas dadas de alta en el sistema.
Escenarios:	<p>Flujo de eventos principal:</p> <ol style="list-style-type: none"> 1. Accede a la cuenta de un Usuario. 2. Accede a la sección de "Gestión de recetas". 3. Selecciona el botón de "Borrar receta" en la receta seleccionada. 4. Confirma el borrado. <p>Flujo de eventos excepcional 1 :</p> <ol style="list-style-type: none"> 1. El usuario cancela el borrado de la receta. <p>Flujo de eventos excepcional 2 :</p> <ol style="list-style-type: none"> 1. Se produce un error en el sistema a la hora de borrar la receta.
Pos-condición:	<p>Según el flujo de eventos principal 1 y 2:</p> <ul style="list-style-type: none"> ▪ El sistema borra la receta. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ El sistema notifica al usuario del error y regresa a la página principal de gestión de recetas del usuario..

Tabla 4.8: Caso de uso 8: Borrar una receta

Caso de uso 9: Comentar una receta	
Descripción:	El usuario quiere comentar una receta que ha visitado.
Actor:	Usuario, anónimo
Pre-condición:	El usuario ha visitado una de la recetas de la aplicación.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la sección dedicada para comentar la receta. 2. Escribe un comentario en la sección indicada. 3. Selecciona el botón de 'Publicar'. 4. El sistema registra el comentario. <p>Flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ▪ Se produce un error a la hora de registrar el comentario.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ El comentario es publicado en la receta. <p>Según el flujo de eventos excepcional 1 :</p> <ul style="list-style-type: none"> ▪ Se notifica al usuario del error y se mantiene en la página de la receta.

Tabla 4.9: Caso de uso 9: Comentar una receta

Caso de uso 10: Compartir una receta	
Descripción:	El usuario quiere compartir una receta
Actor:	Usuario, anónimo
Pre-condición:	El usuario ha visitado una receta.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la sección de la receta que quiere comentar. 2. Selecciona el botón de 'Compartir receta'. 3. Copia el enlace a la receta. 4. El usuario comparte la receta.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ El sistema provee al usuario de un enlace para compartir recetas.

Tabla 4.10: Caso de uso 10: Compartir una receta

Caso de uso 11: Crear un recetario	
Descripción:	El usuario quiere crear un recetario para guardar las recetas.
Actor:	Administrador, usuario
Pre-condición:	<ol style="list-style-type: none"> 1. El usuario ha de tener una cuenta dada de alta en el sistema. 2. El usuario ha de tener una sesión iniciada en el sistema.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la sección de “Gestión de recetas”. 2. Selecciona el botón de ‘Crear un recetario’. 3. Cumplimenta el formulario de creación. 4. El usuario crea el recetario. <p>Flujo de eventos excepcional 1 :</p> <ol style="list-style-type: none"> 1. El usuario cancela la creación del recetario. <p>Flujo de eventos excepcional 2 :</p> <ol style="list-style-type: none"> 1. Se produce un error en el sistema a la hora de crear un recetario.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ El sistema crea el recetario en el sistema. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ El sistema notifica al usuario del error y se mantiene en la página de gestión de recetas.

Tabla 4.11: Caso de uso 11: Crear un recetario

Caso de uso 12: Guardar una receta en el recetario	
Descripción:	El usuario quiere guardar una receta en el recetario.
Actor:	Usuario
Pre-condición:	<ol style="list-style-type: none"> 1. El usuario ha de tener una cuenta dada de alta en el sistema. 2. El usuario ha de tener una sesión iniciada en el sistema. 3. Deben haber recetas dadas de alta en el sistema. 4. Deben haber recetarios dados de alta en el sistema.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede la receta. 2. Selecciona el botón de 'Guardar'. 3. Selecciona una recetario. 4. El usuario guarda la receta. <p>Flujo de eventos excepcional 1 :</p> <ol style="list-style-type: none"> 1. El usuario cancela el proceso de guardar la receta. <p>Flujo de eventos excepcional 2 :</p> <ol style="list-style-type: none"> 1. Se produce un error en el sistema a la hora de guardar la receta.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ El sistema guarda la receta en el recetario. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ El sistema notifica al usuario del error y vuelve a la página de la receta.

Tabla 4.12: Caso de uso 12: Guardar una receta en el recetario

Caso de uso 13: Dar de alta un tema en el foro	
Descripción:	El usuario quiere crear un tema de discusión en el foro.
Actor:	Administrador, usuario
Pre-condición:	<ol style="list-style-type: none"> 1. El usuario ha de tener una cuenta dada de alta en el sistema. 2. El usuario ha de tener una sesión iniciada en el sistema.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la sección de 'Foro' de la página web. 2. Selecciona el botón de 'Crear un tema' en el foro. 3. Cumplimenta el formulario de creación de tema en el foro. 4. El usuario crea el tema de discusión. 5. El sistema muestra el tema creado en el foro. <p>Flujo de eventos excepcional 1 :</p> <ol style="list-style-type: none"> 1. El usuario cancela la creación del tema de discusión. <p>Flujo de eventos excepcional 2 :</p> <ol style="list-style-type: none"> 1. Se produce un error en el sistema a la hora de el tema de discusión.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ El sistema crea el tema de discusión en el foro y se mantiene en la página. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ El sistema notifica al usuario del error y se mantiene en a la página principal de foros.

Tabla 4.13: Caso de uso 13: Dar de alta un tema en el foro

Caso de uso 14: Cerrar un tema en el foro	
Descripción:	El usuario quiere cerrar un tema de discusión en el foro.
Actor:	Usuario
Pre-condición:	<ol style="list-style-type: none"> 1. El usuario ha de tener una cuenta dada de alta en el sistema. 2. El usuario ha de tener una sesión iniciada en el sistema. 3. El usuario ha de tener temas de discusión en el foro.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la sección de "Foro" de la página web. 2. Selecciona un tema de discusión. 3. Selecciona el botón de 'Cerrar discusión'. 4. El usuario confirma el proceso de cerrar el tema de discusión. <p>Flujo de eventos excepcional 1 :</p> <ol style="list-style-type: none"> 1. El usuario cancela el cierre del tema de discusión. <p>Flujo de eventos excepcional 2 :</p> <ol style="list-style-type: none"> 1. Se produce un error en el sistema a la hora de el tema de discusión.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ El sistema cierra el tema de discusión en el foro. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ El sistema notifica al usuario del error y regresa a la página principal de foros.

Tabla 4.14: Caso de uso 14: Cerrar un tema en el foro

Caso de uso 15: Responder un tema del el foro	
Descripción:	El usuario quiere crear una entrada en el tema de discusión en el foro.
Actor:	Administrador, usuario
Pre-condición:	<ol style="list-style-type: none"> 1. El usuario ha de tener una cuenta dada de alta en el sistema. 2. El usuario ha de tener una sesión iniciada en el sistema. 3. Deben haber temas en el foro de discusión dados de alta en el foro.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la sección de 'Foro' de la página web. 2. Selecciona el botón de 'Reply' en el foro. 3. Cumplimenta el formulario de una respuesta al tema. 4. El usuario confirma la creación de la respuesta. 5. EL sistema muestra la respuesta en el tema de discusión. <p>Flujo de eventos excepcional 1 :</p> <ol style="list-style-type: none"> 1. El usuario cancela la creación de la respuesta. <p>Flujo de eventos excepcional 2 :</p> <ol style="list-style-type: none"> 1. Se produce un error en el sistema a la hora de elaborar la respuesta.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ El sistema crea la respuesta en el foro. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ El sistema notifica al usuario del error y regresa a la página de dicho tema.

Tabla 4.15: Caso de uso 15: Responder un tema del el foro

Caso de uso 16: Cambiar contraseña	
Descripción:	El usuario quiere cambiar la contraseña de su cuenta de usuario.
Actor:	Administrador, usuario
Pre-condición:	<ol style="list-style-type: none"> 1. El usuario ha de tener una cuenta dada de alta en el sistema. 2. El usuario ha de tener una sesión iniciada en el sistema.
Escenarios:	<p>Flujo de eventos principal :</p> <ol style="list-style-type: none"> 1. El usuario accede a la página de perfil de usuario. 2. Selecciona el botón de “Cambiar contraseña”. 3. El usuario cumplimenta el formulario de cambio de contraseña, indicando la contraseña nueva. 4. EL sistema cambia la contraseña del usuario. <p>Flujo de eventos excepcional 1 :</p> <ol style="list-style-type: none"> 1. El usuario cancela el proceso de cambio de contraseña. <p>Flujo de eventos excepcional 2 :</p> <ol style="list-style-type: none"> 1. Se produce un error en el sistema a la hora de cambiar la contraseña.
Pos-condición:	<p>Según el flujo de eventos principal :</p> <ul style="list-style-type: none"> ▪ El sistema cambia su contraseña. <p>Según el flujo de eventos excepcional 2 :</p> <ul style="list-style-type: none"> ▪ El sistema notifica al usuario del error y regresa a la página de perfil del usuario.

Tabla 4.16: Caso de uso 16: Comentar en el foro

4.2.1. Modelado por actores

En esta sección se va a especificar en varios diagramas los casos de uso agrupados por actores, especificados anteriormente en la descripción de casos de uso.

En el sistema se contemplan 2 tipos de roles: usuario y anónimo.

Actor: Usuario

Rol usuario: la figura 4.2 representa la imagen del diagrama de casos de uso para el actor "Usuario".

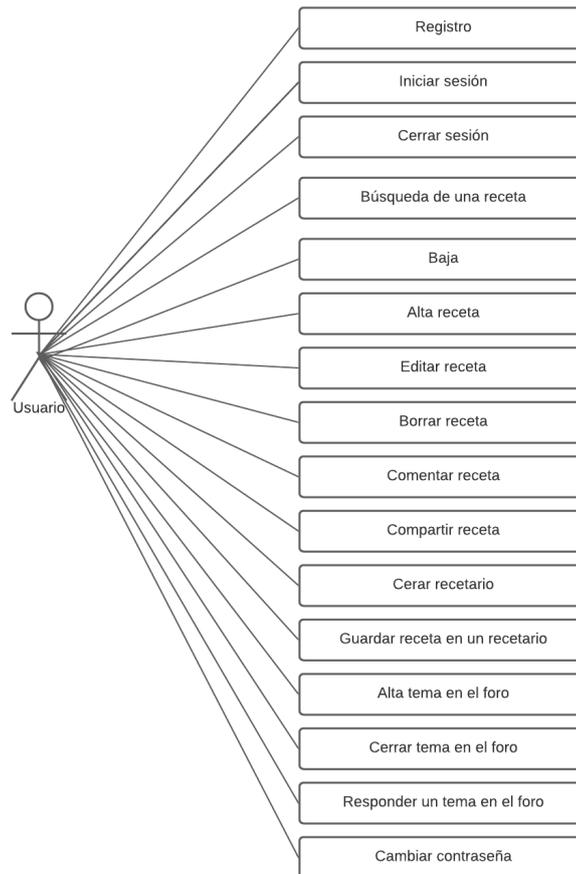


Figura 4.3: Diagrama de casos de uso para el rol Usuario

Actor: Anónimo

Rol anónimo: la figura 4.3 representa la imagen del diagrama de casos de uso para el actor "Anónimo".

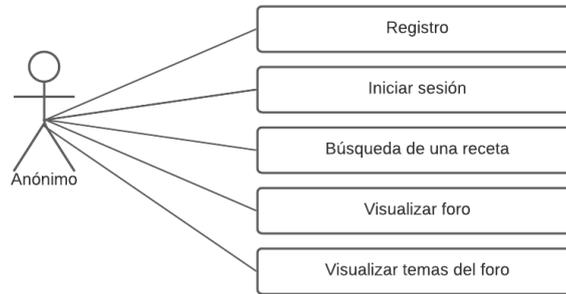


Figura 4.4: Diagrama de casos de uso para el rol Anónimo

CAPÍTULO 5

Diseño

El siguiente apartado describe la fase de diseño del desarrollo software, en el cual se van a especificar los detalles acerca del diseño de la interfaz gráfica de la aplicación y la estructura de los datos.

El diseño del Software tiene un impacto directo sobre la capacidad del sistema para cumplir o no el total de requerimientos establecidos. Un error de diseño en esta fase puede acarrear problemas en todo el proyecto y provocar que este caiga en una espiral de continuos cambios y de rehacer constantemente el trabajo.

5.1 Diseño Arquitectónico: modelo, vista, controlador

El modelo vista controlador[2] es un patrón de diseño arquitectónico software que funciona de la siguiente manera: el modelo se divide en 3 partes bien diferenciadas entre si, en primer lugar el modelo, siendo la estructura básica de los datos como las recetas, los recetarios o el propio usuario, después un controlador que se encarga de manejar los datos y realizar diferentes operaciones según el usuario, y por último la capa visible al usuario de la aplicación, la vista, donde se representan los datos en una interfaz que implementa diferentes funciones para la manipulación de estos.

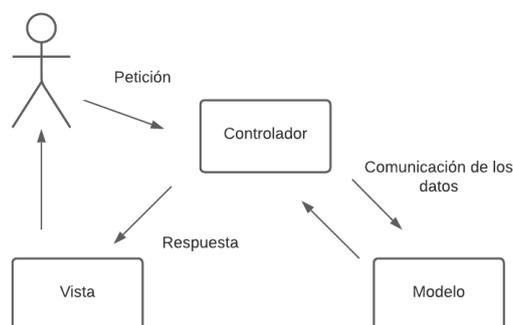


Figura 5.1: Patrón arquitectónico MVC

Como se puede observar en la figura 5.1, el usuario se encarga de interactuar con la interfaz, la vista, y realizar peticiones al controlador, capa que se encarga de la manipulación de los datos. En el caso de la aplicación web ambos, el *back-end* como el *front-end*, implementan este patrón arquitectónico.

5.2 Diseño de la interfaz gráfica de la aplicación web

A lo largo de este apartado se van a representar los prototipos del diseño de la interfaz a implementar. Para ello se ha de tener en cuenta los diferentes casos de uso que se han planteado y los requisitos especificados en el capítulo 3.

Para el desarrollo de los prototipos se emplea la herramienta **Justinmid** que permite desarrollar prototipos con funciones sencillas como declaración de variables o redirección de páginas. Para este caso, se van a implementar prototipos a un nivel de detalle suficiente para presentar los elementos clave y así satisfacer los casos de uso que se han planteado anteriormente.



Figura 5.2: Logotipo de JustInMind

5.2.1. Representación de los prototipos

A continuación se representan los prototipos funcionales de la aplicación donde cada imagen representa una solución de diseño para cada caso de uso planteado. Al ser prototipos, la implementación final de la interfaz puede cambiar.

The screenshot shows a registration form for "FoodCompanion". At the top, it says "Bienvenido a FoodCompanion" with a subtitle "Descubre recetas, conecta con otros cocineros alrededor del mundo y prueba comida increíble!!!". Below this are three input fields: "Email", "Usuario", and "Contraseña". The "Contraseña" field has a "Mostrar" link next to it. Underneath the password field, there are two lines of requirements: "- Ocho caracteres mínimo - Uno numero" and "- Un caracter especial mínimo". At the bottom of the form is a button labeled "Crear cuenta". The entire form is enclosed in a light green border, and a solid green vertical bar is on the right side of the image.

Figura 5.3: Alta de un usuario

Alta de un usuario: la figura 5.3 representa la página para dar de alta un usuario en el sistema.



Iniciar sesión

¿No tienes una cuenta? [Crear una cuenta](#)

Usuario

Contraseña [Mostrar](#)

Mantener sesión iniciada

[¿Has olvidado la contraseña?](#) [¿Has olvidado el usuario?](#)

Figura 5.4: Alta de un usuario

Inicio de sesión: la figura 5.4 muestra la página para iniciar sesión.

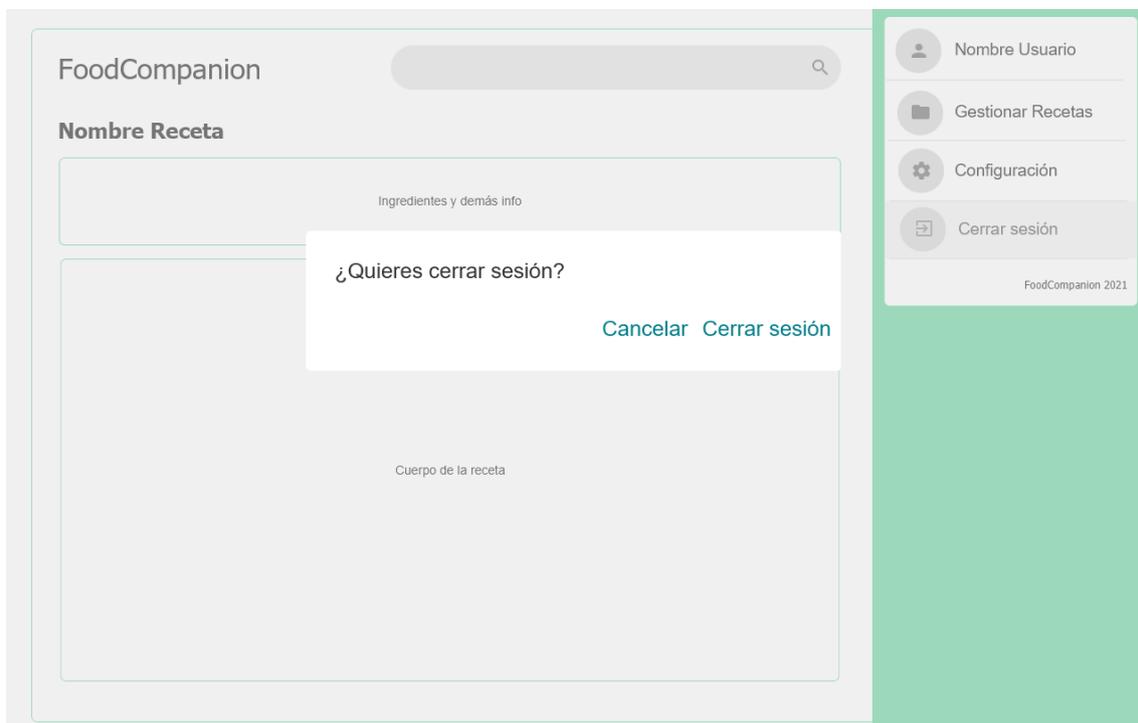


Figura 5.5: Cerrar sesión

Cerrar sesión: la figura 5.5 representa un cerrar sesión en el sistema.

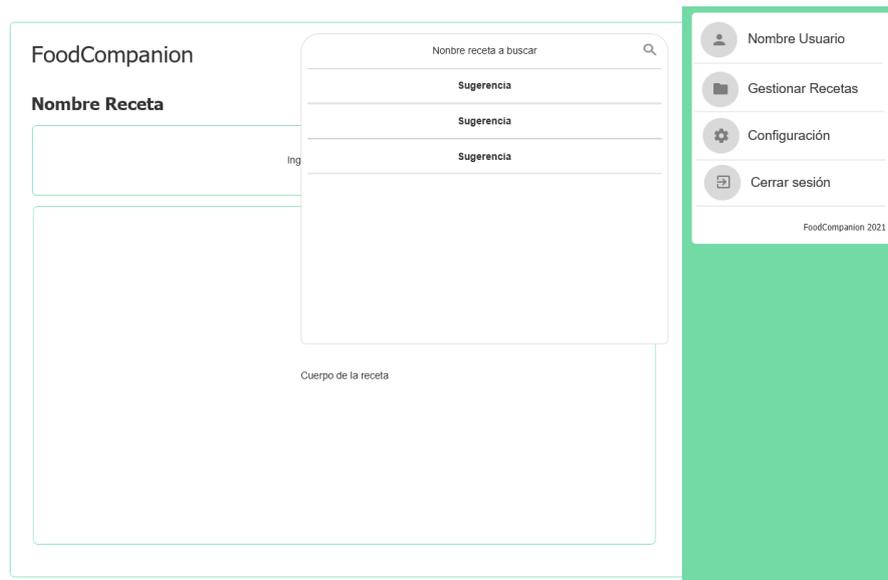


Figura 5.6: Búsqueda receta

Búsqueda receta: la figura 5.6 representa la página para buscar una receta en el sistema.

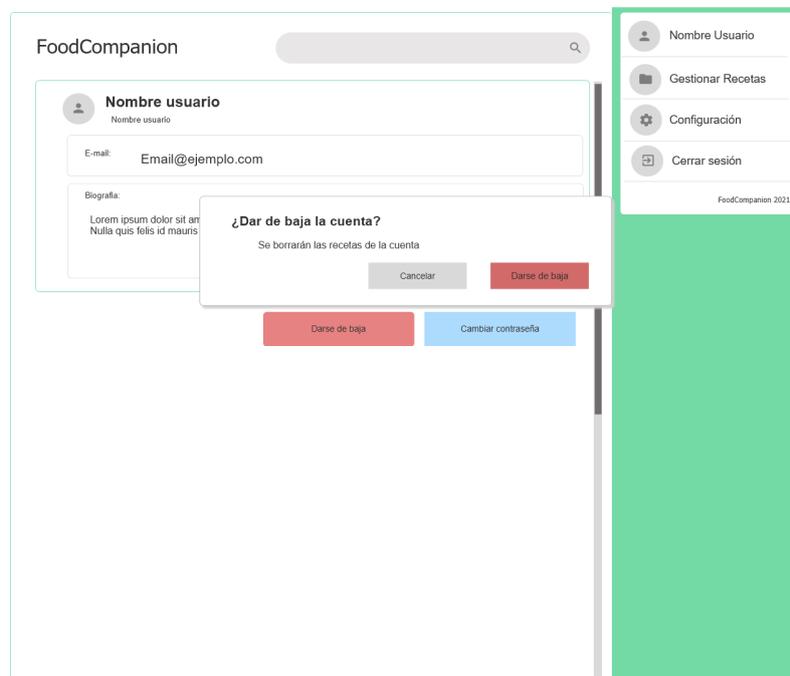


Figura 5.7: Baja de una cuenta de usuario

Baja usuario: la figura 5.7 representa la página junto con el desplegable de confirmación para el proceso de baja de un usuario en el sistema.

The screenshot shows the 'Dar de alta una receta' (Add a recipe) form in the FoodCompanion application. The form is titled 'Dar de alta una receta' and includes the following fields and components:

- Nombre de la receta:** A text input field with the placeholder text 'Nombre receta'.
- Tags de la receta:** A list of tags, currently containing 'Tag receta 1'.
- Lista de ingredientes:** A list of ingredients, currently containing 'Ingrediente 1' and 'Ingrediente 2'.
- Descripción de la receta:** A text input field with the placeholder text 'Descripción receta'.
- Desarrollo de la receta:** A rich text editor with a toolbar and a text area containing the placeholder text 'Desarrollo receta'.

At the bottom of the form are two buttons: 'Cancelar' and 'Dar de alta'. To the right of the form is a sidebar with the following items:

- Nombre Usuario
- Gestionar Recetas
- Configuración
- Cerrar sesión

The sidebar also includes the text 'FoodCompanion 2021' at the bottom.

Figura 5.8: Dar de alta una receta

Dar de alta una receta: la figura 5.8 representa el formulario de alta de una receta en el sistema.

The screenshot shows the 'Editar una receta' (Edit a recipe) form in the FoodCompanion application. The form is titled 'Editar una receta' and includes the following fields and components:

- Nombre de la receta:** A text input field with the placeholder text 'Nombre receta'.
- Tags de la receta:** A list of tags, currently containing 'Tag receta 1'.
- Lista de ingredientes:** A list of ingredients, currently containing 'Ingrediente 1' and 'Ingrediente 2'.
- Descripción de la receta:** A text input field with the placeholder text 'Descripción receta'.
- Desarrollo de la receta:** A rich text editor with a toolbar and a text area containing the placeholder text 'Nombre receta' and a block of Lorem Ipsum text.

At the bottom of the form are two buttons: 'Cancelar' and 'Guardar receta'. To the right of the form is a sidebar with the following items:

- Nombre Usuario
- Gestionar Recetas
- Configuración
- Cerrar sesión

The sidebar also includes the text 'FoodCompanion 2021' at the bottom.

Figura 5.9: Editar una receta

Editar una receta: la figura 5.9 representa la sección de la página donde se puede editar una receta.

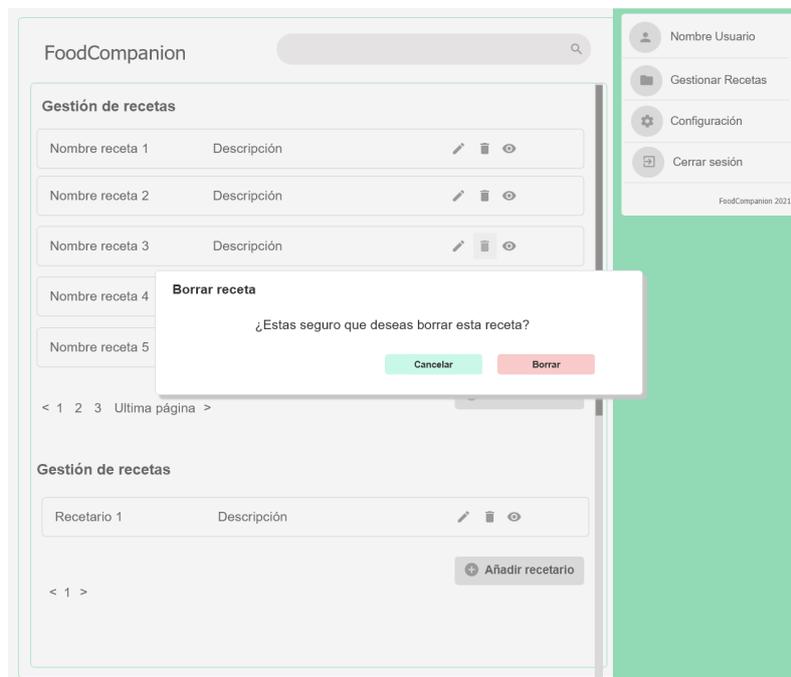


Figura 5.10: Borrar una receta

Borrar una receta: la figura 5.10 representa la página web junto con una alerta de confirmación para el borrado de la receta.

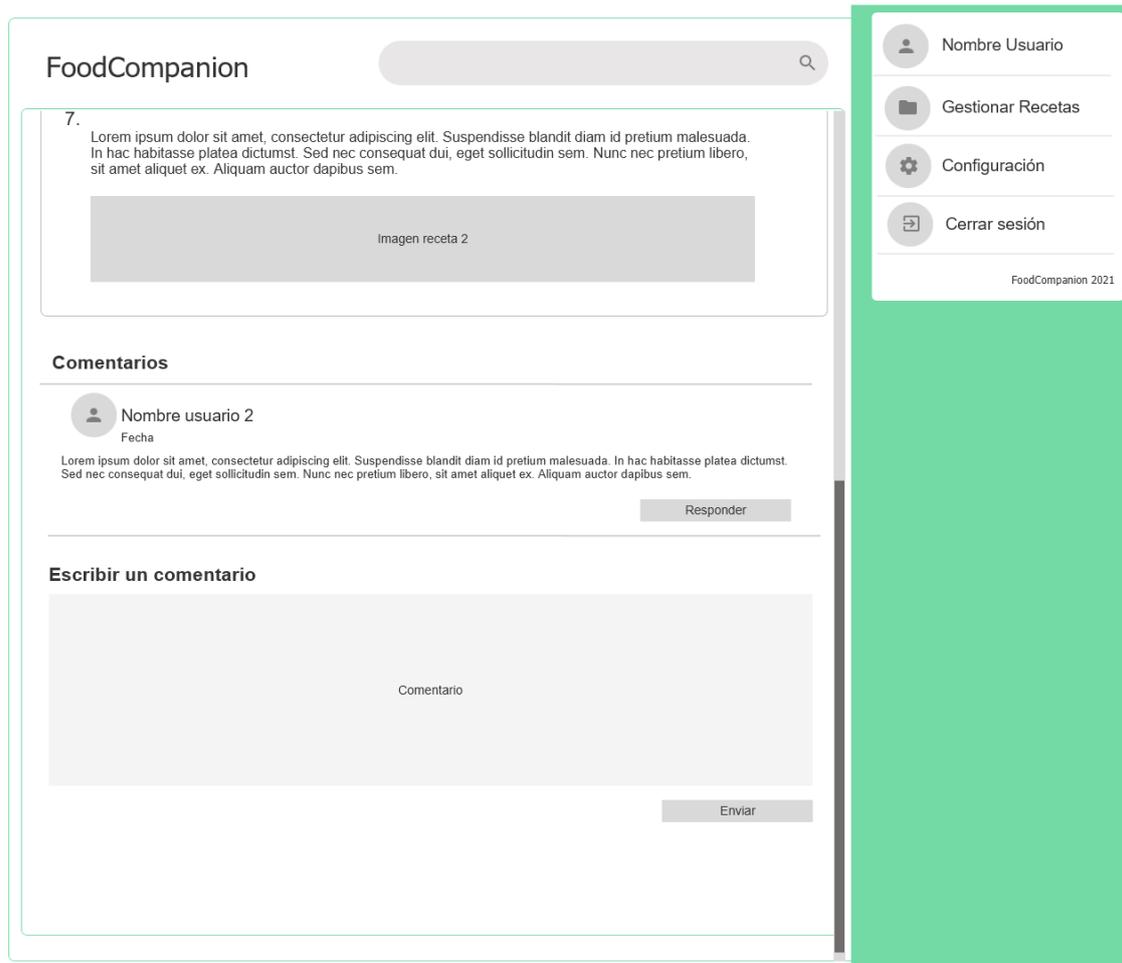


Figura 5.11: Comentar una receta

Comentar una receta: la figura 5.11 representa la sección de comentarios de una receta.

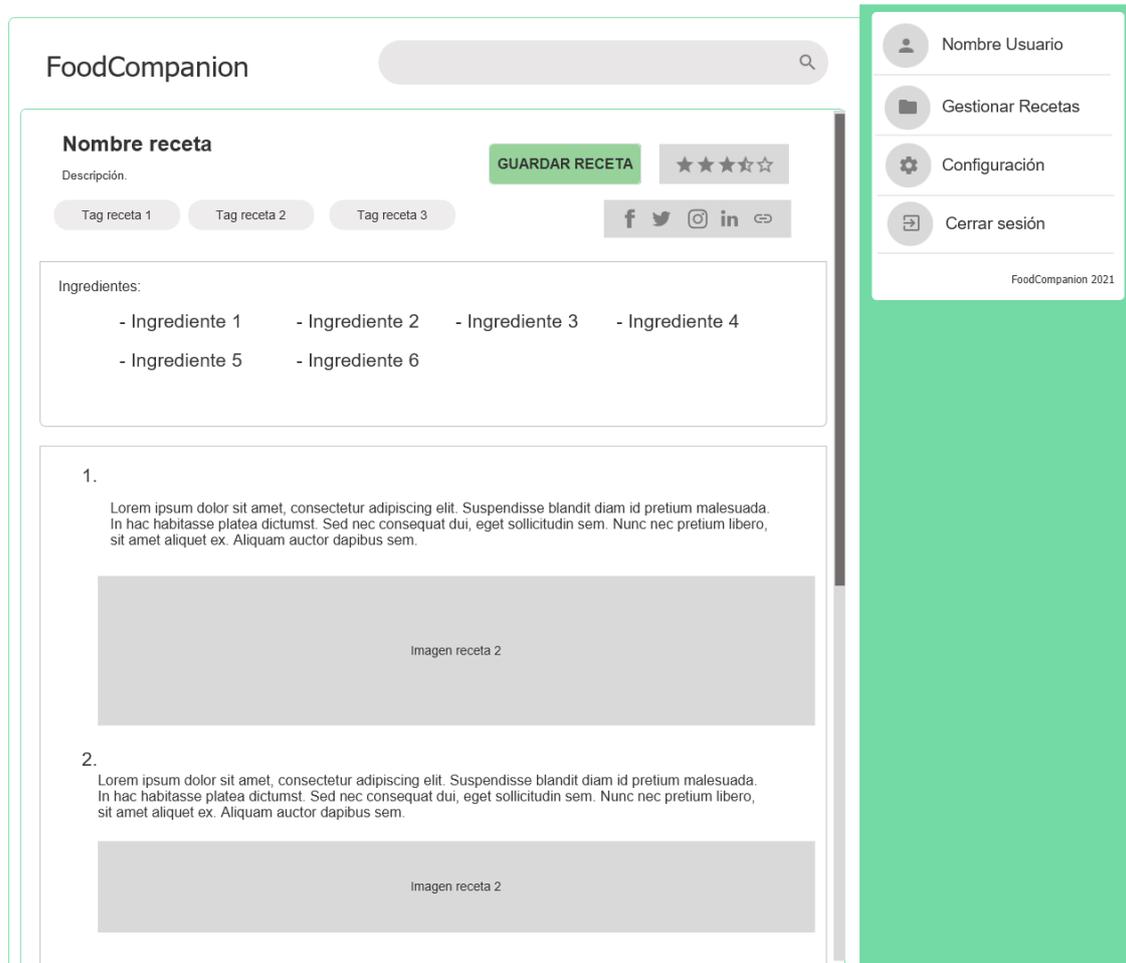


Figura 5.12: Representación de una receta

Página de una receta: la figura 5.12 representa la página principal de una receta, donde se pueden observar las secciones de:

- Redes sociales y la obtención de un enlace a la receta.
- El botón para guardar una receta.
- Las etiquetas de la receta.
- Los ingredientes de la receta.
- El cuerpo de la receta.

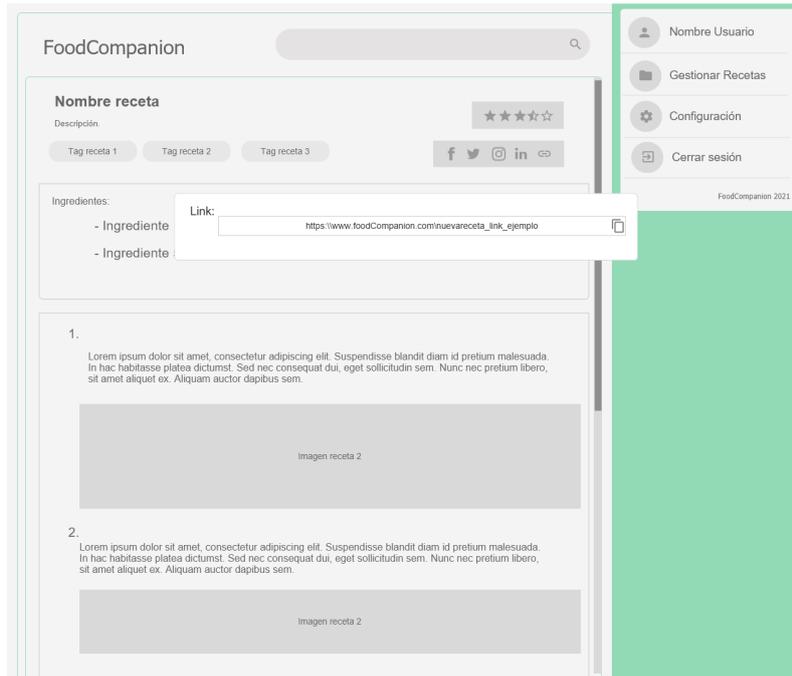


Figura 5.13: Compartir una receta vía enlace

Compartir una receta vía enlace: la figura 5.13 representa el proceso de copiar un enlace de una receta.

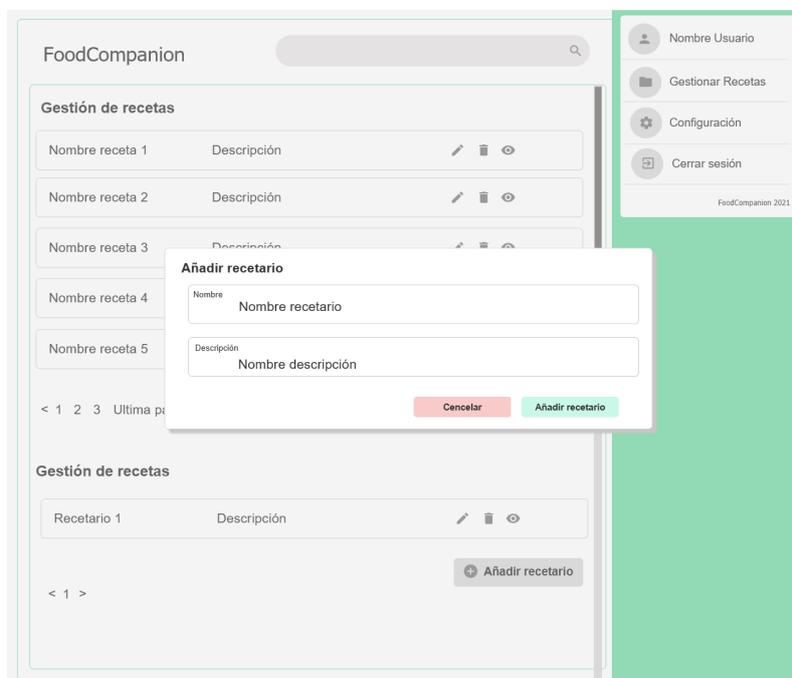


Figura 5.14: Crear una recetario

Crear una recetario: la figura 5.14 muestra el proceso de creación un recetario.

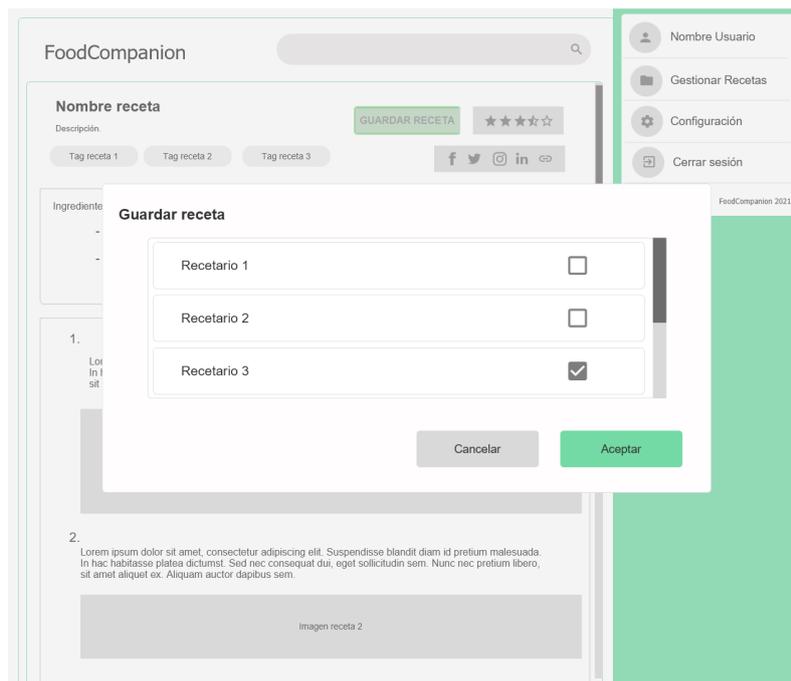


Figura 5.15: Guardar una receta en el recetario

Guardar una receta en el recetario: la figura 5.15 muestra el proceso para guardar una receta en un recetario dado.

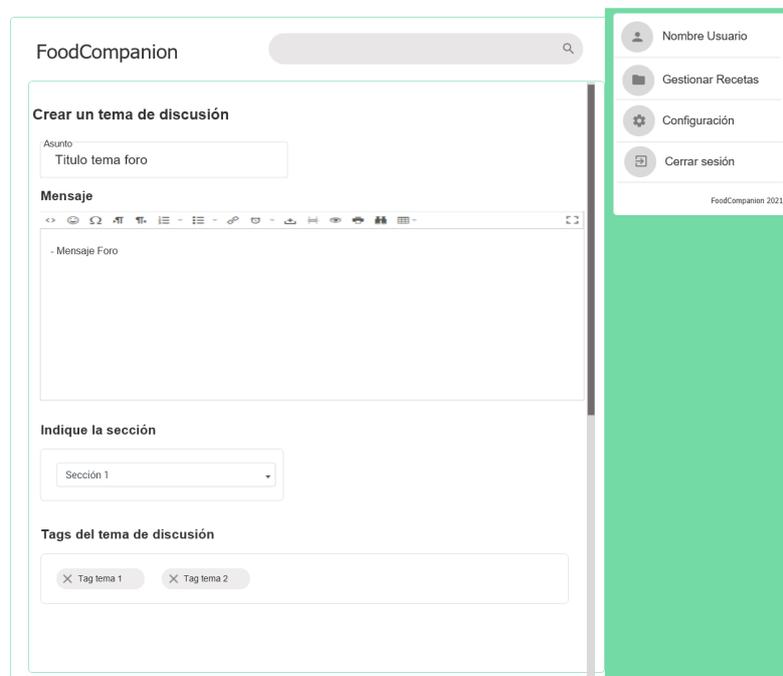


Figura 5.16: Dar de alta un tema en el foro

Dar de alta un tema en el foro: la figura 5.16 representa el proceso de alta de un tema en el foro.

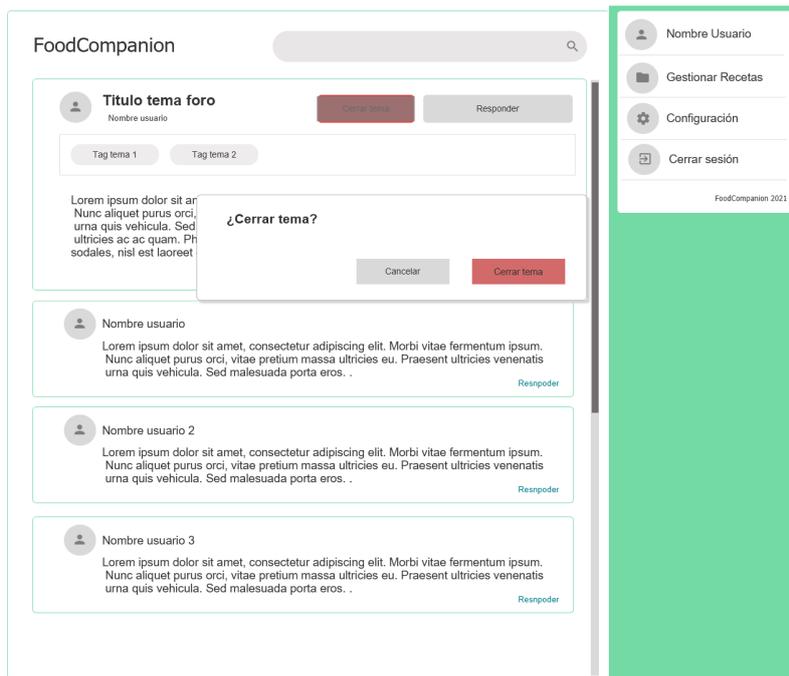


Figura 5.17: Cerrar un tema en el foro

Cerrar un tema en el foro: la figura 5.17 representa el proceso de baja de un tema en el foro.

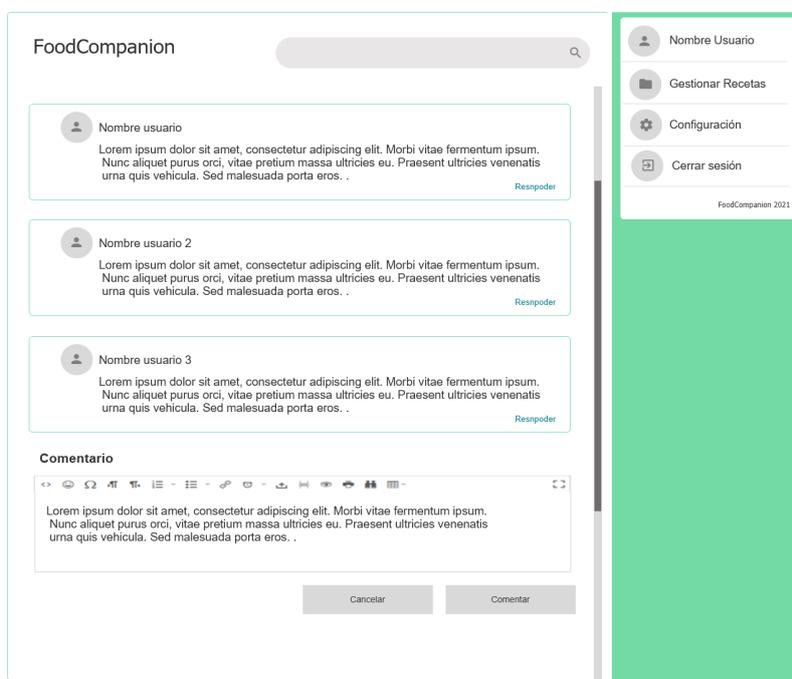


Figura 5.18: Comentar en el foro

Comentar en el foro: la figura 5.18 imagen representa la página de participación en el foro.

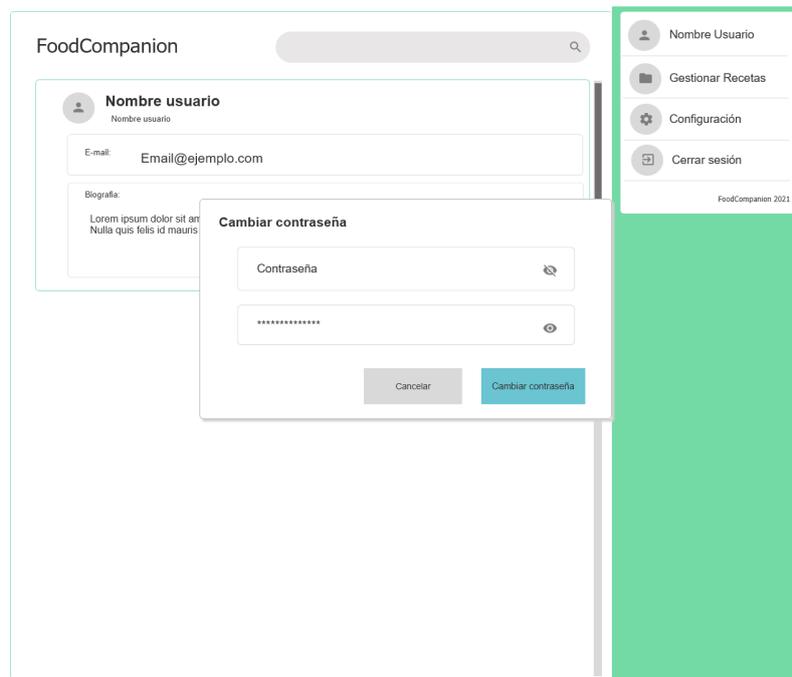


Figura 5.19: Cambiar contraseña

Cambiar contraseña: la figura 5.19 imagen representa el proceso de cambio de contraseña por parte del usuario.

CAPÍTULO 6

Implementación

Junto con la especificación de requisitos y el prototipo ya elaborado en la fase de diseño, se puede empezar con la fase de implementación. A lo largo de esta fase se describen detalladamente las tecnologías que se van a emplear para el desarrollo de la aplicación, junto con sus herramientas de desarrollo.

6.1 Back-End

En esta sección se van a detallar las principales tecnologías que se van a emplear para la implementación del **Back-End**, que se va a encargar de manejar la lógica de la aplicación, garantizar la seguridad y proporcionar las funciones necesarias para satisfacer las necesidades de los casos de uso planteados anteriormente.

6.1.1. Tecnologías

- **Spring**¹: *framework* de desarrollo que provee un modelo de programación y configuración para aplicaciones basadas en Java. *Spring* proporciona las herramientas necesarias para el fácil desarrollo de aplicaciones sin la necesidad de estar atados a configuraciones de despliegue específicas.
- **Spring Boot**²: herramienta que facilita la creación un proyecto basado en *Spring*. Gracias a *Spring Boot* se pueden añadir los módulos y las dependencias necesarias al proyecto y para construir una aplicación completa.
- **MySQL**³: Servicio de gestión de base de datos relacionales de código abierto respaldado por Oracle⁴ y basado en el lenguaje de consulta SQL.

6.1.2. Elección de las tecnologías

Se han escogido estas tecnologías en base a diferentes razones, la cuales son:

- El *back-end* que se va a implementar esta desarrollado íntegramente en Java, uno de los lenguajes de desarrollo *software* más usados por los programadores. Java es un

¹<https://spring.io/>

²<https://spring.io/projects/spring-boot>

³<https://www.mysql.com/>

⁴<https://www.oracle.com/es/index.html>

lenguaje muy extendido, apoyado por una vasta documentación y con una comunidad que gracias a una gran variedad de foros y guías, es fácil resolver problemas y poder construir la aplicación deseada.

- *Spring* ofrece una extensa cantidad de módulos que se usarán como apoyo para implementar de manera rápida y segura el servicio API *REST*, interfaz a la cual el *front-end* hará peticiones mediante el protocolo HTTP.
- Gracias a *Spring* se manejan las operaciones de gestión de base de datos de manera muy sencilla, sin necesidad de saber el lenguaje de gestión SQL de manera integral.
- La seguridad es uno de los aspectos clave de la aplicación y *Spring* ofrece una serie de módulos para el manejo de sesiones y control de acceso a las funciones accesibles únicamente a los usuarios autenticados en el sistema.
- Son tecnologías multiplataforma, portables y ligeras, por lo cual se puede trabajar desde cualquier parte sin necesidad de poseer un ordenador con grandes prestaciones.
- Gestión de memoria eficiente y rápida.
- *Spring* ofrece el manejo de datos en formato JSON, permitiendo el intercambio entre el *front-end* y el *back-end*.

6.1.3. Implementación del Back-End

El *back-end* esta compuesto por diferentes capas, las cuales se definen a continuación:

1. **Modelos:** los modelos identifican la definición básica de cada objeto que se van a manejar en la base de datos.
2. **Repositorios:** es la capa de acceso a la base de datos, que se encargará de las operaciones CRUD (Create, Read, Update and Delete).
3. **Services:** se encargarán de acceder a las funciones de los repositorios.
4. **Resources:** capa que ofrecerá las funciones definidas en los servicios a los clientes mediante peticiones HTTP. Mediante los *resources* se manejarán los *inputs*, enviados por la aplicación cliente y decidirá cuales son los *outputs* de respuesta.

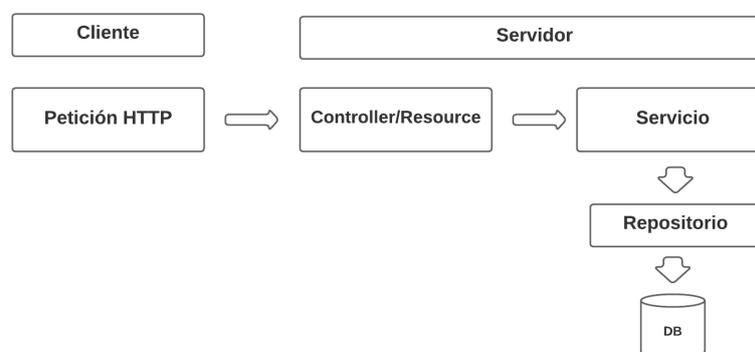


Figura 6.1: Descripción del diseño del servidor

Observando la figura 6.1 se pueden identificar las diferentes capas de las que consta el servidor, las peticiones son recibidas por los **controllers/resources** que manejan los

datos de entrada, luego se llaman a las funciones definidas en los **servicios** que funcionan como una capa de abstracción a los manejadores de la base de datos, los **repositorios**, que manipulan de forma directa los modelos indexados en la base de datos.

La figura 6.2 se representa la estructura de carpetas del *back-end*, donde se definen las siguientes carpetas:

- **Assemblers:** clases en las que el servidor de apoyará para indexar los enlaces para cumplimentar con el estándar HATEOAS.
- **Controllers:** clases para controlar las funciones de autenticación y alta en el *back-end*.
- **Exception:** clases que extienden “RuntimeException” para el lanzamiento de errores personalizados.
- **Payload:** paquete donde se definen las clases para la recepción y respuesta de los métodos de autenticación y alta.
- **Security:** paquete donde se ubica la configuración y los servicios necesarios implementar la seguridad del sistema.
- **Utilis:** paquete donde se almacena una clase para obtener y enviar mensajes como objetos de una clase.

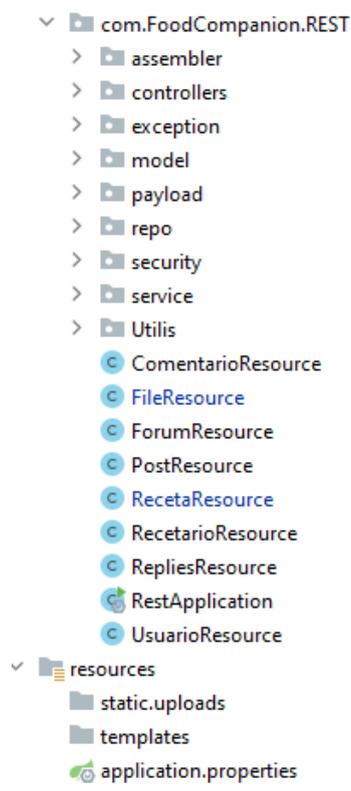


Figura 6.2: Estructura general del *back-end*

Módulos Spring

Las dependencias que se van a incluir en el servidor son las siguientes:

- **Spring JPA⁵**: módulo que provee de los métodos necesarios para la implementación de repositorios basados en JPA. JPA conocido por sus siglas *Java Persistence API* permite manejar datos relacionales en aplicaciones basadas en Java. JPA ofrece una serie de anotaciones para declarar objetos Java como entidades relacionales y métodos para realizar las tareas CRUD mas rutinarias, además de poder elaborar métodos personalizados, los cuales se adaptan a sentencias SQL basándose únicamente en la declaración de dichos métodos, como se observa en la figura 6.3 .

```
List<Replies> findRepliesFromPostById(Long id);
```

Figura 6.3: Creación de un método personalizado en JPA.

- **Spring HATEOAS⁶**: permite indexar enlaces a las respuestas JSON permitiendo cumplir con el estándar HATEOAS. El estándar HATEOAS plantea que parte de los objetos relacionados con un modelo serán representados en forma de hipervínculos, esto permite al cliente aligerar es uso del API, facilitando su uso.
- **MYSQL** dependencia que permite la conexión de *Spring* al servidor MYSQL.
- **Lombok⁷** ayuda a mantener el código mas limpio, evitando la constante creación de *getters* y *setters* en la definición de los modelos así como de las variables privadas y su inicialización en los constructores.
- **Spring Starter Web**: permite configurar el servidor *Spring* para incluir los servicios RESTful necesarios. Utiliza Tomcat como servidor web por defecto.
- **Spring Starter Security** dependencia que permite definir directivas de seguridad de acceso a los métodos expuestos por por los controladores.
- **JsonWebToken**: generación de *tokens* de acceso que permite securizar las comunicaciones entre cliente y servidor.

Configuración de la seguridad

En primer lugar para securizar el acceso al servidor se han de proveer una serie de rutas a las cuales cualquier cliente pueda acceder y que rutas a los métodos están protegidas, de este modo en el servidor se configura la clase “WebSecurityConfig” que extiende la clase “WebSecurityConfigurerAdapter” donde se sobrescriben una serie de métodos [3]:

- El método “configure(HttpSecurity http)” que configura el CORS y el CSRF, además de especificar si los usuarios deben estar autenticados o no, que filtro aplicar y cual es el manejador de excepciones(figura 6.4), se puede observar que una vez autenticado, se tiene acceso a cualquier ruta del servidor:
- El método “configure(WebSecurity web)” que habilita las rutas accesibles por cualquier cliente , sin tener la necesidad de estar autenticado (figura 6.5):

⁵<https://spring.io/projects/spring-data-jpa>

⁶<https://spring.io/projects/spring-hateoas>

⁷<https://www.baeldung.com/intro-to-project-lombok>

```

@Override
protected void configure(HttpSecurity http) throws Exception {
    http.cors().and().csrf().disable()
        .exceptionHandling().authenticationEntryPoint(unauthorizedHandler).and()
        .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS).and()
        .authorizeRequests().antMatchers("/**").permitAll()
        .antMatchers("/").permitAll()
        .anyRequest().authenticated();

    http.addFilterBefore(authenticationJwtTokenFilter(), UsernamePasswordAuthenticationFilter.class);
}

```

Figura 6.4: Configuración de seguridad

```

@Override
public void configure(WebSecurity web) throws Exception {
    web
        .ignoring().antMatchers("/receta/**", "/api/auth/**", "/forum/**", "/posts/**");
}

```

Figura 6.5: Configuración de rutas accesibles por cualquier cliente

6.2 Front-End

En esta sección se detallan las tecnologías empleadas para la implementación del *front-end*. En primer lugar se van a indicar que tecnologías se han empleado y el porqué, seguido de las librerías que se incluyen en el proyecto web.

6.2.1. Tecnologías

Para la implementación del *front-end* se ha escogido Angular como *framework* de desarrollo web. Angular es una plataforma de desarrollo web para crear aplicaciones basadas en *TypeScript*⁸, un lenguaje de desarrollo que extiende las funcionalidades de *JavaScript*⁹, añadiendo tipos estáticos y objetos basado en clases.

6.2.2. Elección de Angular

Se ha escogido Angular como *framework* de desarrollo web debido a las siguientes razones:

- **Ahorro en tiempos de desarrollo:** Angular está organizado de tal manera que permite ahorrar tiempo a la hora de organizar la aplicación. La definición de módulos, componentes y servicios está claramente definida, ayudando así al programador a no ocupar su tiempo en tareas de organización.
- **Componentes web:** Angular divide sus aplicaciones en componentes, piezas individuales de una misma aplicación. Los componentes son reutilizables en cualquier sección de la aplicación web, además ayuda a dividir y mantener ordenado las piezas que componen la totalidad de la aplicación web.
- **Gran soporte y herramientas:** gracias a la gran comunidad de programadores que utilizan Angular, se pueden encontrar solución a cualquier problema que surja a lo largo del desarrollo, además de una muy extensa documentación tanto como para Angular, TypeScript y HTML.

⁸<https://www.typescriptlang.org/>

⁹<https://developer.mozilla.org/es/docs/Web/JavaScript>

6.2.3. Implementación del *front-end*

Angular estructura de manera muy sencilla sus aplicaciones, facilitando la tarea del programador y así aligerando la carga de trabajo.

En primer lugar Angular define sus módulos en el fichero “app.module.ts”, donde cada módulo es importado para su uso en todo el proyecto.

En la raíz del proyecto Angular define su aplicación base, en la cual se pueden incluir componentes o código HTML de forma directa. Estos ficheros son “app.component.ts”, “app.component.html” y “app.component.css” que se describen a continuación:

- En el fichero “**app.component.html**” se define la aplicación base combinando HTML y las etiquetas de cada componente para la creación de aplicaciones modulares.
- El fichero “**app.component.ts**” define la lógica del componente, en la cual se definen datos, métodos y llamadas a servicios definidos en Angular donde se realizarán peticiones a la interfaz que definida en el *back-end*.
- Por último el fichero “**app.component.css**” que define los estilos asociados a dicho componente HTML.

Esta primera estructura define la aplicación web principal, pero gracias a los componentes se pueden definir piezas individuales que se pueden añadir en esta primera estructura creada por defecto por Angular.

Cada componente está definido por los tres ficheros detallados anteriormente, donde se construyen piezas individuales de contenido web y se pueden reutilizar en otros componentes.

Definición de los módulos

A continuación se especifican los módulos a incluir en el proyecto de Angular para la creación de la aplicación web:

- **BrowserModule**¹⁰: importa la infraestructura requerida para todas las aplicaciones en Angular.
- **AppRoutingModule**¹¹: permite adherir navegación a la aplicación, definiendo diferentes rutas en el fichero “app-routing.module.ts”, donde se importan los componentes y se definen las rutas al componente deseado.
- **HttpClientModule**¹²: módulo gracias al cual se pueden implementar operaciones HTTP de consulta al *back-end*.
- **FormsModule**¹³: permite controlar los formularios y su validación.
- **ReactiveFormsModule**¹⁴: importa la infraestructura requerida para el uso de directivas “*reactive forms*” para su uso en formularios y validaciones.
- **AutocompleteLibModule**¹⁵: módulo que permite implementar un buscador según una estructura de datos definida.

¹⁰<https://angular.io/api/platform-browser/BrowserModule>

¹¹<https://angular.io/tutorial/toh-pt5>

¹²<https://angular.io/api/common/http/HttpClientModule>

¹³<https://angular.io/api/forms/FormsModule>

¹⁴<https://angular.io/api/forms/ReactiveFormsModule>

¹⁵<https://www.npmjs.com/package/angular-ng-autocomplete>

Además de los módulos descritos en esta sección, la página web hará uso del *framework* de desarrollo de interfaces web **Bootstrap**¹⁶ y el conjunto de iconos que tiene disponible **Bootstrap Icons**. Para el uso de **Bootstrap** se han de importar los componentes necesarios para su correcto funcionamiento en el fichero “angular.json” (figura 6.6).

```

"assets": [
  "src/favicon.ico",
  "src/assets"
],
"styles": [
  "src/styles.css",
  "node_modules/bootstrap/dist/css/bootstrap.css"
],
"scripts": [
  "node_modules/bootstrap/dist/js/bootstrap.bundle.js",
  "node_modules/jquery/dist/jquery.js"
]

```

Figura 6.6: Componentes Bootstrap importados al proyecto web

Estructura de la aplicación:

Se ha definido la estructura básica de los ficheros en los apartados anteriores, ahora se define la estructura principal de la aplicación explicando su definición y como se interrelacionan los componentes entre si.

La aplicación web esta dividida en tres partes bien diferenciadas entre si y definidas en el fichero “app.component.html”:

```

1 <app-header></app-header>
2 <!--Contenido web-->
3 <router-outlet></router-outlet>
4 <app-footer></app-footer>

```

Figura 6.7: Estructura de la página web

Como se observa en la figura 6.7 se observa una estructura basada en tres partes, un *header* donde se ubican enlaces de acceso a las diferentes páginas dentro de la aplicación y un buscador de recetas.

Luego se encuentra la directiva “<router-outlet></router-outlet>”, que define donde se van a ubicar los elementos definidos en el modulo “app-routing.module.ts”, componente que cambia según la página a la que se accede y por último un *footer* donde se define un enlace a la página principal de la aplicación web.

Cada componente se ubica en la carpeta “components” donde se generarán mediante el siguiente comando, ubicándose en la raíz del proyecto:

```
ng g component components/nombre_del_componente
```

Dentro de dicha carpeta se encuentran los componentes que forman parte de la aplicación web, que se verifican en la sección de pruebas de este documento (figura 6.8).

Además en la raíz del proyecto se incluye una carpeta denominada “interfaces” (figura 6.9) donde se almacenan interfaces similares a las clases definidas como “modelos” en el *back-end* para cuando se hace una llamada a los servicios ofrecidos por el servidor, Angular maneja una estructura de datos similar para homogeneizar el manejo y manipulación de los datos.

¹⁶<https://getbootstrap.com/>

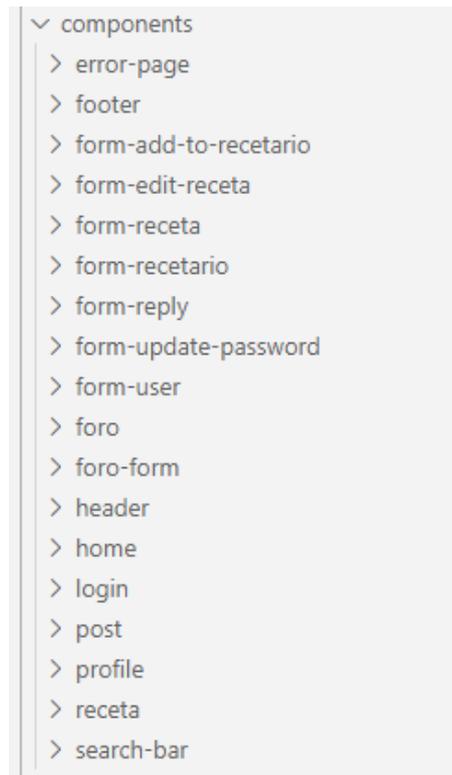


Figura 6.8: Estructura de los componentes

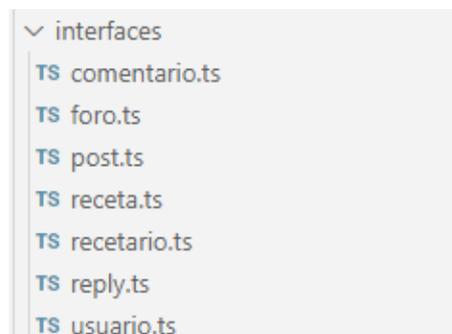


Figura 6.9: Modelos definidos en Angular

Por último se definen dos carpetas más en la raíz del proyecto [5], “_services” y la carpeta “_helpers”, la primera contiene los servicios que acceden a la interfaz definida en el *back-end* para realizar peticiones HTTP y almacenar el *token* de sesión en el navegador, y la carpeta “_helpers” contiene una función para indexar dicho token en cada petición a al servidor para autenticar al usuario y poder acceder a las funciones de la interfaz HTTP (figura 6.10).

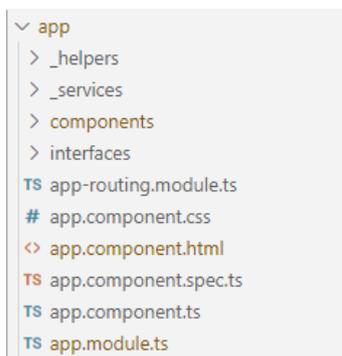


Figura 6.10: Carpeta raíz de la aplicación

6.3 Herramientas

En las secciones anterior se detallan las tecnologías que se empelan para el desarrollo de la aplicación web tanto para el *back-end* como para el *front-end*, en esta sección se especificarán las herramientas necesarias para manipular dichas tecnologías.

- Para el entorno de desarrollo del *back-end* se va a emplear el IDE de desarrollo **IntelliJ IDEA**¹⁷, desarrollado por JetBrains, aporta las herramientas necesarias para el desarrollo del API, haciendo mención especial a su integración con el *Spring*.
- Como entorno de pruebas para el *back-end* se emplea la herramienta **Postman**¹⁸, en la cual se realizarán peticiones HTTP al *back-end* de forma muy sencilla e intuitiva.
- Para el desarrollo del *front-end* se ha empleado **Visual Studio Code**¹⁹. Un entorno de desarrollo ligero y fácil de usar.
- Como cliente web se empleará el navegador web **Mozilla Firefox**²⁰, un navegador muy completo y compatible con muchas de las funciones *TypeScript* que se van a implementar.

¹⁷<https://www.jetbrains.com/es-es/idea/>

¹⁸<https://www.postman.com/>

¹⁹<https://code.visualstudio.com/>

²⁰<https://www.mozilla.org/es-ES/>

CAPÍTULO 7

Pruebas

Una vez implementada la aplicación web se verifican que los caso de uso se satisfacen de manera correcta, para ello se ejecuta una serie de pruebas a la aplicación para verificar que se cumplen los casos de uso planteados en la sección de “Análisis”.

7.1 Registro

Un formulario de registro sencillo permite al usuario dar de alta una cuenta en el sistema, como se indica en la figura 7.1, el formulario cuenta con un nombre de usuario, el e-mail y una contraseña. El sistema comprueba que no existe un usuario con el mismo nombre o e-mail y registra al usuario:

Crear una cuenta de usuario

Nombre de usuario

Contraseña

E-mail

[Crear cuenta](#)

¿Ya tienes una cuenta? [Iniciar sesión](#)

Figura 7.1: Registro

7.2 Inicio de sesión

En la figura 7.2 se representa la sección para iniciar sesión en el sistema, consta de un formulario donde se introduce un nombre de usuario y la contraseña:

Iniciar sesión

Nombre de usuario

Contraseña

¿No tienes una cuenta? [Crear una cuenta](#)

Figura 7.2: Inicio de sesión

7.2.1. Fallo en el inicio de sesión

En el caso de iniciar sesión con datos no registrados en el sistema, el servidor notifica el error al usuario (figura 7.3):

Iniciar sesión

Nombre de usuario

Contraseña

Login failed: Error usuario o la contraseña
erroneos

¿No tienes una cuenta? [Crear una cuenta](#)

Figura 7.3: Fallo en el inicio de sesión

Una vez iniciada la sesión, la página de gestión es la que representa la figura 7.4:

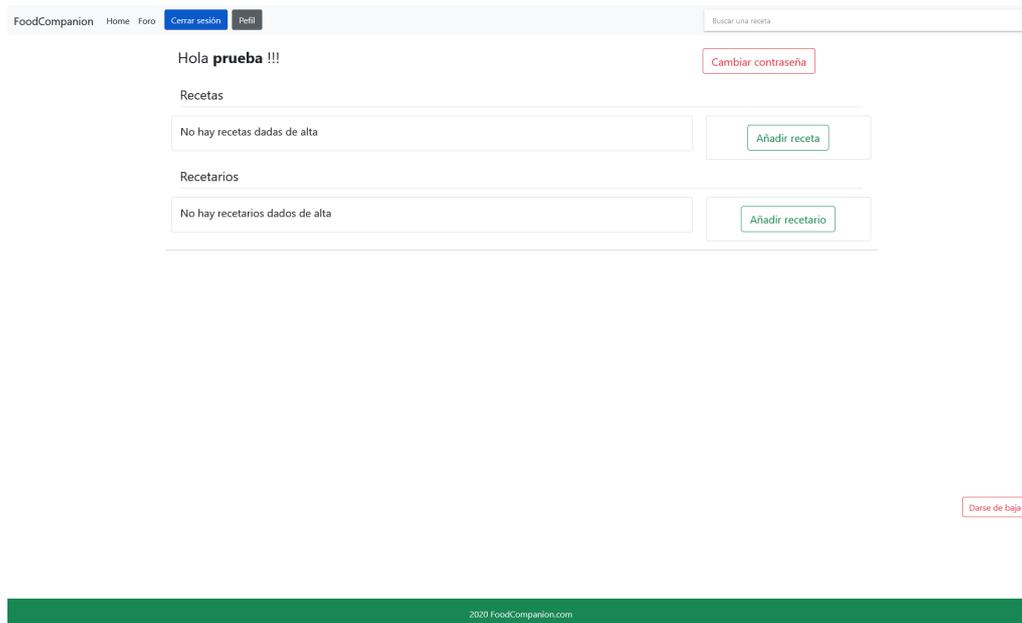


Figura 7.4: Perfil de usuario

La figura 7.4 refleja las tres partes en las cuales se divide la aplicación web, el *header*, con enlaces a la página principal de la aplicación, al foro y botones para cerrar sesión y acceso al perfil, el cuerpo de la página que cambia según el componente al que se accede y por último un *footer* sencillo con un enlace a la página principal.

7.3 Cerrar sesión

La figura 7.5 representa la acción de cerrar sesión donde se selecciona el botón de “Cerrar sesión” ubicado en el *header* de la aplicación web y se despliega una alerta para confirmar la acción :

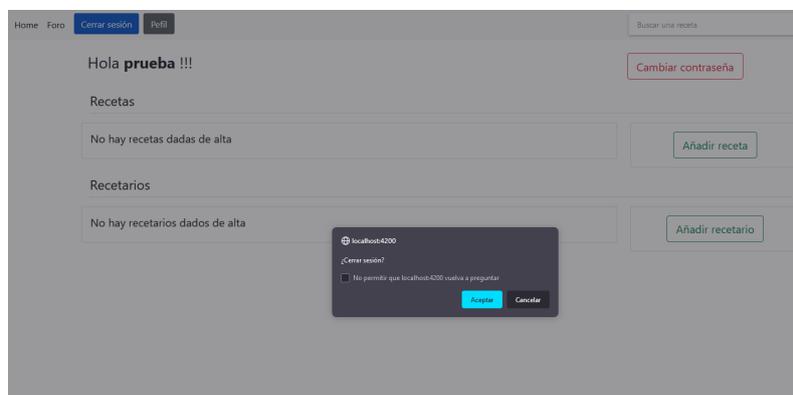
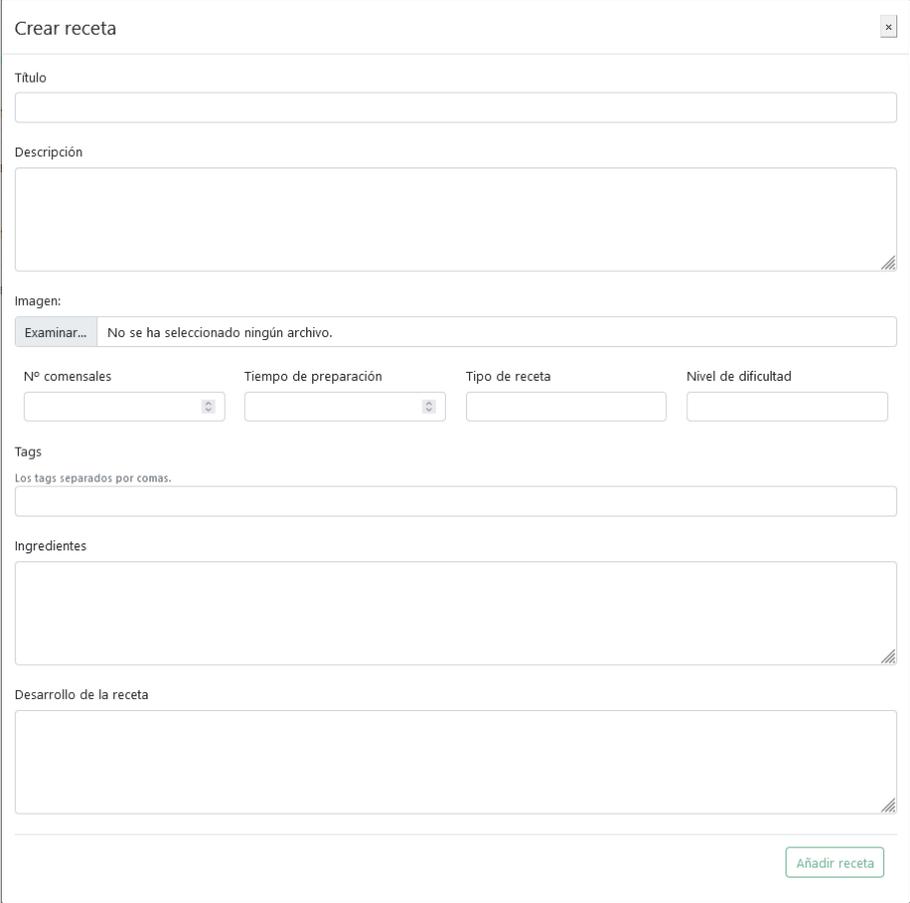


Figura 7.5: Cerrar sesión

7.4 Alta de una receta

Para dar de alta una receta en la página de gestión del usuario se ubica un botón en la parte superior derecha, seleccionando el botón la aplicación despliega un formulario con los campos necesarios para dar de alta una receta:



Crear receta

Título

Descripción

Imagen:

Examinar... No se ha seleccionado ningún archivo.

Nº comensales

Tiempo de preparación

Tipo de receta

Nivel de dificultad

Tags

Los tags separados por comas.

Ingredientes

Desarrollo de la receta

Añadir receta

Figura 7.6: Alta de una receta

Como se observa en la figura 7.6 el formulario de alta consta de varias secciones, el título de la receta, una breve descripción, una imagen de la receta una vez elaborada, el número de comensales, el tiempo de preparación, el tipo de receta, el nivel de dificultad de la receta, las etiquetas que identifican a la receta, los ingredientes que la componen y por último el desarrollo completo de la receta.

Se toma de ejemplo una receta de “Pasta pesto” par dar de alta una receta y observar el cambio en la página de gestión de recetas, la figura 7.7 representa la receta añadida al sistema de gestión:

Hola **prueba** !!! Cambiar contraseña

Recetas

Lista de la recetas publicadas por prueba

Titulo	Tipo	Acciones
Pasta Pesto	Principal	Editar 🗑️ 👁️

< 1 >

Añadir receta

Recetarios

No hay recetarios dados de alta

Añadir recetario

Figura 7.7: Receta añadida a la página de gestión

7.5 Editar una receta

Para editar una receta el usuario selecciona el botón “Editar” del sistema de gestión de recetas y se despliega un editor con los campos de la receta ya cargados para poder ser editados (figura 7.8):

Editar receta

Título
Pasta Pesto

Descripción
El pesto es una salsa italiana de origen genovés que lleva albahaca, queso, piñones, ajo y aceite de oliva. Unos ingredientes sencillos que la convierten en una salsa fresca y ligera a la que pocas personas pueden resistirse.

Imagen:
Examinar... No se ha seleccionado ningún archivo.

Nº comensales: 2 Tiempo de preparación: 20 Tipo de receta: Principal Nivel de dificultad: Medio

Tags
Los tags separados por comas.
pesto, albahaca, Pasta

Ingredientes
60 gr. de albahaca fresca (las hojas más tiernas)
1 diente de ajo (o menos)
50 gr. de queso parmesano
50 gr. de queso pecorino
50 gr. de piñones

Desarrollo de la receta
Hervimos la pasta que queramos siguiendo las indicaciones del fabricante.
Empezamos a preparar el pesto. Es importante usar las hojas más tiernas de la albahaca, las más pequeñas. Las lavamos y quitamos los tallos. La cantidad de ajo va al gusto, podemos usar sólo un trocito o usar un ajo entero si preferimos un sabor más potente.

Editar receta

Figura 7.8: Editar receta

7.6 Borrar una receta

En el caso de que el usuario quiera borrar una receta, el sistema proporciona un botón con un icono de un basurero para borrar la receta, la figura 7.9 representa la acción de borrado desplegando una alerta de confirmación:

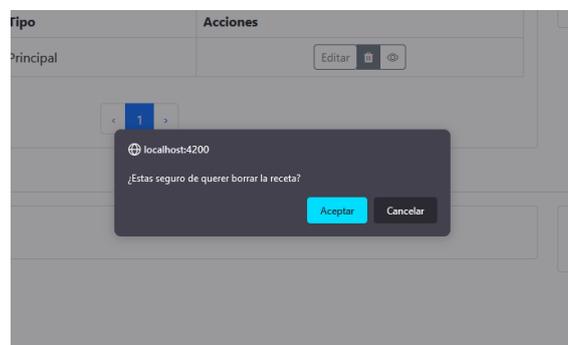


Figura 7.9: Borrar receta

7.7 Visualizar una receta

Para acceder a visualizar la receta el usuario selecciona el botón con el icono de un ojo y es redirigido a la receta:

FoodCompanion Home Foro [Cerrar sesión](#) [Perfil](#)

Pasta Pesto

Guardar receta
Compartir receta



2

20

Medio

Principal

pesto
albahaca
Pasta

Ingredientes

- 60 gr. de albahaca fresca (las hojas más tiernas)
- 1 diente de ajo (o menos)
- 50 gr. de queso parmesano
- 50 gr. de queso pecorino
- 50 gr. de piñones
- Una cucharadita de sal gruesa
- 100-120 ml. de aceite de oliva virgen extra
- Un poco de agua de cocción (de la pasta)

Descripción

El pesto es una salsa italiana de origen genovés que lleva albahaca, queso, piñones, ajo y aceite de oliva. Unos ingredientes sencillos que la convierten en una salsa fresca y ligera a la que pocas personas pueden resistirse.

Instrucciones de la receta

Hervimos la pasta que queramos siguiendo las indicaciones del fabricante.

Empezamos a preparar el pesto. Es importante usar las hojas más tiernas de la albahaca, las más pequeñas. Las lavamos y quitamos los tallos. La cantidad de ajo va al gusto, podemos usar sólo un trocito o usar un ajo entero si preferimos un sabor más potente.

Tradicionalmente la salsa se prepara con mortero y energía, no con robot de cocina (pero no se lo diremos a nadie si usáis una batidora o robot).

Con mortero o robot de cocina vamos triturando todos los ingredientes. Manualmente se suele picar primero el ajo y después se va añadiendo el resto. El color tiene que ser un verde brillante. No tenemos que pasarnos con la cantidad de queso y fijarnos en el color es una buena manera de encontrar el punto justo.

Si después de triturar el pesto nos ha quedado un poco espeso, podemos añadir una cucharada o dos del agua de cocción de la pasta.

Según los genoveses hay que servir el pesto en el fondo del plato y echar la pasta encima, no al revés como solemos hacer. Removemos hasta que la pasta y la salsa queden bien integradas, y a comer.

Comentarios

El usuario prueba comenta:
2021-08-31
Ésta receta es genial!!!!!!!

Escribir un comentario

Publicar comentario

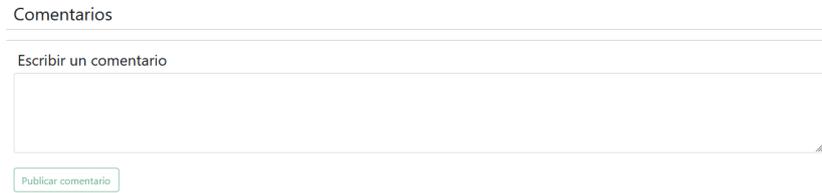
2020 FoodCompanion.com

Figura 7.10: Visualizar receta

En la figura 7.10 se puede observar la receta, la imagen de presentación, los atributos de la receta, el numero de comensales, el tiempo de cocción, el nivel de dificultad y el tipo de receta. Luego la receta se divide en tres partes, los ingredientes, la descripción y por último las instrucciones de la receta.

7.8 Comentar una receta

Para comentar una receta, como se observa en la imagen 7.11, al final de la receta se encuentra una sección para añadir comentarios:



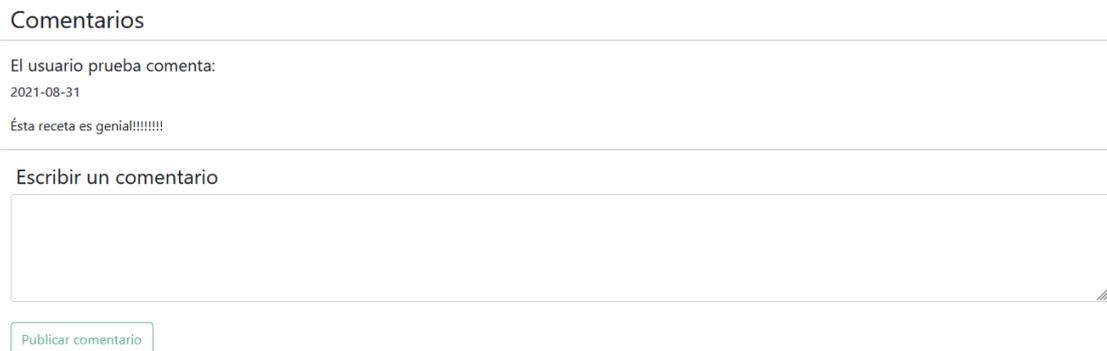
Comentarios

Escribir un comentario

Publicar comentario

Figura 7.11: Comentar receta

En este caso, para comentar una receta el usuario tiene que estar dado de alta en el sistema y con una cuenta iniciada (figura 7.12). Una vez escrito un comentario, este se puede visualizar en la parte superior de la sección:



Comentarios

El usuario prueba comenta:
2021-08-31
Ésta receta es genial!!!!!!!

Escribir un comentario

Publicar comentario

Figura 7.12: Comentario en una receta

7.9 Compartir una receta

Para compartir la receta vía enlace, el usuario dispone de un botón para obtener el enlace a la receta con un solo *click*.

Para ello en la sección de la receta, junto con el botón para guardar recetas, el usuario dispone de un botón denominado “Compartir receta”, la figura 7.13 representa la cabecera del la receta:

Pasta Pesto

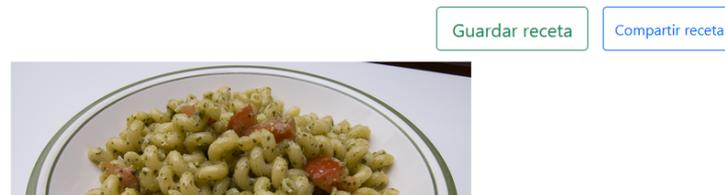


Figura 7.13: Botón para compartir una receta.

De seleccionar el botón, se despliega un “popover”¹ a la derecha del botón con el enlace generado (figura 7.14).

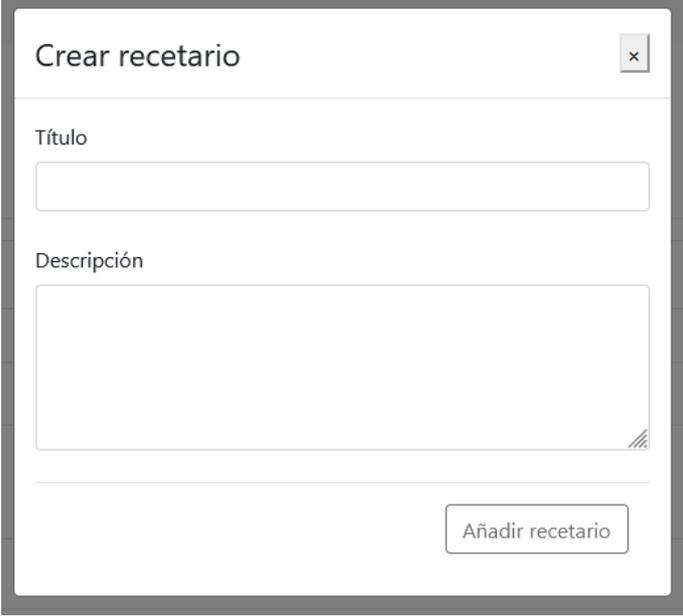


Figura 7.14: Enlace generado para compartir la receta

¹<https://getbootstrap.com/docs/4.0/components/popovers/>

7.10 Alta de un recetario

En la sección de gestión, el usuario puede dar de alta un recetario para guardar diferentes recetas, seleccionado el botón de “Añadir recetario”, la imagen 7.15 representa el formulario de alta de un recetario:



Crear recetario

Título

Descripción

Añadir recetario

Figura 7.15: Alta recetario

Luego de crear el recetario se puede observar que esta ha sido creado en la sección indicada, en la figura 7.16 se observa el recetario y un botón para borrarlo:

Hola **prueba** !!! [Cambiar contraseña](#)

Recetas

Lista de la recetas publicadas por prueba

Titulo	Tipo	Acciones
Pasta Pesto	Principal	Editar Borrar Compartir

< 1 >

[Añadir receta](#)

Recetarios

Titulo: Platos de pasta | Descripción: Para cenas [^](#)

[Borrar recetario](#) [Añadir recetario](#)

Figura 7.16: Recetario añadido

7.11 Borrar un recetario

Para borrar un recetario, el usuario accede al recetario en cuestión y selecciona el botón “Borrar recetario”, el sistema borra el recetario del usuario eliminando el recetario de la sección de gestión de recetarios. La figura 7.17 representa una alerta de confirmación de borrado del recetario:

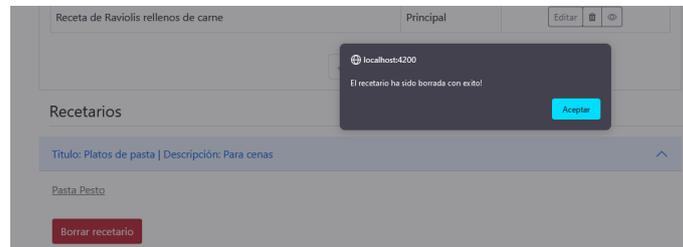


Figura 7.17: Eliminar recetario

7.12 Añadir una receta al recetario

En el caso de querer añadir una receta a cualquier recetario dado de alta por un usuario, en la receta en cuestión se selecciona el botón “Guardar receta”, que abre un formulario con los recetarios del usuario, la figura 7.18 representa el formulario para añadir una receta al recetario, indicando que recetarios el usuario tiene dados de alta en el sistema.



Figura 7.18: Añadir una receta al recetario

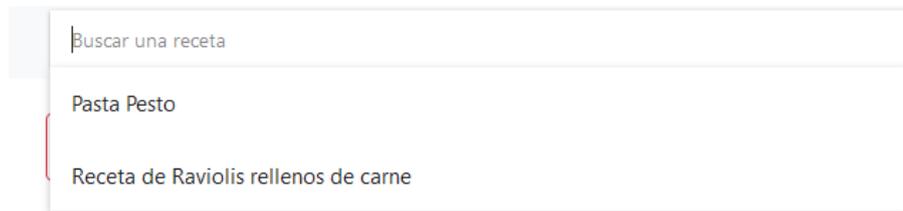
Añadida la receta al recetario, si se vuelve a la página de gestión de recetas se observa en la figura 7.19 que el recetario contiene la receta como enlace para acceder a ella:



Figura 7.19: Receta añadida añadida al recetario

7.13 Búsqueda de una receta

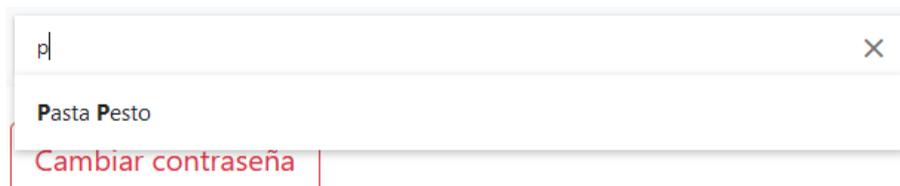
El buscador situado en el *header* de la página permite buscar cualquier receta según el nombre de la receta, la figura 7.20 representa el buscado con dos recetas dadas de alta:



A search bar with the placeholder text "Buscar una receta". Below the search bar, a dropdown menu is open, displaying two search results: "Pasta Pesto" and "Receta de Raviolis rellenos de carne".

Figura 7.20: Búsqueda

Como se puede observar en la figura 7.21, el usuario teclea a la receta a buscar y el buscador filtra los resultados:



A search bar with the text "p" entered. Below the search bar, a dropdown menu is open, displaying two search results: "Pasta Pesto" and "Cambiar contraseña".

Figura 7.21: Filtro

Si se selecciona la receta, dirige al usuario a la receta en cuestión.

7.14 Alta de una tema en el foro

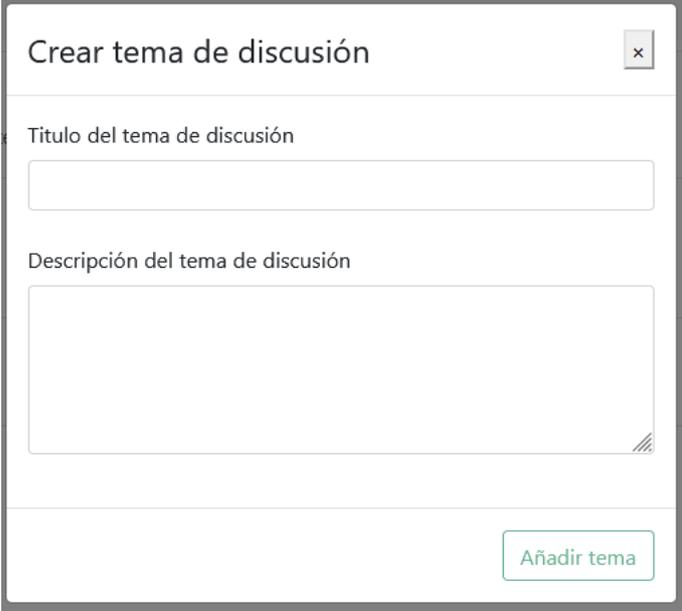
En la sección del foro se puede observar una sección para dar de alta un tema en el foro, la figura 7.22 representa el foro, con un botón para añadir temas:



A forum page titled "Foro FoodCompanion". Below the title, there is a button labeled "Crear tema". Below the button, there is a message: "No existen temas en el foro!!!! Da de alta un tema para poder empezar."

Figura 7.22: Foro

Si el usuario selecciona el botón "Crear tema", se despliega formulario con los campos necesarios para dar de alta el tema en el foro, la figura 7.23 representa el propio formulario con las secciones necesarias para crear el tema: un título y la descripción del tema:



Crear tema de discusión

Título del tema de discusión

Descripción del tema de discusión

Añadir tema

Figura 7.23: Añadir tema al foro

Una vez completado el formulario y enviado, el servidor registra el tema en el foro, la imagen 7.24 representa el tema creado en el foro y diversa información acerca de el tema, como el usuario, la fecha de creación y el título:



Foro FoodCompanion

En el foro de FoodCompanion puedes discutir de muchos temas relacionados con el mundo de las recetas!!!

Crear tema

La pasta esta buenísima!!!!

Creado por : prueba

Fecha de creación: 31 ago. 2021 23:20:12

Figura 7.24: Tema añadido al foro

7.15 Responder un tema en el foro

Para responder un tema creado en foro, el usuario accede al tema que quiera responder y selecciona el botón “replies”:

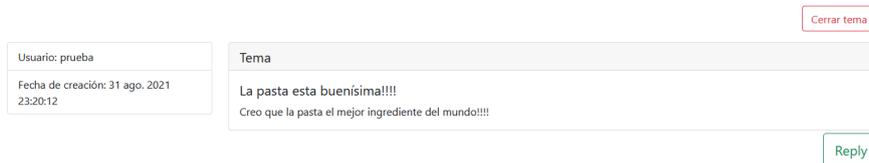


Figura 7.25: Tema de discusión

Como se observa en la figura 7.25, un tema en el foro consta de tres partes, la cabecera del foro, con los datos del creador y la fecha de creación del tema, el título del tema y por último el cuerpo del tema.

Además se disponen de dos botones, uno solo accesible por el creador del tema, el botón “cerrar tema” para cerrar el tema de conversación del foro, y un botón de “reply” para responder a dicho tema.

Si el usuario desea responder al tema seleccionado el botón de respuesta se despliega un formulario con un campo para subir la respuesta, se observa en la figura 7.26 el formulario con un único campo para responder el tema:

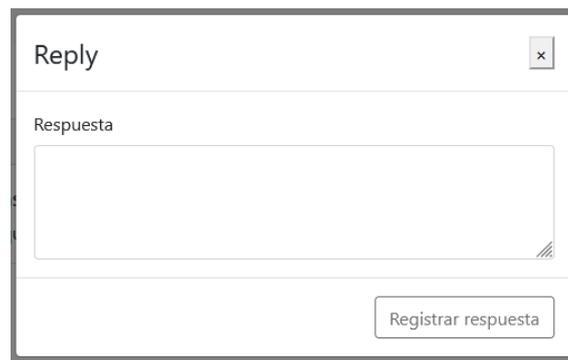


Figura 7.26: Formulario de respuesta a un tema

El usuario escribe su respuesta y el servidor la registra, junto con el nombre del usuario y la fecha de la respuesta. Las respuestas no pueden ser editadas, al igual que los comentarios, la figura 7.27 representa la respuesta registrada en el tema:



Figura 7.27: Respuesta subida al foro

7.16 Cambiar contraseña

Para cambiar la contraseña el usuario dispone de botón que despliega un formulario para cambiar su contraseña, la figura 7.28 representa la cabecera de la página de gestión junto con el botón par cambiar la contraseña:



Figura 7.28: Botón para cambiar la contraseña

El formulario pide al usuario su contraseña actual, y la nueva contraseña dos veces. En caso de no coincidir las contraseñas, el botón para enviar el formulario no se activa, por otro lado en caso de fallar la contraseña actual, el sistema notifica al usuario del fallo, la figura 7.29 representa un intento de cambio de contraseña con fallo a la hora de indicar la contraseña actual:

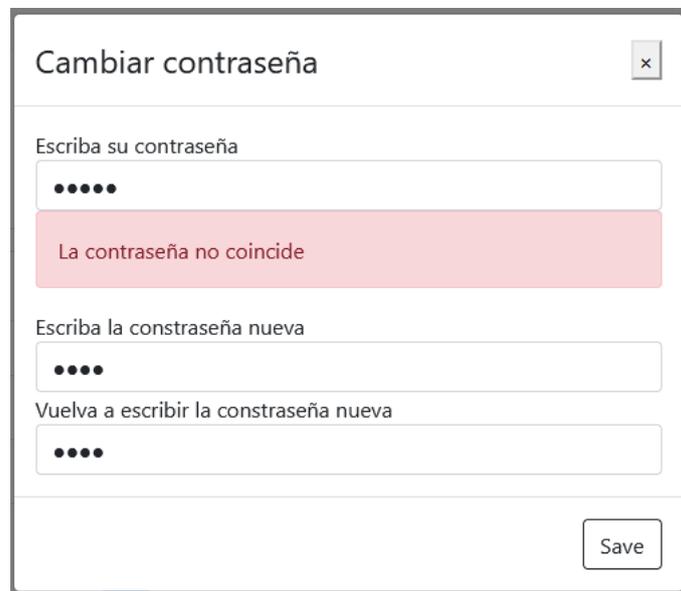
A screenshot of a modal window titled "Cambiar contraseña" with a close button (X) in the top right corner. The form contains three input fields: "Escriba su contraseña" (with 5 dots), "Escriba la contraseña nueva" (with 4 dots), and "Vuelva a escribir la contraseña nueva" (with 4 dots). A red error message "La contraseña no coincide" is displayed between the first and second input fields. A "Save" button is located at the bottom right of the form.

Figura 7.29: Error en el cambio de la contraseña

7.17 Dar de baja una cuenta de usuario

Para que el usuario pueda darse de baja de la aplicación, al final de la página de gestión del usuario, se encuentra un botón para darse de baja, que despliega una ventana para confirmar la baja de la aplicación (figura 7.30) :

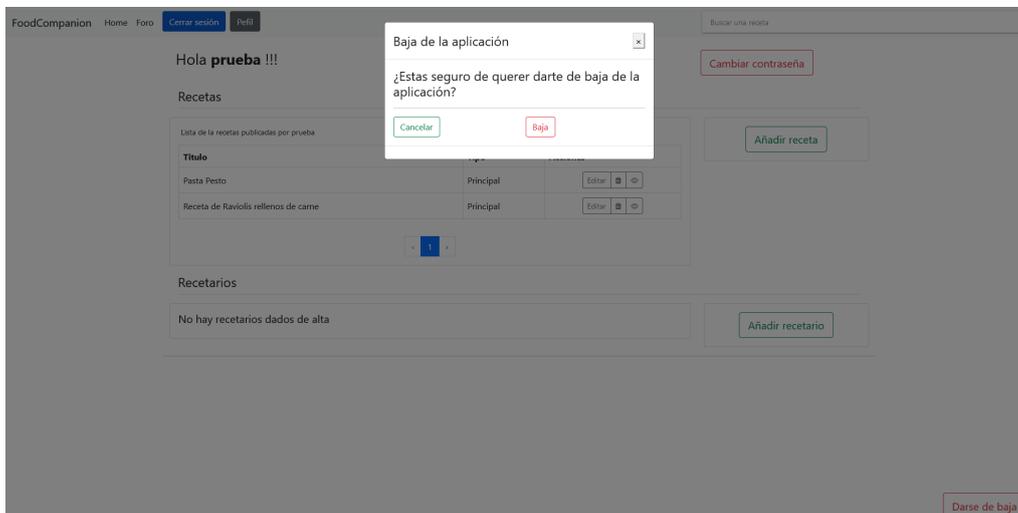


Figura 7.30: Baja de la aplicación

La baja procesa los datos del usuario y borra las recetas, pero los temas de discusión, respuestas y comentarios en las recetas se quedan guardados.

7.18 Pruebas para el usuario anónimo

La aplicación permite su uso a usuarios anónimos. El usuario anónimo tiene acceso a funciones básicas del sistema, como dar de alta una cuenta de usuario e iniciar sesión, el visualizado de recetas y por último acceso al foro, sus temas de discusión y respuestas. El usuario anónimo solo puede visualizar el contenido de las recetas y el foro, tiene restringido el uso de la sección de comentarios y poder responder temas de discusión.

7.18.1. Página principal

La página principal de la aplicación consta de del mismo *header* y *footer* que en las secciones anteriores, en el *header* se encuentra un botón para iniciar sesión y el buscador. Esta página es accesible tanto por un usuario anónimo como un usuario con una sesión iniciada.

Como se puede observar en la figura 7.31 se visualizan una serie de cartas con la imagen de la receta, sus características, como el numero de comensales, el tiempo de preparación, la dificultad y el tipo de plato y por último su descripción.

7.18.2. Visualización de la receta y comentarios

Si el usuario accede a la página de la receta, se observa en la figura 7.32 que el botón para guardar una receta no se encuentra disponible:

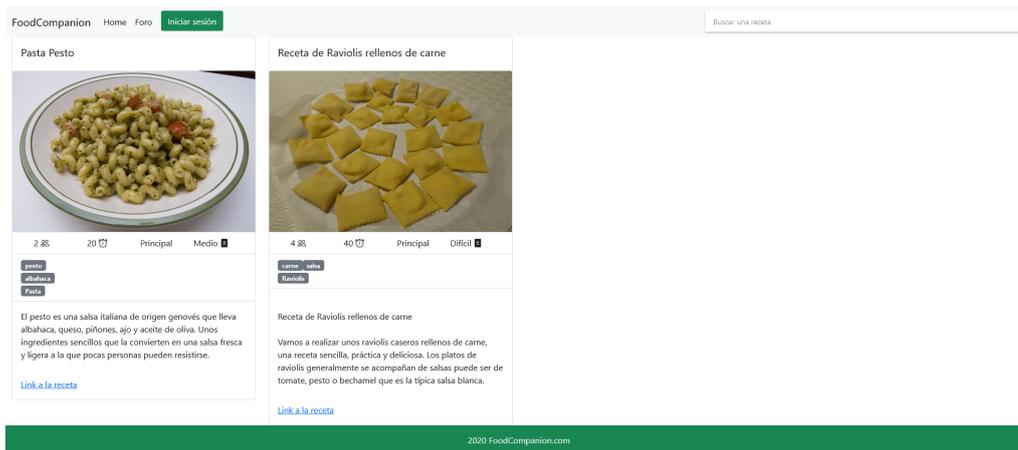


Figura 7.31: Página principal de la aplicación

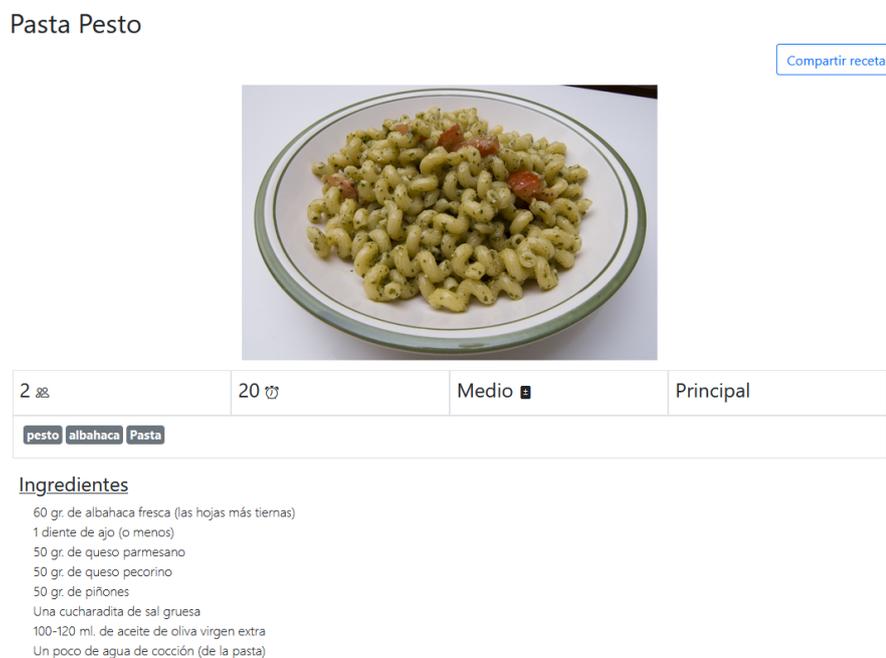


Figura 7.32: Cebecera de a receta visualizada por un usuario anónimo

Además en la la sección de comentarios, en el cuadro del formulario para comentar la receta, el usuario no puede escribir y el texto indica que si se quiere comentar la receta, el usuario debe dar se de de alta o iniciar sesión sesión (figura 7.33).

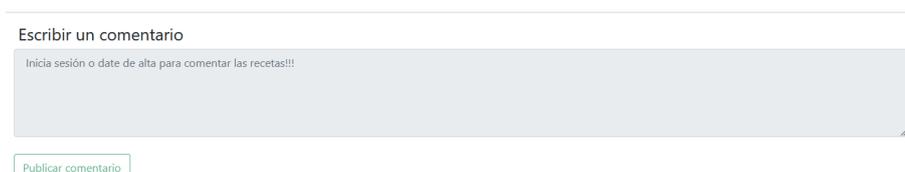


Figura 7.33: Comentarios restringidos al usuario anónimo

7.18.3. Foro y respuestas

En el caso del foro el usuario anónimo observa que la aplicación restringe el uso del botón para crear tema e insiste en darse de alta o iniciar sesión como lo indica la figura 7.34



Figura 7.34: Visualización del foro siendo un usuario anónimo

En el caso de acceder al tema en cuestión, el usuario anónimo visualiza el tema además de sus respuestas. Como en el caso anterior, el sistema insiste en el alta en la aplicación o el inicio de sesión para el uso del las respuestas en tema de discusión (figura 7.35).



Figura 7.35: Visualización del los temas siendo un usuario anónimo

CAPÍTULO 8

Presupuesto

A lo largo de este apartado se van a estimar los costes de la aplicación web, teniendo en cuenta diferentes aspectos, como el salario promedio de un ingeniero informático recién graduado, los costes de *hardware* para desarrollar la aplicación y el uso de las tecnologías y servicios.

Para el desarrollo de la aplicación web se han estimado una serie de costes que se detallan a continuación:

- El sueldo promedio de un ingeniero informático recién graduado ronda entre los 18.000€ a 30.000 euros en la categoría *junior*[6].
- El coste de consumo para el desarrollo del proyecto promedia 0,14€/kWh de uso, el precio promedio del consumo de un ordenador encendido[7], además del coste de un ordenador con una relación coste/prestaciones óptima para el desarrollo del proyecto ronda los 1000€[10].

En el caso del ordenador, se considera el coste prorrateado del equipo a lo largo del las semanas de desarrollo del proyecto, dejando un coste de 210,96 euros.

- Considerando las horas invertidas en el proyecto, 330 horas junto con las tutorías concertadas con el tutor, que rondan las 12 horas, teniendo en cuenta un sueldo promedio según el documento de “RETRIBUCIONES PROFESORADO FUNCIONARIO EN BASE A LA LEY DE PRESUPUESTOS PARA EL EJERCICIO 2021” ronda los 31.100 euros y 35.2000 euros[7].

- El *software* IntelliJ IDEA en su versión *Ultimate* tiene un coste de 149 euros/año¹, siendo esta la versión utilizada para el desarrollo gracias a la integración con *Spring Framework* que la versión gratuita no incluye.

Teniendo en cuenta que se hará uso de la herramienta a lo largo del desarrollo del proyecto se prorratea el coste a lo largo de las 11 semanas de desarrollo dejando un coste de 24,05 euros.

Las demás herramientas y tecnologías no tienen ningún coste adicional.

Según las estimaciones anteriores, el alumno cobraría 11,54 euros la hora y el profesor 19,90 euros la hora. La disposición del coste total esta detallado en la siguiente tabla:

¹<https://www.jetbrains.com/es-es/idea/buy/#personal?billing=yearly>

Recurso	Horas	Coste	
Alumno	330	11,54	3.808
Profesor	12	19,90	238,8
Cosumo	320	0,14	44,8
Equipo	-	-	210,96
Herramientas	-	-	24,05
Coste Total			3598,61

Tabla 8.1: Tabla de costes

El coste total del proyecto es de 3.598,61 euros.

CAPÍTULO 9

Conclusiones

A lo largo del presente documento se han especificado las fases del desarrollo de proyecto, una aplicación para la gestión de recetas, donde se puede añadir, borrar o visualizar las recetas añadidas, comentar las recetas, crear colecciones de recetas y por último participar en un foro de discusión.

Se han cumplido los diferentes objetivos de proyecto, implementando las funciones deseadas tanto en el servidor como en la interfaz de usuario. Además de los objetivos puramente relacionados con el proyecto, se han alcanzado una serie de objetivos personales que mejoran mis habilidades como programador.

9.1 Trabajo futuro

A lo largo del proceso de desarrollo de la aplicación se han tomado en cuenta una serie de mejoras que pueden ser implementadas en el futuro y así extender sus funcionalidades.

Las ideas principales a trabajar en un futuro son:

- Envío de correos electrónicos para la confirmación de alta y recuperación de contraseña.
- Una página para aplicar filtros, basándose en las características de la receta, como el numero de comensales ola dificultad e la receta.
- Implementación de un nuevo usuario, moderador, con acceso a las funcionalidades del foro y comentarios para su gestión y supervisión.
- Una función para compartir las recetas por redes sociales.
- Posibilidad de añadir una página de perfil de usuario para modificar sus datos y añadir foto de perfil.
- Nuevos formatos de receta, con la posibilidad de añadir un editor de textos con mas funcionalidades, como insertar imágenes o vídeos entre la descripción del texto.
- Página de gestión de foros, donde se darían de alta varios foros, cada uno con sus propios temas.

9.2 Valoración

Gracias a la habilidades adquiridas a lo largo este proyecto, como el uso de nuevas tecnologías de desarrollo herramientas y procesos, el desarrollo de este proyecto me aporta de una serie de habilidades que facilitarán mi inserción en el mercado laboral y mejoran sustancialmente mi rendimiento a la hora de elaborar mis tareas como trabajador.

Durante el desarrollo del proyecto han hecho falta muchas horas de comprensión de cada una de las herramientas y tecnologías de desarrollo. Gracias a esto he adquirido habilidades para la búsqueda rápida en la documentación y síntesis.

Además de la búsqueda en la documentación, mi habilidad como programador han mejorado sustancialmente gracias a uso de varios lenguajes de programación, HTML, CSS, TypeScript y Java, aprendiendo las peculiaridades de cada lenguaje.

Personalmente he aprendido a gestionar de manera mas eficiente mi tiempo, respetando tanto los tiempos de trabajo como los de descanso. He aprendido a superar las dificultades que aparecen a lo largo del desarrollo del proyecto, sean tanto técnicas como mentales y en general mejorando mis habilidades como programador y desarrollador de aplicaciones web.

Bibliografía

- [1] Especificación de requisitos según el estándar de IEEE 830, 22 de Octubre de 2008 <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>.
- [2] <https://modeloayuda.com/vista-controlador/> <https://modeloayuda.com/vista-controlador/>.
- [3] YOUTUBE, “Spring Boot Full Stack with Angular | Full Course [2021] [NEW]”, 5 de febrero de 2021 <https://www.youtube.com/watch?v=Gx4iBLKLVHk>.
- [4] Spring Boot Token based Authentication with Spring Security & JWT, 19 de agosto de 2021 <https://www.bezkoder.com/spring-boot-jwt-authentication/>.
- [5] Angular 12 Login and Registration example with JWT & Web Api, 8 de agosto de 2021
- [6] YOUTUBE “Ingeniería de Requisitos - Modelado Conceptual: Diagramas de Casos de Uso - Raquel Martínez España” <https://www.youtube.com/watch?v=eStxoMGVqm8>.
- [7] Qué hace un ingeniero informático o un graduado en informática, 15 de marzo de 2021 <https://www.epitech-it.es/ingeniero-informatico/>.
- [8] España, de los países más baratos para comprar PC, móviles Android y Smart TV, 17 de enero, 2017 <https://www.adslzone.net/2017/01/17/espana-los-paises-mas-baratos-comprar-pc-moviles-android-smart-tv/>.
- [9] “RETRIBUCIONES PROFESORADO FUNCIONARIO EN BASE A LA LEY DE PRESUPUESTOS PARA EL EJERCICIO 2021” http://www.upv.es/entidades/SRH/menu_urlv.html?/entidades/SRH/retribuciones/U0894376.pdf.
- [10] Los 5 Mejores portátiles para programadores, 22 de julio de 2021 <https://www.guiahardware.es/mejores-portatiles-para-programadores/>.

