

MÁSTER UNIVERSITARIO EN TECNOLOGÍAS WEB,
COMPUTACIÓN EN LA NUBE Y APLICACIONES
MÓVILES



VNIVERSITAT
DE VALÈNCIA

TRABAJO DE FIN DE MÁSTER

PLATAFORMA GASTRONÓMICA PARA
PRINCIPIANTES: GESTIÓN Y CREACIÓN DE
RECETAS DE COCINA

AUTOR:
PABLO ALBERT PUCHOL

TUTORÍA:
INMACULADA COMA TATAY

JULIO, 2024



VNIVERSITAT
DE VALÈNCIA [Q] Escola Tècnica Superior
d'Enginyeria ETSE-UV

TRABAJO FIN DE MASTER

PLATAFORMA GASTRONÓMICA PARA PRINCIPIANTES: GESTIÓN Y CREACIÓN DE RECETAS DE COCINA

AUTOR: PABLO ALBERT PUCHOL

TUTOR: INMACULADA COMA TATAY

TRIBUNAL

PRESIDENTE/A:

VOCAL 1:

VOCAL 2:

FECHA DE DEFENSA:

CALIFICACIÓN:

Declaración de autoría:

Yo, Pablo Albert Puchol, declaro la autoría del Trabajo Fin de Grado titulado “Plataforma Gastronómica para Principiantes: Gestión y Creación de Recetas de Cocina” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual. El material no original que figura en este trabajo ha sido atribuido a sus legítimos autores.

Valencia, 12 de junio de 2024

Fdo: Pablo Albert Puchol

Resumen:

El presente proyecto se enfoca en la creación de dos aplicaciones: una móvil para la elaboración de recetas y otra web para la gestión de las mismas, así como de los ingredientes utilizados por la comunidad de la plataforma.

En el desarrollo de la aplicación móvil inicial, se empleará React junto con Firebase para garantizar una experiencia completa. Por otro lado, para la generación de la página web destinada a las labores administrativas, se optará por Angular y Wildfly.

Una vez completadas ambas aplicaciones, se llevarán a cabo pruebas exhaustivas para evaluar su funcionamiento, rendimiento y usabilidad.

El propósito último consiste en establecer una plataforma donde miembros de una comunidad con diversos niveles de experiencia culinaria puedan compartir tanto recetas tradicionales como originales. Además, se busca facilitar la gestión de recetas e ingredientes para el propietario de la plataforma, contribuyendo así a una experiencia culinaria enriquecedora y colaborativa.

Abstract:

This project focuses on the creation of two applications: a mobile one for preparing recipes and another website for managing them, as well as the ingredients used by the platform community.

In the initial mobile app development, React will be used along with Firebase to ensure a complete experience. On the other hand, to generate the web page for administrative tasks, Angular and Wildfly will be chosen.

Once both applications are completed, extensive testing will be carried out to evaluate their operation, performance and usability.

The ultimate purpose is to establish a platform where members of a community with various levels of culinary experience can share both traditional and original recipes. In addition, the aim is to facilitate the management of recipes and ingredients for the owner of the platform, thus contributing to an enriching and collaborative culinary experience.

Agradecimientos:

A mi tutora, por apoyarme en el desarrollo de este proyecto.

Al profesorado del máster, por aportarme críticas constructivas desde diferentes puntos de vista y proporcionarme consejos en los diversos problemas que fueron surgiendo durante el desarrollo del proyecto.

Finalmente, a mi familia y amigos, por haberme apoyado desde mis primeros años de aprendizaje hasta mis últimos años de vida universitaria.

Índice general

1. Introducción	25
1.1. Contexto	25
1.2. Objetivos	26
2. Estado del Arte	29
2.1. Introducción	29
2.2. Comunidad a través del diálogo y la participación	30
2.3. Herramienta para fomentar el ocio	31
2.4. Rentabilidad de Digitalizar las Recetas de Cocina	32
2.5. Tecnologías empleadas en la plataforma	32
2.5.1. Frameworks de desarrollo de aplicaciones web	33
2.5.2. Desarrollo de aplicaciones móviles	34
2.5.3. Plataforma de servicios	35
2.6. Antecedentes : aplicaciones móviles similares	35
2.6.1. Tasty [2]	36
2.6.2. Cookpad [6]	37
2.6.3. Nestlé Cocina [4]	38
3. Especificación del proyecto	41
3.1. Introducción	41
3.2. Requisitos	42
3.2.1. Requisitos Funcionales	42
3.2.2. Requisitos No Funcionales	43
3.3. Especificación de las funcionalidades	44
3.4. Gestión de alcance	44
3.5. Planificación temporal	48
3.6. Estimación de costes	51
3.6.1. Costes de hardware	52
3.6.2. Costes de software	53
3.6.3. Costes personal	54
3.7. Viabilidad y Riesgos	55
3.7.1. Viabilidad económica	55
3.7.2. Viabilidad legal	56
3.7.3. Análisis de riesgos	57
4. Desarrollo	59

4.1. Introducción	59
4.2. Análisis	59
4.2.1. Casos de uso	60
4.2.2. Diagrama de estados	67
4.2.3. Diagrama de actividad	67
4.3. Diseño	70
4.3.1. Diseño de interfaces de usuario de la aplicación móvil	70
4.3.2. Diseño de base de datos	77
4.3.3. Diagrama de clases	80
4.3.4. Diagramas de secuencia	93
4.3.5. Herramientas de desarrollo	100
4.3.6. Lenguaje de programación	102
4.3.7. Componentes externos	102
4.3.8. Desarrollo de la conexión entre la aplicación móvil y la página web .	104
4.4. Desarrollo de la Aplicación Móvil	109
4.4.1. Interfaces de listado de recetas	110
4.4.2. Interfaz de creación de receta	111
4.4.3. Interfaz de información de receta	112
4.4.4. Interfaces de favoritos	113
4.4.5. Interfaz de clasificación	114
4.4.6. Interfaz de búsqueda	115
4.4.7. Interfaces de cuenta	116
4.5. Desarrollo de la Página Web	117
4.5.1. Interfaz de inicio	118
4.5.2. Interfaz de inicio de sesión	119
4.5.3. Interfaz de gestión de ingredientes	120
4.5.4. Interfaz de gestión de recetas	121
4.5.5. Apirest	121
5. Pruebas y resultados	123
5.1. Introducción	123
5.2. Prueba de rendimiento	124
5.3. Prueba de funcionalidad con usuarios	126
5.4. Test SUS	128
6. Conclusiones	131
6.1. Conclusiones	131
6.2. Trabajo Futuro	132
Bibliografía	135

Índice de figuras

2.1. Esquema de funcionamiento del proyecto	30
2.2. Logo de la app de Tasty	37
2.3. Interfaz de las pantallas de Cookpad	38
2.4. Logo de la app de Nestle	39
3.1. Diagrama de Gantt (primera parte)	51
4.1. Diagrama de casos de uso de la aplicación móvil	60
4.2. Diagrama de casos de uso de la página web	64
4.3. Diagrama de actividad del sistema de la aplicación web	68
4.4. Diagrama de actividad de la aplicación móvil.	69
4.5. Diagrama de UI del listado de recetas	70
4.6. Diagrama de UI del listado de favoritos del usuario	71
4.7. Diagrama de UI de búsqueda y ranking	72
4.8. Diagrama de UI de gestión de la sesión	72
4.9. Diagrama de UI para crear una receta	73
4.10. Diagrama de UI de Opciones	74
4.11. Diagrama de UI del inicio de la web sin iniciar sesión	74
4.12. Diagrama de UI del inicio de la web	75
4.13. Diagrama de UI de inicio de sesión	75
4.14. Diagrama de UI de gestión de ingredientes	76
4.15. Diagrama de UI de gestión de recetas	76
4.16. Diagrama del modelo de datos para la aplicación de web	79
4.17. Diagrama del modelo de datos para la aplicación de móvil	80
4.18. Diagrama de clases de la plataforma	83
4.19. Diagrama de secuencia de “Iniciar Sesión”en Móvil	94
4.20. Diagrama de secuencia de “Registrarse”	95
4.21. Diagrama de secuencia de “Listar Recetas”	96
4.22. Diagrama de secuencia de “Listar Ranking de Recetas”	97
4.23. Diagrama de secuencia de “Buscar Receta”	97
4.24. Diagrama de secuencia de “Mostrar Información Receta”	98
4.25. Diagrama de secuencia de “Iniciar Sesión”en Web	99
4.26. Diseño final de interfaces de listado de recetas	110
4.27. Diseño final de interfaz de información de creación de receta	111
4.28. Diseño final de interfaz de información de receta	112
4.29. Diseño final de interfaces de favoritos	113
4.30. Diseño final de interfaz de clasificación	114

4.31. Diseño final de interfaz de búsqueda	115
4.32. Diseño final de interfaces de cuenta	116
4.33. Diseño final de interfaz de inicio	118
4.34. Diseño final de interfaz de inicio de sesión	119
4.35. Diseño final de interfaz de gestión de ingredientes	120
4.36. Diseño final de interfaz de gestión de recetas	121
5.1. Capturas de los distintos finales del juego	124
5.2. Gráfica del rendimiento de la página web	125
5.3. Gráfica del rendimiento de la aplicación móvil	125
5.4. Gráfica de resultado de las pruebas de funcionalidad	127

Índice de ecuaciones

3.1. Estimación temporal por tres valores	48
---	----

Índice de cuadros

2.1. Comparación de Spring, React y Angular con notas como Muy Malo, Regular, Bueno y Excelente.	33
2.2. Comparación de los diferentes software de desarrollo de aplicaciones móviles.	34
2.3. Comparación de las diferentes plataformas para proporcionar servicios a las aplicaciones.	35
3.1. Especificaciones del sistema.	44
3.2. Estructura de descomposición de tareas (EDT) - Primera parte	45
3.3. Estructura de descomposición de tareas (EDT) - Segunda parte	46
3.4. Lista de primera entrega	47
3.5. Lista de segunda entrega	47
3.6. Lista de tercera entrega	47
3.7. Lista de cuarta entrega	47
3.8. Lista de quinta entrega	48
3.9. Lista de sexta entrega	48
3.10. Estimación de tres valores del proyecto.	49
3.11. Estimación temporal final por bloque de tareas.	50
3.12. Información portátil utilizado.	52
3.13. Información pantalla auxiliar utilizada.	52
3.14. Información ratón utilizado.	52
3.15. Información tableta gráfica utilizada.	53
3.16. Coste económico de los componentes hardware del equipo utilizado por los miembros.	53
3.17. Costes de licencia de los softwares utilizados.	54
3.18. Costes del personal presente en el proyecto.	54
3.19. Tabla de estimación de coste total del proyecto.	55
3.20. Tabla de estimación de coste del producto para obtener beneficios.	56
3.21. Lista de posibles riesgos del proyecto	57
4.1. Descripción del CU1 de “Iniciar Sesión“	61
4.2. Descripción del CU2 de “Gestionar Perfil“.	61
4.3. Descripción del CU3 de “Crear Receta“	62
4.4. Descripción del CU4 de “Listar Favoritos“.	62
4.5. Descripción del CU5 de “Añadir Favoritos“	62
4.6. Descripción del CU6 de “Cerrar Sesión“.	63
4.7. Descripción del CU7 de “Registrarse“.	63
4.8. Descripción del CU8 de “Listar Recetas“	63

4.9. Descripción del CU9 de “Listar Ranking de Recetas“.	63
4.10. Descripción del CU10 de “Buscar Receta“.	64
4.11. Descripción del CU11 de “Mostrar Información de Receta“.	64
4.12. Descripción del CU1 de “Iniciar Sesión“.	65
4.13. Descripción del CU2 de “Eliminar Ingrediente“.	65
4.14. Descripción del CU3 de “Crear Ingrediente“.	65
4.15. Descripción del CU4 de “Eliminar Receta“.	66
4.16. Descripción del CU5 de “Cerrar Sesión“.	66
4.17. Descripción de la clase “Aplicación Móvil“.	84
4.18. Descripción de la clase “Ingrediente“.	84
4.19. Descripción de la clase “Receta“.	85
4.20. Descripción de la clase “PantallaCrearReceta“.	86
4.21. Descripción de la clase “PantallaInfoReceta“.	86
4.22. Descripción de la clase “PantallaListaRecetas“.	87
4.23. Descripción de la clase “PantallaRanking“.	87
4.24. Descripción de la clase “PantallaRanking“.	87
4.25. Descripción de la clase “PantallaListaFavoritos“.	88
4.26. Descripción de la clase “PantallaCuenta“.	88
4.27. Descripción de la clase “PantallaLoginMóvil“.	88
4.28. Descripción de la clase “PantallaRegistro“.	89
4.29. Descripción de la clase “UsuarioMovil“.	89
4.30. Descripción de la clase “Favoritos“.	89
4.31. Descripción de la clase “Comentario“.	90
4.32. Descripción de la clase “AplicaciónWeb“.	90
4.33. Descripción de la clase “PantallaInicio“.	90
4.34. Descripción de la clase “PantallaLoginWeb“.	91
4.35. Descripción de la clase “PantallaGestionRecetas“.	91
4.36. Descripción de la clase “PantallaGestionIngredientes“.	92
4.37. Descripción de la clase “UsuarioWeb“.	92
5.1. Resultados de la Prueba SUS	128

Índice de códigos

4.1. Ejemplo de consulta SQL en un servlet	77
4.2. Ejemplo de guardar una receta en Firestore	78
4.3. Función de obtención de ingredientes	104
4.4. Inicialización de Firebase	105
4.5. Servlet de gestión de recetas	106

Capítulo 1

Introducción

1.1. Contexto

En el ámbito de la gastronomía contemporánea, el intercambio de recetas y conocimientos culinarios ha experimentado una transformación notable. Históricamente, la transmisión de recetas se realizaba principalmente a través de libros de cocina, programas de televisión especializados y clases presenciales. Sin embargo, con el avance de la tecnología, especialmente en el ámbito digital, han surgido nuevas formas de compartir recetas de manera más accesible y rápida.

En la actualidad, la difusión de recetas se lleva a cabo a través de una variedad de medios digitales. Los vídeos instructivos en plataformas en línea y las redes sociales han ganado popularidad como formas efectivas de compartir recetas paso a paso, demostraciones de técnicas culinarias y consejos prácticos. Estos vídeos ofrecen una experiencia visual y práctica que atrae a una amplia audiencia, desde entusiastas de la cocina hasta profesionales gastronómicos.

Además de los videos, los blogs culinarios se han convertido en una fuente invaluable de inspiración y conocimiento. Los bloggers comparten sus experiencias personales, recetas probadas y consejos útiles en plataformas de blogging. Estos blogs suelen incluir fotografías atractivas de los platos, instrucciones detalladas y comentarios sobre ingredientes y técnicas de cocción.

Por otro lado, las aplicaciones móviles dedicadas a la cocina han revolucionado la forma en que las personas acceden a las recetas y organizan su experiencia culinaria. Estas aplicaciones ofrecen una amplia variedad de recetas clasificadas por categorías, ingredientes y preferencias personales. Además, muchas de estas aplicaciones permiten a los usuarios crear perfiles, guardar recetas favoritas y compartir sus propias creaciones culinarias con la comunidad.

Este panorama de intercambio de conocimientos culinarios a través de medios digitales se analiza en profundidad en el artículo de El Pespunte titulado "*¿Cómo compartir tus conocimientos por Internet?*"[8]. El artículo explora las diferentes formas en que las personas comparten sus conocimientos culinarios en línea, destacando la importancia de la democratización del conocimiento en el ámbito gastronómico.

En definitiva, el intercambio de recetas y conocimientos culinarios ha evolucionado con el avance de la tecnología digital. Desde videos instructivos hasta blogs y aplicaciones móviles, existen múltiples formas en las que las personas pueden compartir y acceder a recetas de manera rápida y conveniente. Este contexto tecnológico y cultural proporciona la base para el desarrollo de nuestro proyecto, que busca aprovechar estas tendencias para crear una plataforma especializada en la difusión colaborativa de recetas y consejos culinarios.

1.2. Objetivos

El propósito general del proyecto es desarrollar una aplicación móvil compatible con Android e iOS que permita a los usuarios subir y compartir recetas culinarias. Esta aplicación estará respaldada por un backend para gestionar la información de las recetas y los ingredientes. Además, se creará una aplicación web para administrar estas recetas e ingredientes, utilizando otro backend desarrollado en Java que facilite la interacción con el servidor de la aplicación móvil.

Para alcanzar este objetivo principal, se plantean los siguientes subobjetivos:

- Diseño de los recursos primitivos de nuestra plataforma:
 - Crear un conjunto coherente de imágenes para la interfaz de usuario de ambas aplicaciones, incluyendo iconos y elementos decorativos que reflejen la identidad visual de la marca.
 - Diseñar de las primeras imágenes de recetas para que sean visualmente atractivas y representen fielmente el tipo de comunidad que se quiere crear tanto en la aplicación móvil como en la web.
 - Adaptar las imágenes al formato requerido para ambas plataformas, considerando las dimensiones y resoluciones recomendadas para una óptima visualización.
- Desarrollo de la aplicación móvil
 - Implementar la interfaz de usuario de la aplicación móvil para que sea intuitiva y fácil de usar para los usuarios.

- Integrar la funcionalidad de subir recetas a la aplicación móvil, permitiendo a los usuarios agregar información detallada sobre las recetas, incluyendo ingredientes, instrucciones de preparación y fotografías.
 - Configurar la comunicación entre los dos backends para almacenar y recuperar la información de las recetas y los ingredientes.
 - Asegurar la compatibilidad de la aplicación móvil con los sistemas operativos Android e iOS, garantizando una experiencia consistente para todos los usuarios.
- Desarrollo de la aplicación web de administración
- Crear una interfaz web para administrar las recetas e ingredientes, permitiendo a los administradores gestionar el contenido de la plataforma de manera eficiente.
 - Desarrollar un panel de administración intuitivo y fácil de usar que proporcione funcionalidades como agregar, editar y eliminar recetas e ingredientes.
 - Integrar el backend desarrollado en Java para facilitar la interacción entre la aplicación web y el servidor que almacenará los datos de la aplicación móvil.
 - Garantizar la seguridad de la aplicación web mediante la implementación de medidas de autenticación y autorización adecuadas para proteger la información confidencial de los usuarios y administradores.

Capítulo 2

Estado del Arte

2.1. Introducción

A lo largo de este capítulo, exploraremos cómo la gastronomía puede servir como vehículo para crear comunidades a través del diálogo y la participación, así como para facilitar la socialización y el conocimiento entre personas. Nos adentraremos en cómo esta aplicación móvil puede llegar a ser un negocio en el futuro, centrándonos en la creación de una comunidad de cocina entre personas que desean aprender o mejorar sus habilidades culinarias. Además, se revisarán las tecnologías más idóneas para llevar a cabo el proyecto de manera eficiente, tanto en términos temporales como económicos. También se realizará una exhaustiva búsqueda de antecedentes de aplicaciones móviles que fomenten la participación comunitaria en torno a la gastronomía o que puedan inspirar y enriquecer nuestras funcionalidades propuestas.

En esta sección, nos sumergiremos en el marco teórico general que sustenta las ideas de nuestro proyecto. Se expondrán los métodos en los cuales se basan nuestras propuestas, y se realizarán comparaciones entre las tecnologías de software recomendadas, considerando los beneficios que cada una pueda aportar en términos de funcionalidad y experiencia de usuario. Además, se analizarán los antecedentes existentes en el ámbito de las aplicaciones móviles gastronómicas, que nos servirán como puntos de referencia y competidores a los cuales aprender y, eventualmente, superar.

El propósito de este capítulo es repasar los fundamentos teóricos y tecnológicos esenciales para el desarrollo y comprensión de nuestro proyecto, así como identificar las oportunidades y desafíos que encontraremos en el camino. A través de un análisis detallado del estado actual del arte en el campo de las aplicaciones móviles de gastronomía y comunidad, buscamos construir sobre los conocimientos existentes y aportar un nuevo medio para compartir información sobre cocina.

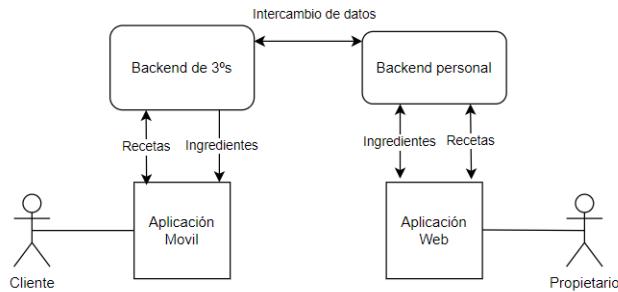


Figura 2.1: Esquema de funcionamiento del proyecto

En esta figura se muestra cómo fluiría la información en nuestra plataforma gastronómica y cómo se facilitaría su uso tanto para los clientes como para los propietarios de la aplicación. Se puede observar que se comparte el mismo tipo de datos, pero que las funcionalidades disponibles para cada tipo de usuario en cada aplicación serán diferentes, dependiendo del rol y las necesidades específicas de cada uno.

2.2. Comunidad a través del diálogo y la participación

En el contexto de nuestro proyecto, definimos *comunidad* como un grupo de individuos que comparten intereses, valores o propósitos comunes, y que trabajan juntos para lograr objetivos mutuos. Esta definición se alinea con la interpretación que se hace en el artículo "La educación inclusiva hoy. Reconocer al otro y crear comunidad a través del diálogo y la participación" de Teresa Susinos Rada y Carlos Rodríguez-Hoyos [9], donde se destaca la importancia de la comunidad escolar como un espacio acogedor donde cada miembro es reconocido y valorado, y donde se promueve la participación activa y la inclusión de todos.

Nuestro proyecto se inspira en esta visión de comunidad gastronómica, con el objetivo de fomentar un ambiente en el que todos los jóvenes, independientemente de sus habilidades culinarias o antecedentes, se sientan parte integral de la escena gastronómica local. Reconocemos que la inclusión no es solo un discurso, sino un compromiso activo de crear un entorno en el que cada individuo tenga voz y pueda contribuir al desarrollo y la mejora de la comunidad gastronómica.

Para lograr este objetivo, nuestro proyecto se centra en el empoderamiento de los jóvenes a través del diálogo y la participación activa en la organización y ejecución de clasificaciones gastronómicas, donde varios usuarios pueden competir la cantidad de valoraciones que se les proporcione. Creemos que al dar voz a los jóvenes y permitirles participar en

este evento gastronómico, no solo promovemos su sentido de pertenencia y criterio, sino que también fomentamos una cultura de colaboración y creatividad en el ámbito culinario.

Al igual que en el artículo mencionado, reconocemos que el desarrollo de una comunidad gastronómica inclusiva es un proceso continuo y en constante evolución. Por lo tanto, nuestro proyecto se compromete a mantener una actitud crítica y reflexiva, revisando constantemente nuestras prácticas y políticas para asegurarnos de que estén alineadas con los principios de inclusión y participación. Creemos que este enfoque nos permitirá responder de manera efectiva a las necesidades cambiantes de nuestra comunidad gastronómica y trabajar hacia un ambiente más equitativo y enriquecedor para todos sus miembros.

2.3. Herramienta para fomentar el ocio

El concepto de gastronomía social, como se expone en el artículo "Gastronomía social: La cocina tiene el poder de mejorar la sociedad" de Ivan Caraballo [1], destaca el potencial transformador de la cocina en la sociedad. Este enfoque no solo se limita a los grandes restaurantes o establecimientos de renombre, sino que abarca una amplia gama de iniciativas, desde proyectos comunitarios hasta acciones individuales de emprendedores locales.

La gastronomía social reconoce la estrecha relación entre la preparación de alimentos y el impacto en la salud, el medio ambiente y el desarrollo social. A través de programas como Gastromotiva en Brasil o iniciativas locales como Heladera Social en Argentina, se evidencia cómo la cocina puede utilizarse como una herramienta para promover la educación, el empleo y la inclusión social.

Estos ejemplos muestran que la gastronomía va más allá de la mera preparación de alimentos; es un medio para generar un cambio positivo en la comunidad. La implementación de prácticas sostenibles y responsables en el sector gastronómico no solo beneficia al medio ambiente, sino que también fortalece los lazos sociales y contribuye al desarrollo económico local.

En este sentido, la plataforma busca fomentar la gastronomía como un medio para socializar y aprender, inspirada en la filosofía de la gastronomía social. Al proporcionar una amplia variedad de recetas y facilitar la interacción entre usuarios, nuestra plataforma busca promover la educación culinaria y el intercambio cultural. Además, al destacar estas iniciativas de gastronomía social en nuestro contenido de referencia, buscamos inspirar a los usuarios a adoptar prácticas responsables y a contribuir positivamente a la sociedad a través del arte culinario.

2.4. Rentabilidad de Digitalizar las Recetas de Cocina

En la era digital actual, la integración de la tecnología en nuestra vida cotidiana se ha vuelto una práctica generalizada. Las aplicaciones móviles, en particular, desempeñan un papel fundamental al proporcionarnos acceso a una amplia gama de servicios y actividades con solo unos pocos toques en nuestros dispositivos móviles.

En este contexto, las aplicaciones de recetas de cocina han ganado popularidad de manera significativa en los últimos años. Estas aplicaciones no solo ofrecen una amplia variedad de recetas para los entusiastas de la cocina, sino que también representan una excelente oportunidad de negocio para desarrolladores y marcas.

La rentabilidad de digitalizar las recetas de cocina radica en varios aspectos clave. En primer lugar, estas aplicaciones permiten a los desarrolladores llegar a una amplia audiencia de usuarios interesados en la gastronomía, lo que facilita la distribución de marcas y la promoción de productos y servicios relacionados con la cocina.

Además, según el artículo "App de recetas de cocina: Oportunidad de negocio" publicado por Zoom [12], las aplicaciones de recetas de cocina ofrecen diversas formas de generar ingresos. Desde la publicidad hasta la venta de productos y servicios dentro de la aplicación, los desarrolladores tienen múltiples opciones para monetizar sus aplicaciones y obtener beneficios económicos.

Otro aspecto importante es la capacidad de estas aplicaciones para retener y fidelizar a los usuarios. Al ofrecer contenido de calidad y personalizado, las aplicaciones de recetas pueden mantener a los usuarios comprometidos y hacer que regresen por más, lo que contribuye a la rentabilidad a largo plazo del negocio.

Por último, estas aplicaciones también pueden ser una plataforma efectiva para promover restaurantes y chefs. Al ofrecer recetas de establecimientos reconocidos, las aplicaciones pueden aumentar la visibilidad de estos profesionales culinarios y atraer a más personas a sus locales, lo que beneficia tanto a los restaurantes como a los usuarios de la aplicación.

2.5. Tecnologías empleadas en la plataforma

En esta sección, se justifica el uso de las tecnologías seleccionadas para el desarrollo de la plataforma gastronómica. Se examinan diversas tecnologías web y móviles utilizadas en proyectos similares, así como aquellas presentadas durante el desarrollo académico.

La comparación de los diferentes tipos de tecnologías se realizará mediante tablas

comparativas que resaltarán los beneficios y ventajas que ofrecen en relación a los aspectos relevantes para lograr una mayor eficiencia temporal y distribución de recursos.

La metodología empleada para la selección de una tecnología sobre otras se basará principalmente en la cantidad de ventajas que ofrezca en comparación con el resto. En caso de que la comparación arroje resultados similares, se dará prioridad a la importancia de las ventajas dentro del conjunto comparado.

2.5.1. Frameworks de desarrollo de aplicaciones web

En este subapartado, se realiza una comparativa entre los principales frameworks¹ de desarrollo de aplicaciones web utilizados en la industria actualmente. Los frameworks de desarrollo web desempeñan un papel crucial en la creación de aplicaciones modernas, proporcionando a los desarrolladores herramientas y estructuras que facilitan la construcción eficiente y escalable de aplicaciones web.

Se analizan y comparan diversos aspectos clave de cada framework, incluyendo su facilidad de aprendizaje, rendimiento, comunidad y soporte, escalabilidad, integración con otras tecnologías y costo. Esta comparativa ayudará a los desarrolladores y equipos de proyecto a tomar decisiones informadas sobre qué framework utilizar en función de las necesidades específicas de su proyecto y los requisitos del cliente.

A continuación, se presenta una descripción detallada de los principales frameworks de desarrollo de aplicaciones web seleccionados, así como una evaluación de sus características y funcionalidades:

	Spring	React	Angular
Facilidad de aprendizaje	Regular	Bueno	Excelente
Rendimiento	Regular	Excelente	Excelente
Comunidad y soporte	Bueno	Bueno	Bueno
Escalabilidad	Bueno	Excelente	Excelente
Integración con otras tecnologías	Malo	Excelente	Bueno
Precio	Regular	Bueno	Bueno

Cuadro 2.1: Comparación de Spring, React y Angular con notas como Muy Malo, Regular, Bueno y Excelente.

La selección de Angular [10] se justifica por su accesibilidad para aprender a utilizarla y por su capacidad para crear aplicaciones web con un alto nivel de profesionalismo. Además de ofrecer un excelente rendimiento durante la navegación y otras acciones cruciales,

¹Estructura básica y herramientas específicas para crear sitios web y aplicaciones de manera más eficiente

Angular cuenta con un sólido respaldo comunitario y de soporte, comparable con otros programas contemporáneos. Esta elección también se fundamenta en la capacidad de Angular para facilitar la escalabilidad de la aplicación en términos de tamaño y servicios, así como en su asequible precio, lo que lo hace accesible para la generación de diversos elementos web, muchos de los cuales pueden ser generados por la comunidad.

A parte, también destaca por la separación de responsabilidades y su estructura modular que permite que las aplicaciones sean mantenibles, comprobables y extensibles. Además, la seguridad de tipos proporcionada por Typescript ofrece una experiencia de desarrollo más sólida y segura, similar a otros lenguajes de programación como C# o Java.

2.5.2. Desarrollo de aplicaciones móviles

En este subapartado, se realiza una comparación entre las herramientas de desarrollo de aplicaciones móviles que permiten crear aplicaciones compatibles con dispositivos Android e iOS. Estos programas ofrecen una variedad de funciones y características destinadas a facilitar el proceso de desarrollo y optimizar el rendimiento de las aplicaciones en los dispositivos móviles.

Entre las herramientas más utilizadas, se destacan aquellas que ofrecen una curva de aprendizaje accesible y brindan un conjunto completo de recursos para el diseño, implementación y prueba de aplicaciones móviles. A continuación, se presentan los siguientes programas seleccionados para su comparación:

	React Expo	Kotlin	Android Studio
Plataforma objetivo	Excelente	Bueno	Regular
Productividad del desarrollador	Excelente	Excelente	Malo
Rendimiento de la aplicación	Bueno	Bueno	Malo
Compatibilidad y acceso a las características de la plataforma	Excelente	Excelente	Bueno
Comunidad y soporte	Malo	Excelente	Bueno
Precio	Excelente	Regular	Bueno

Cuadro 2.2: Comparación de los diferentes software de desarrollo de aplicaciones móviles.

La elección de React Expo [7] se basa en su excelencia en la plataforma objetivo y su destacada productividad para los desarrolladores. React Expo ofrece un entorno altamente eficiente que permite a los desarrolladores crear aplicaciones móviles con facilidad y rapidez, gracias a su enfoque en la reutilización de componentes y su amplia gama de herramientas y bibliotecas disponibles. Además, su sólido rendimiento y compatibilidad con la plataforma garantizan una experiencia de usuario fluida y satisfactoria. Sin embargo, se reconoce que el soporte comunitario podría ser mejorado para ofrecer una asistencia aún más robusta y completa. A pesar de este aspecto, la accesibilidad y asequibilidad de React Expo la convierten en una opción sólida y atractiva para el desarrollo de aplicaciones móviles, especialmente en proyectos con recursos limitados.

2.5.3. Plataforma de servicios

En este subapartado, se realiza una comparación entre diferentes plataformas de servicios para aplicaciones web y móviles. Estas plataformas ofrecen una variedad de servicios que van desde la autenticación de usuarios hasta el almacenamiento y la gestión de datos, así como la actualización de información en tiempo real.o.

Se han seleccionado los siguientes programas, que son ampliamente utilizados y cuentan con una metodología de aprendizaje ágil, para llevar a cabo la comparación:

	Wildfly	Spring	Firebase
Escalabilidad y rendimiento	Excelente	Regular	Excelente
Facilidad de configuración y despliegue	Excelente	Malo	Excelente
Costos de mantenimiento	Bueno	Regular	Bueno
Cantidad de funcionalidades	Excelente	Bueno	Excelente
Integración con otras tecnologías	Excelente	Malo	Bueno
Precio de adquisición	Excelente	Regular	Excelente

Cuadro 2.3: Comparación de las diferentes plataformas para proporcionar servicios a las aplicaciones.

La decisión de utilizar Wildfly [11] como backend ² para la aplicación Angular se basa en varios factores. Principalmente, Wildfly ofrece una buena escalabilidad y rendimiento, lo que es crucial para manejar el flujo de datos y solicitudes en una aplicación web con un alto volumen de usuarios. Además, su integración con otras tecnologías es aceptable, lo que facilita la conexión con otras herramientas y servicios necesarios para el desarrollo de la aplicación.

Por otro lado, Firebase [5] se elige como la opción preferida para ambas aplicaciones, tanto móviles como web. Esta elección se justifica por la excelente escalabilidad y rendimiento que ofrece Firebase al igual que la anterior, así como su facilidad de configuración y despliegue. Firebase permite una rápida implementación de características como la autenticación de usuarios, el almacenamiento de datos en la nube y la sincronización en tiempo real, lo que lo convierte en una opción atractiva y versátil para ambas plataformas.

2.6. Antecedentes : aplicaciones móviles similares

En este apartado, se resalta la importancia de revisar aplicaciones móviles líderes que comparten características similares a las que buscamos implementar en nuestro proyecto actual. Se utilizarán estas referencias para obtener una comprensión más clara sobre la

²Parte de un sistema informático que se encarga de procesar y gestionar los datos, generalmente de manera invisible para el usuario final

personalización de la estructura y la dinámica que queremos generar con algunos elementos de esta aplicación. Algunos objetivos de esta revisión incluyen:

Además, es crucial realizar esta revisión de antecedentes por los siguientes motivos:

- Obtener perspectivas sobre las mejores prácticas y tendencias actuales en el diseño y desarrollo de aplicaciones móviles, lo que nos permitirá crear una experiencia de usuario moderna y atractiva.
- Aprender de métodos de autenticación de usuarios utilizados en aplicaciones similares para implementar un proceso seguro y fácil de usar.
- Identificar posibles desafíos y obstáculos que otros proyectos similares hayan enfrentado en términos de personalización de la navegación, autenticación de usuarios y distribución de componentes. Esto nos ayudará a anticipar y evitar posibles problemas durante la implementación de nuestro proyecto.

2.6.1. Tasty [2]

Tasty es conocida por su interfaz intuitiva y su capacidad para ofrecer una experiencia de usuario fluida y atractiva en la búsqueda y visualización de recetas de cocina. La distribución de elementos en la aplicación se centra en facilitar la navegación del usuario y en resaltar visualmente los pasos para la preparación de cada receta.

Las principales características de la distribución de elementos y la navegabilidad en Tasty son:

- Navegación fluida: La aplicación ofrece una navegación fluida entre las diferentes secciones, como el buscador de recetas, el listado de favoritos, las recetas guardadas y las recomendaciones personalizadas.
- Presentación visual: Las recetas se presentan de manera visualmente atractiva, con imágenes de alta calidad que muestran cada paso de preparación y el resultado final.
- Interacción intuitiva: Tasty permite una interacción intuitiva con las recetas, como la posibilidad de marcar los ingredientes necesarios, ajustar las cantidades según el número de porciones y guardar las recetas para consultarlas posteriormente.

Es importante destacar que la distribución de elementos y la navegabilidad en Tasty se diseñan con el objetivo de proporcionar una experiencia de usuario satisfactoria y eficiente en la búsqueda y preparación de recetas de cocina.

Sin embargo, es importante tener en cuenta que Tasty presenta una limitación significativa: los usuarios no pueden contribuir con nuevas recetas, ya que estas están pregeneradas en la aplicación o son actualizadas exclusivamente por el equipo de desarrollo. Esta restricción puede limitar la capacidad de los usuarios para compartir sus propias creaciones culinarias y participar activamente en la comunidad de la aplicación.



Figura 2.2: Logo de *Tasty*.

2.6.2. Cookpad [6]

Cookpad es reconocida por su enfoque en permitir a los usuarios compartir sus propias recetas de cocina de manera sencilla y accesible. La aplicación se destaca por su interfaz intuitiva y su capacidad para inspirar a los usuarios a experimentar en la cocina.

Las principales características de nuestro interés de la aplicación de Cookpad son:

Compartir recetas: Cookpad permite a los usuarios compartir sus propias recetas de cocina de manera fácil y rápida, lo que fomenta una comunidad activa de cocineros aficionados.

Organización por categorías: Las recetas están organizadas en categorías y etiquetas, lo que facilita la búsqueda y exploración según las preferencias individuales de los usuarios.

Comentarios y valoraciones: Los usuarios pueden dejar comentarios y valoraciones en las recetas, lo que ayuda a otros usuarios a tomar decisiones informadas al cocinar.

Personalización de la experiencia: Cookpad ofrece funciones de personalización, como la posibilidad de guardar recetas favoritas y recibir recomendaciones basadas en las preferencias del usuario.

A diferencia de Cookpad, nuestra aplicación se enfocará en atender las necesidades

específicas de un público inexperto en la cocina, como universitarios o personas que están comenzando a cocinar por sí mismas. Nos centraremos en proporcionar recetas simples y fáciles de seguir, adaptadas a las habilidades y preferencias culinarias de este público objetivo.

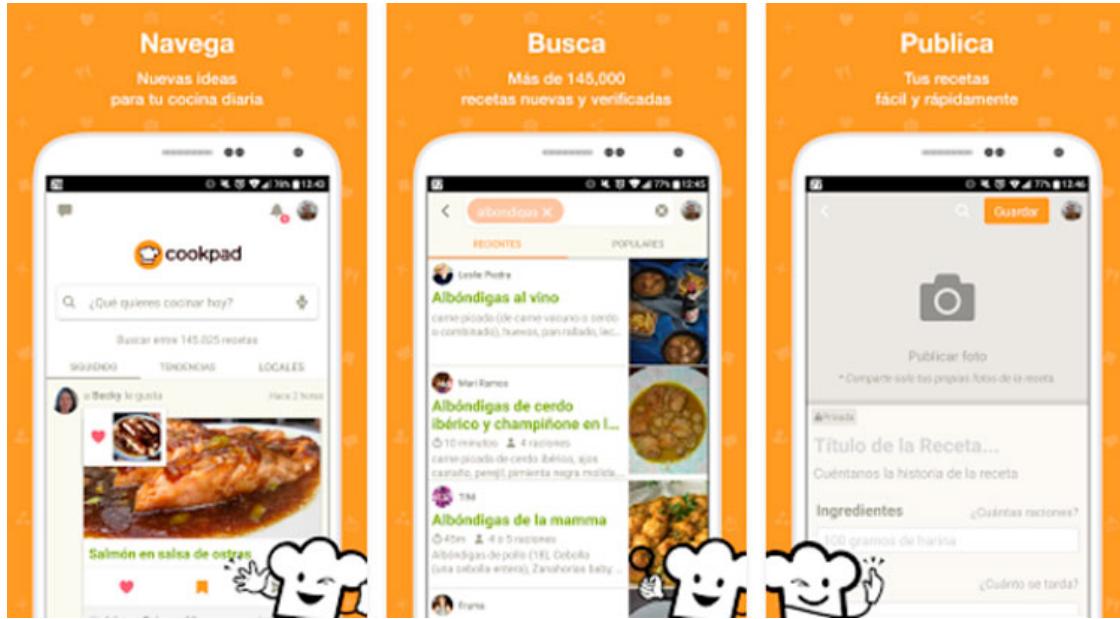


Figura 2.3: Interfaz de las pantallas de *Cookpad*.

2.6.3. Nestlé Cocina [4]

Nestlé Cocina es conocida por ofrecer una amplia variedad de recetas y menús para inspirar a los usuarios en la cocina. La aplicación se destaca por su colección diversa y su enfoque en proporcionar opciones saludables y equilibradas.

Las principales características que nos interesaron sobre la aplicación de Nestlé Cocina eran:

Gran variedad de recetas: La aplicación cuenta con una amplia selección de recetas de cocina, desde platos principales hasta postres, adaptadas a diferentes gustos y necesidades dietéticas.

Exploración y búsqueda intuitivas: Los usuarios pueden explorar y buscar recetas fácilmente mediante filtros por tipo de plato, tiempo de preparación, ingredientes y más, lo que facilita encontrar la receta perfecta para cada ocasión.

Ideas de menús: Nestlé Cocina ofrece sugerencias de menús completos para diferentes

momentos del día, ayudando a los usuarios a planificar comidas equilibradas y variadas.

Es importante resaltar que Nestlé, al asociar su nombre con la aplicación, ha contribuido significativamente a su fama y popularidad. Este aspecto es relevante ya que buscamos emular este éxito en nuestra propia aplicación, dotándola de un ícono distintivo que represente la comunidad y la colaboración entre los usuarios en el ámbito culinario. A diferencia de la aplicación mencionada, la nuestra se centrará en promover una participación activa de la comunidad en la creación y compartición de recetas.



Figura 2.4: Logo de *Nestlé Cocina*.

Capítulo 3

Especificación del proyecto

3.1. Introducción

En esta sección, presentamos una descripción detallada de nuestro proyecto, delineando sus estimaciones, alcance y requisitos principales. Aquí, exploraremos la esencia misma de nuestra propuesta revelando la visión y los resultados esperados que buscamos lograr.

Nuestro proyecto surge como respuesta a un análisis exhaustivo de las necesidades y tendencias recientes en el ámbito de las aplicaciones culinarias. Observamos una demanda creciente de soluciones que faciliten la preparación de alimentos y promuevan la interacción social en torno a la gastronomía.

A lo largo de esta especificación, resaltaremos los componentes esenciales que definen nuestro proyecto, desde los aspectos técnicos y funcionales hasta los operativos. Además, detallaremos los recursos requeridos, el cronograma de actividades y los entregables esperados.

En resumen, esta especificación del proyecto establece las bases sobre las cuales se erigirá y ejecutará todo el desarrollo. Servirá como un punto de referencia fundamental para una planificación precisa, una comunicación efectiva y el logro exitoso de nuestros objetivos. A medida que avancemos en esta especificación, el lector obtendrá una comprensión más completa y detallada de nuestro proyecto y sus implicaciones.

3.2. Requisitos

Los requisitos están estrechamente ligados con los objetivos que se desean alcanzar a través del proyecto y mediante su clarificación establecemos una guía sólida que orientará todas las etapas del proyecto.

En esta sección, se proporcionará una visión general de los requisitos funcionales y no funcionales en los que se pueden clasificar dependiendo en los atributos que se centren.

3.2.1. Requisitos Funcionales

Los requisitos funcionales son los encargados de especificar las funciones y características específicas que el sistema ha de ser capaz de proporcionar. En el caso que se presenta, son los siguientes:

- Implementar una aplicación móvil funcional con dos interfaces: una para clientes y otra para administradores.
- Desarrollar una interfaz móvil con navegación entre cuatro pantallas principales: Inicio, Favoritos, Ranking e Iniciar Sesión.
- Mostrar en Inicio una lista de recetas obtenidas desde un backend y permitir la visualización detallada de cada receta.
- Habilitar la funcionalidad de comentarios en las recetas para usuarios registrados.
- Integrar un botón para que los usuarios registrados puedan crear y subir nuevas recetas.
- Mostrar en Favoritos las recetas marcadas como favoritas por el usuario, permitiendo la gestión de esta lista.
- Mostrar en Ranking las recetas mejor valoradas según los comentarios almacenados en el backend.
- Implementar una pantalla de Iniciar Sesión que permita a los usuarios registrados editar su perfil y cerrar sesión.
- Desarrollar una interfaz web para administradores con dos secciones para modificar ingredientes y gestionar recetas.
- En la sección para modificar ingredientes, permitir la creación y eliminación de ingredientes almacenados en el backend.

- En la sección gestionar recetas, permitir la eliminación de recetas almacenadas que se consideren oportunas.
- Implementar acciones de búsqueda de recetas en la interfaz móvil.
- Integrar la promoción de recetas de ciertas tiendas a partir de publicaciones en la aplicación.
- Habilitar la función de personalización del perfil de usuario, incluyendo la edición de foto de perfil, nombre y correo electrónico.
- Crear un backend que simplifique los servicios a los administradores.

3.2.2. Requisitos No Funcionales

Estos requisitos complementan a los requisitos anteriores, centrándose en los atributos de calidad y aspectos técnicos, tales como rendimiento, seguridad y usabilidad. En el caso que nos ocupa, dichos requisitos se detallan a continuación:

- La aplicación debe ser intuitiva y fácil de usar para los usuarios finales.
- El tiempo de respuesta de la aplicación móvil y web debe ser rápido, garantizando una experiencia fluida para el usuario.
- La aplicación debe ser compatible con dispositivos móviles y navegadores web populares, asegurando una amplia accesibilidad.
- Se debe garantizar la seguridad de los datos del usuario, implementando medidas de cifrado y protección contra accesos no autorizados.
- Asegurar la escalabilidad del sistema para manejar un gran volumen de usuarios y recetas.
- La interfaz de usuario debe ser atractiva y estéticamente agradable, contribuyendo a una experiencia visualmente satisfactoria.
- Se debe garantizar la disponibilidad del servicio.
- La aplicación debe cumplir con las regulaciones de privacidad de datos y protección del consumidor vigentes en la jurisdicción donde se utilice.
- El rendimiento de la aplicación debe ser óptimo, incluso en condiciones de carga elevada o conexiones de red lentas.
- Usar React Expo para la aplicación móvil de clientes y Angular para la interfaz web de administradores.
- Utilizar Firebase como backend para almacenar y gestionar datos de recetas, comentarios, usuarios y favoritos.

3.3. Especificación de las funcionalidades

En esta parte se completa la descripción de cómo va a operar nuestro proyecto. A partir de los requisitos funcionales que hemos establecido previamente, se detallará un listado más extenso de funcionalidades.

Funcionalidades	
1º	Implementación de una aplicación móvil funcional para clientes
2º	Implementación de una aplicación web funcional para administradores
3º	Visualización de una lista de recetas en la pantalla de Inicio obtenidas desde un backend
4º	Creación de pantalla de detalles de cada receta
5º	Habilitación de la funcionalidad de comentarios en las recetas para usuarios registrados
6º	Permitir una gestión de las recetas favoritas para el usuario
7º	Visualización de las recetas mejor valoradas según los comentarios almacenados en el backend de la web
8º	Permitir a usuario registrados modificar sus perfiles
9º	Desarrollo de una interfaz web para administrar las recetas e ingredientes
10º	Permitir a administradores la creación y eliminación de ingredientes
11º	Permitir a administradores controlar las recetas creadas por los clientes
12º	Creación de un backend que simplifique los servicios a los administradores
13º	Creación de un backend que permita almacenar las recetas y usuarios de nuestra aplicación móvil
14º	La temática de las aplicaciones se adaptara al logo de la plataforma
15º	Implementar acciones de búsqueda de recetas en la interfaz móvil

Cuadro 3.1: Especificaciones del sistema.

3.4. Gestión de alcance

Este apartado tiene como objetivo definir y delimitar claramente los límites y metas del proyecto, garantizando así que se cumplan las expectativas de los interesados y se entreguen los resultados especificados.

Realizar una gestión del alcance implica analizar la temática que abarca el trabajo y determinar las actividades y tareas necesarias para alcanzar los resultados deseados, así como los recursos y el tiempo estimado para su realización.

Es importante tener en cuenta estos aspectos para poder trazar un proyecto sólido que no se desvíe durante su desarrollo. Siguiendo las tareas descritas en este apartado en orden, evitaremos olvidar algún punto importante y evitaremos problemas de dependencia en caso de tener una planificación incorrecta.

La gestión del alcance de este proyecto se basará en una Estructura de Descomposición de Tareas (EDT) que proporcionará una representación gráfica de la descomposición jerárquica del proyecto, dividiéndolo en elementos más pequeños llamados paquetes de trabajo. Cada paquete de trabajo corresponderá a una tarea o actividad específica que contribuye al logro del objetivo general del proyecto.

La posición de una tarea en la lista determinará su nivel de detalle y alcance. Las tareas de mayor jerarquía tendrán descripciones más generales y abarcarán un mayor contenido temático, mientras que las tareas de menor nivel estarán más detalladas y presentarán una estructura más simple.

Nombres de las tareas	Entreg.
1 Preparación	E1
1.1 Búsqueda bibliográfica y webgrafía	
1.2 Redacción de motivos y objetivos	
1.3 Redacción del Estado del Arte	
1.4 Definir requisitos del proyecto	
2 Análisis	E2
2.1 Crear "Casos de Uso"	
2.2 Crear "Diagrama de Estados"	
2.3 Crear "Diagrama de Actividad"	
2.4 Crear "Modelo de guardado de datos"	
2.5 Crear "Diagrama de Secuencia"	
2.6 Crear "Diagrama de Clases"	
2.7 Estimaciones temporales y económicas	
3 Diseño	E3
3.1 Diseño de la aplicación móvil	
3.1.1 Diseño de imágenes didácticas y decorativas	
3.1.2 Diseño de las interfaces de inicio	
[inicio + crear receta + información receta]	
3.1.3 Diseño de las interfaces de favoritos	
[lista favoritos + información receta]	
3.1.4 Diseño de las interfaces de ranking	
[lista mas votados + información receta]	
3.1.5 Diseño de las interfaces de búsqueda	
[lista de recetas + información receta]	
3.1.6 Diseño de las interfaces de sesión	
[iniciar sesión + crear sesión + perfil invitado + perfil registrado]	
3.2 Diseño de la navegación en la aplicación móvil	
3.1 Diseño de la aplicación web	
3.1.1 Diseño de imágenes didácticas y decorativas	
3.1.2 Diseño de la interfaz de inicio	
3.1.2 Diseño de la interfaz de gestión de recetas	
3.1.2 Diseño de la interfaz de gestión de ingredientes	
3.1.2 Diseño de la interfaz de iniciar sesión	
3.4 Diseño de las navegación aplicación web	
3.5 Diseño del backend de la aplicación web	

Cuadro 3.2: Estructura de descomposición de tareas (EDT) - Primera parte

Nombres de las tareas		Entreg.
4 Implementación		E4
4.1	Desarrollar frontend móvil	
4.2	Desarrollar frontend web	
4.3	Implementar backend móvil	
4.4	Implementar backend web	
4.5	Crear navegador con 4 pantallas	
4.5	Crear navegador con 4 pantallas	
4.6	Desarrollar pantalla de inicio	
4.7	Desarrollar página de detalle de receta	
4.8	Implementar comentarios	
4.9	Crear función para crear recetas	
4.10	Crear función para buscar recetas	
4.11	Implementar favoritos	
4.12	Crear página de ranking	
4.13	Implementar inicio de sesión	
4.14	Desarrollar navegación de usuario	
4.15	Implementar modificación de ingredientes	
4.16	Implementar eliminación de recetas	
4.16	Integrar API REST para aplicación móvil	
5 Pruebas y resultados		E5
5.1	Descripción de experimentos	
5.2	Resultados obtenidos	
6 Finalización		E6
6.1	Redacción de la memoria	
6.2	Calibración del programa	

Cuadro 3.3: Estructura de descomposición de tareas (EDT) - Segunda parte

En los EDT anteriores, se puede observar la división de las tareas en 3 grandes grupos que se identifican como niveles generales de estructuración del trabajo. A partir de estos puntos generales, se pueden derivar los elementos específicos que conformarán nuestro proyecto.

El primer grupo corresponde al análisis preliminar, donde se establecen los objetivos a cumplir o alcanzar en nuestro proyecto, las bases técnicas sobre las que se fundamenta nuestro tema y las motivaciones que impulsan su realización. Este grupo se aborda en los capítulos 1 y 2 de esta memoria.

A continuación, el siguiente bloque se refiere a todas las acciones concretas que dan forma a nuestro proyecto, y se desarrolla en este mismo capítulo. Aquí se realiza la planificación temporal y de alcance, se definen los requisitos y se realizan las estimaciones necesarias para una gestión óptima del proyecto. También se abordan las fases previas a las pruebas, como el diseño e implementación. Todo este trabajo y sus tareas se describe en el capítulo 4.

En el tercer apartado, centrado en las pruebas y los resultados, se aborda el capítulo 5, donde se documentan los diferentes experimentos realizados para el desarrollo final de nuestro módulo, y se presentan los resultados obtenidos a partir de diversos planteamientos.

Las entregas en este proyecto están relacionadas con los capítulos que deberían formar parte de una memoria de proyecto formal, y en ellos se incluirá la documentación pertinente para su corrección por parte del supervisor del proyecto.

E1	Preparación
Distribución temporal	Al finalizar la tarea 1
Descripción	En esta entrega se entregaría la preparación inicial del proyecto, que incluye la búsqueda bibliográfica y webgrafía relacionada, la redacción de los motivos y objetivos del proyecto, así como la redacción del Estado del Arte y la definición de los requisitos del proyecto.

Cuadro 3.4: Lista de primera entrega

E2	Análisis
Distribución temporal	Al finalizar la tarea 2
Descripción	En esta entrega se entregaría el análisis del proyecto, que incluye la creación de casos de uso, diagramas de estados, diagramas de actividad, modelo de guardado de datos, diagrama de secuencia y diagrama de clases. También se incluirían las estimaciones temporales y económicas del proyecto

Cuadro 3.5: Lista de segunda entrega

E3	Diseño
Distribución temporal	Al finalizar la tarea 3
Descripción	En esta entrega se entregaría el diseño de toda la plataforma, que abarca aspectos como la creación de imágenes que informaran a los usuarios el funcionamiento, hasta el establecimiento de navegación en ambas aplicaciones.

Cuadro 3.6: Lista de tercera entrega

E4	Implementación
Distribución temporal	Al finalizar la tarea 4
Descripción	En esta entrega se entregaría la implementación de la plataforma, que incluye la adquisición de los materiales visuales, como imágenes e iconos. También se realizaría la implementación del frontend y del backend de las 2 aplicaciones que conforman la plataforma, la navegabilidad que se presenta en cada una de ellas y se programaran e incorporaran las funciones que cumplen con los requisitos de nuestro proyecto.

Cuadro 3.7: Lista de cuarta entrega

E5	Pruebas y resultados
Distribución temporal	Al finalizar la tarea 5
Descripción	En esta entrega se entregarían las pruebas y resultados de las aplicaciones, que incluyen una descripción de los experimentos realizados y los resultados obtenidos en dichas pruebas.

Cuadro 3.8: Lista de quinta entrega

E6	Finalización
Distribución temporal	Al finalizar la tarea 6
Descripción	En esta entrega se entregaría la finalización del proyecto, que implica la redacción de la memoria del proyecto y la calibración de la plataforma.

Cuadro 3.9: Lista de sexta entrega

3.5. Planificación temporal

Dentro de la gestión del tiempo en los proyectos, es fundamental comenzar con la definición de las actividades a realizar. Estas actividades se traducen en tareas e hitos que conforman el alcance del proyecto, tal como se ha establecido en la sección anterior mediante la Estructura de Descomposición de Tareas (EDT).

El proceso de planificación temporal implica realizar un estudio de las duraciones estimadas para cada actividad y programar la secuencia entre ellas. Una vez que se ha establecido el conjunto de tareas, el siguiente paso es estimar el tiempo necesario para completar cada una de ellas.

Utilizaremos la técnica de estimación por tres valores, la cual consiste en obtener tres estimaciones para cada tarea: el tiempo más optimista (t_o), el más pesimista (t_p) y el más probable (t_m).

$$t_e = \frac{t_o + 4t_m + t_p}{6} \quad (3.1)$$

Al hacer uso de esta técnica, obtendremos una estimación más precisa y realista del tiempo necesario para completar cada tarea en el proyecto.

Gracias a la técnica mencionada anteriormente, se logró generar la siguiente tabla de estimaciones sin recurrir a la opinión de ningún experto, basando las estimaciones en la experiencia académica:

3.5. PLANIFICACIÓN TEMPORAL

49

Nombres de las tareas	t. optimo (horas)	t. medio (horas)	t. peor (horas)	t. estimado (horas)
1 Preparación	44	68	86	67,00
1.1 Búsqueda bibliográfica y webgrafía	12	24	30	23,00
1.2 Redacción de motivos y objetivos	4	8	10	7,67
1.3 Redacción del Estado del Arte	18	24	30	24,00
1.4 Definir requisitos del proyecto	10	12	16	12,33
2 Análisis	58	72	100	74,33
2.1 Crear "Casos de Uso"	10	12	16	12,33
2.2 Crear "Diagrama de Estados"	6	8	12	8,33
2.3 Crear "Diagrama de Actividad"	6	8	12	8,33
2.4 Crear "Modelo de guardado de datos"	6	8	12	8,33
2.5 Crear "Diagrama de Secuencia"	10	12	16	12,33
2.6 Crear "Diagrama de Clases"	10	12	16	12,33
2.7 Estimaciones temporales y económicas	10	12	16	12,33
3 Diseño	65	130	162	124,50
3.1 Diseño de la aplicación móvil	29	58	72	55,5
3.1.1 Diseño de imágenes didácticas y decorativas	2	4	6	4
3.1.2 Diseño de las interfaces de inicio	5	10	12	9,50
3.1.3 Diseño de las interfaces de favoritos	5	10	12	9,50
3.1.4 Diseño de las interfaces de ranking	5	10	12	9,50
3.1.5 Diseño de las interfaces de búsqueda	5	10	12	9,50
3.1.6 Diseño de las interfaces de sesión	5	10	12	9,50
3.1.7 Diseño de la navegación en la aplicación móvil	2	4	6	4
3.2 Diseño de la aplicación web	36	72	90	69
3.2.1 Diseño de imágenes didácticas y decorativas	2	4	6	4
3.2.2 Diseño de la interfaz de inicio	5	10	12	9,50
3.2.3 Diseño de la interfaz de gestión de recetas	5	10	12	9,50
3.2.4 Diseño de la interfaz de gestión de ingredientes	5	10	12	9,50
3.2.5 Diseño de la interfaz de iniciar sesión	5	10	12	9,50
3.2.6 Diseño de la navegación en la aplicación web	2	4	6	4
3.2.7 Diseño del backend de la aplicación web	12	24	30	23,00
4 Implementación	160	320	418	309,67
4.1 Desarrollar frontend móvil	20	40	50	38,33
4.2 Desarrollar frontend web	18	36	48	35
4.3 Implementar backend móvil	25	50	62	47,83
4.4 Implementar servicios web y móvil	97	194	258	188,50
4.5 Crear navegación entre páginas	15	30	40	29,17
4.6 Desarrollar pantalla de inicio (para aplicación web)	10	20	26	19,33
4.7 Implementar comentarios	7	14	20	13,83
4.8 Crear función para crear recetas	6	12	16	11,67
4.9 Crear función para buscar recetas	5	10	14	9,83
4.10 Implementar favoritos	7	14	20	13,83
4.11 Implementar ranking	6	12	16	11,67
4.12 Implementar inicio de sesión (para aplicación web)	8	16	20	15,33
4.13 Desarrollar navegación de usuario	10	20	26	19,33
4.14 Implementar modificación de ingredientes	6	12	16	11,67
4.15 Implementar eliminación de recetas	5	10	14	9,83
4.16 Integrar API REST (para aplicación móvil)	12	24	30	23
5 Pruebas y resultados	8	16	20	15,33
5.1 Descripción de experimentos	4	8	10	7,67
5.2 Resultados obtenidos	4	8	10	7,67
6 Finalización	44	80	96	76,67
6.1 Redacción de la memoria	32	56	64	53,33
6.2 Calibración del programa	12	24	32	23,33
TOTAL	379	686	882	667,50

Cuadro 3.10: Estimación de tres valores del proyecto.

A partir de los datos extraídos de las tablas anteriores, encontramos que el total de horas necesarias para la realización del proyecto sería de 667,5 horas.

El resultado proporcionado por estas estimaciones se presenta como una media en horas, sin embargo, para una mejor visualización en un diagrama de Gantt, se realizará la conversión a días (considerando un día de tiempo libre para trabajar de un estudiante universitario es de 4 horas) y se mostrarán las medias de cada una de estas tareas en días.

Bloque de tareas	Duración (horas)	Duración (días aprox.)
Preparación	67,00	16,75
Análisis	74,33	18,58
Diseño	124,5	31,125
Implementación	309,67	77,42
Pruebas y resultados	15,33	3,83
Finalización	76,67	19,17
Total	667,50	166,875

Cuadro 3.11: Estimación temporal final por bloque de tareas.

A continuación, presentamos las estimaciones previamente realizadas para cada tarea mediante un diagrama de Gantt. En este diagrama, las duraciones de las tareas se expresan en días estimados:

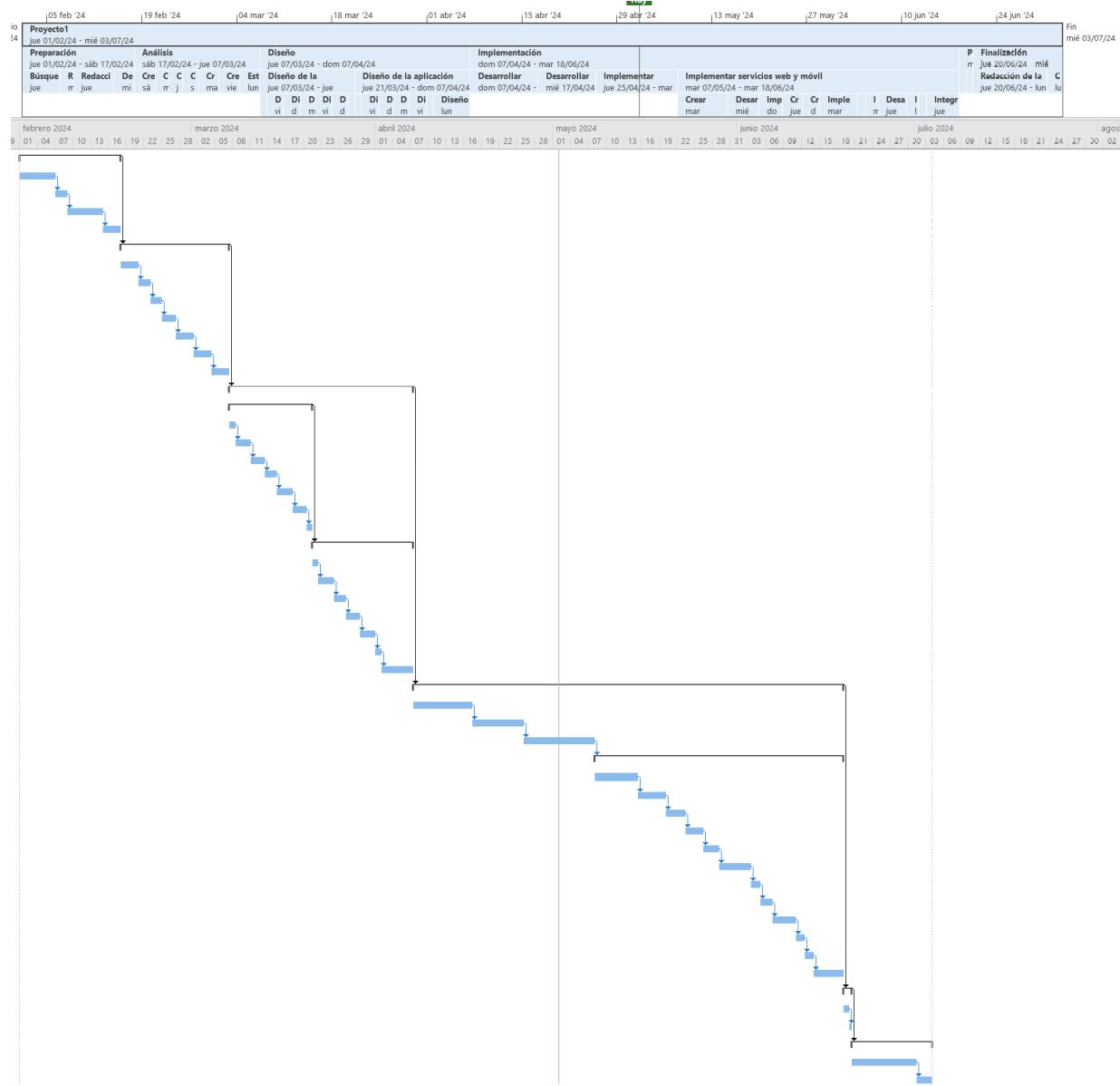


Figura 3.1: Diagrama de Gantt.

3.6. Estimación de costes

Esta sección está dedicada a la gestión de costos directos asociados al proyecto del Trabajo de Fin de Máster. Por lo tanto, se enfoca en la estimación de los valores de diferentes tipos de costos, como costos de hardware, software y personales (trabajo realizado por el propio equipo).

3.6.1. Costes de hardware

En lo relativo a los costes del hardware, en este proyecto se reduce al coste de los equipos informáticos usados para la realización de las actividades del personal.

Portátil

Nombre	ASUS TUF Gaming F15 FX507ZC4-HN002
Procesador	Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz 2.21 GHz
Disco Duro	1TB SSD
Memoria	16,0 GB
Gráfica	Intel Core i7-12700H
Pulgadas de pantalla	15.6"
Precio (PC Componentes)	1299 €

Cuadro 3.12: Información portátil utilizado.

Pantalla

Nombre	AOC 24B2XDAM
Procesador	Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz 2.21 GHz
Frecuencia de actualización	75Hz
Calidad imagen	FullHD
Pulgadas de pantalla	23.8"
Tiempo de respuesta	4 ms
Profundidad de color	8 bits
Precio (PC Componentes)	89,99 €

Cuadro 3.13: Información pantalla auxiliar utilizada.

Dispositivos de entrada

Ratón:

Nombre	Gaming Mouse iMice GW-X7
RGB	Si
Tecnología de conectividad	Inalámbrico
Ajustable DPI	Si
Precio (Amazon)	20,52 €

Cuadro 3.14: Información ratón utilizado.

Tableta Gráfica:

Nombre	Wacom Intuos Tableta Gráfica
Área activa	216 x 135 mm
Conexión Bluetooth	Si
Software creativos y educativos	Si
Sensibilidad	Ajustable
Precio (Amazon)	77,42 €

Cuadro 3.15: Información tableta gráfica utilizada.

Y lo que vamos a ver a continuación es el coste total del material hardware que se usara durante la vida del proyecto (teniendo en cuenta que se usan 225 días anualmente), sus años de vida útil y también su amortización, para poder tenerlos guardados para poder realizar nuestro presupuesto posteriormente con mayor facilidad.

Rol	Componente	Precio/Unidad	Vida Útil	% Uso	Amortización
Director	Portátil	1299€	4 años	5	12,05 €
	Ratón	20,52 €	2 años	4	0,3 €
Diseñador artístico	Portátil	1299€	4 años	34	81,95 €
	Pantalla	89,99 €	10 años	34	2,27 €
	Tableta gráfica	77,42 €	5 años	24	1,38 €
	Ratón	20,52€	2 años	20	1,52 €
Programador	Portátil	1299€	4 años	50	120,52 €
	Pantalla	89,99 €	10 años	50	3,34 €
	Ratón	20,52€	2 años	30	2,28 €
Diseñador técnico	Portátil	1299€	4 años	34	81,95 €
	Pantalla	89,99 €	10 años	34	2,27 €
	Ratón	20,52 €	2 años	20	1,52 €
Tester	Portátil	1299€	4 años	13	31,33 €
	Pantalla	89,99 €	10 años	12	0,8 €
	Ratón	20,52€	2 años	5	0,38 €
Total					343,86 €

Cuadro 3.16: Coste económico de los componentes hardware del equipo utilizado por los miembros.

Además, se considera una amortización de 6 € para los cables HDMI que se conectan a las pantallas, ya que son utilizados por primera vez para el proyecto.

3.6.2. Costes de software

En cuanto al software, se debería analizar el costo económico de utilizar los programas mencionados en el capítulo 2, así como otros utilizados para el desarrollo de la memoria, a fin de obtener el costo total de su uso en todo el proyecto. Sin embargo, como se ha

especificado en ese capítulo, la descarga de los programas es gratuita o son proporcionados por la universidad, por lo que no habría gastos asociados a la instalación básica. Además, al recomendar una estructuración sencilla del escenario para la creación del módulo, no se requerirán licencias de nivel avanzado para un desarrollo complejo del videojuego.

Por lo tanto, la tabla de software reflejará que no se generarán gastos económicos relacionados con el software que afecten al proyecto.

Software	Precio/Licencia
Angular	Código abierto (gratuito)
React Native Expo	Código abierto (gratuito)
Visual Paradigm	Código proporcionado por la universidad (gratuito)
Wildfly	Código abierto (gratuito)
Firebase	Dentro de la escala de este proyecto es gratuito
GanttProject	Código abierto (gratuito)
Overleaf	Plataforma de edición gratuita

Cuadro 3.17: Costes de licencia de los softwares utilizados.

3.6.3. Costes personal

En cuanto a los costes personales, nos referimos a los gastos asociados al personal que conforma el equipo encargado del desarrollo del proyecto. En esta sección, se detallarán los roles que asumirían los miembros del equipo y la duración de su participación a lo largo del proyecto.

Hay que tener en cuenta que la estimación de los salarios de las personas involucradas se basa en datos del mercado laboral de España durante el último año, utilizando información del Boletín Oficial del Estado (B.O.E.) [3] vigente hasta el año 2024, así como una búsqueda en portales de empleo como *LinkedIn*.

Rol	Salario Bruto (€/día)	Duración (días+)	Coste Total
Director general	79,62	8,35	664,83 €
Diseñador artístico	59,78	56,78	3.394,31 €
Diseñador técnico	61,73	56,78	3.505,03 €
Programador	59,78	83,5	4.991,63 €
Tester	42,25	21,71	917,25 €
TOTAL			13.473,05 €

Cuadro 3.18: Costes del personal presente en el proyecto.

3.7. Viabilidad y Riesgos

Para asegurar el desarrollo exitoso del proyecto hasta su finalización, es importante tener en cuenta varios aspectos que pueden tener implicaciones a largo plazo o dar lugar a problemas una vez finalizado. Por lo tanto, en esta sección se explicarán los diversos aspectos de viabilidad del proyecto y los posibles riesgos que podrían surgir durante su período de ejecución.

3.7.1. Viabilidad económica

En esta sección, evaluaremos la viabilidad del presupuesto calculado, considerando si es realista solicitar el coste total al responsable de financiar el proyecto y si existe posibilidad de obtener financiación para su realización.

Coste del proyecto					
Recursos	Duración (días)	Salario (€/día)	Unidades	Contingencias	Coste (€)
Coste personal					
Director general	8,35	79,62	-	-	664,83
Diseñador artístico	56,78	59,78	-	-	3.394,31
Diseñador técnico	56,78	61,73	-	-	3.505,03
Programador	83,5	59,78	-	-	4.991,63
Tester	21,71	42,25	-	-	917,25
Subtotal					13.473,05
Coste material					
Costes directos					
Recursos	Nº meses uso	Precio_Amortizado x mes	Unidades	% Uso	Coste Amortizado (€)
Tabla Hardware 3.16	6	-	15	100	343,86
Portátil Tabla Software 3.17	6	-	7	100	0,00
Costes indirectos					
Alquiler	6	300	-	-	1800,00
Luz	64	90	-	-	540,00
Agua	6	50	-	-	300,00
Internet + teléfono	6	90	-	-	540,00
Mantenimiento	6	45	-	-	270,00
Gastos Administrativos	6	250	-	-	1500,00
Subtotal					5293,86
TOTAL					18.766,91

Cuadro 3.19: Tabla de estimación de coste total del proyecto.

Precio de venta	
	Valor (€)
Precio total del proyecto	18.766,91
Beneficios (40 % sobre total)	7.506,76
Precio de venta (sin IVA)	26.273,67
21 % de IVA	5.517,47
Precio total de venta	31.791,14

Cuadro 3.20: Tabla de estimación de coste del producto para obtener beneficios.

3.7.2. Viabilidad legal

En este apartado, examinaremos los aspectos legales y regulaciones aplicables a la plataforma, que consta de un sitio web y una aplicación móvil, para evaluar su viabilidad legal. Algunos de los aspectos legales que deben considerarse incluyen:

- **Derechos de autor:** Se debe asegurar el uso de contenido original en el sitio web y la aplicación móvil, evitando el uso no autorizado de gráficos, música, texto u otros elementos protegidos por derechos de autor. Se anticipa que la mayoría de los elementos utilizados serán creados internamente o adquiridos de fuentes que permiten su uso comercial sin restricciones.
- **Marcas comerciales:** Es esencial evitar el uso no autorizado de marcas comerciales o nombres de empresas en la plataforma. Se asegurará de que el diseño y el contenido de la plataforma no contengan referencias a marcas comerciales reconocibles sin el permiso correspondiente.
- **Protección de datos:** Debe cumplirse con las leyes y regulaciones de protección de datos aplicables, como el Reglamento General de Protección de Datos (GDPR) en la Unión Europea, al manejar la información personal de los usuarios a través del sitio web y la aplicación móvil. Se implementarán medidas adecuadas para garantizar la privacidad y seguridad de los datos del usuario.
- **Licencias de software:** Se verificará si la plataforma utiliza software de terceros y se garantizará que el proyecto cumpla con los términos de las licencias correspondientes. Se revisarán detenidamente las licencias de cualquier software utilizado en el desarrollo y funcionamiento del sitio web y la aplicación móvil.
- **Contratos de colaboración:** En caso de colaboraciones externas para el desarrollo o mantenimiento de la plataforma, se establecerán contratos claros y vinculantes que definan los términos de la colaboración y los derechos de propiedad intelectual. Se asegurará de que cualquier colaborador externo esté legalmente obligado a respetar los términos del acuerdo.

- **Derechos de patentes:** Si la plataforma introduce características o tecnologías innovadoras susceptibles de protección mediante patentes, se considerará la posibilidad de obtener protección legal para asegurar la exclusividad de dichas innovaciones. Se consultará con expertos legales para evaluar la viabilidad y conveniencia de solicitar patentes en casos pertinentes.
- **Contratos de licencia de usuario final (CLUF):** Se establecerá un contrato de licencia de usuario final que establezca los términos y condiciones de uso de la plataforma por parte de los usuarios. Este contrato será accesible y aceptado por los usuarios al registrarse o utilizar la plataforma, y contendrá disposiciones importantes sobre responsabilidades legales, propiedad intelectual y privacidad de datos.

3.7.3. Análisis de riesgos

En este apartado, nos adentraremos en el análisis de los riesgos asociados a nuestro proyecto. El objetivo principal es evaluar y valorar detenidamente todas las posibles situaciones de riesgo que podrían afectar el desarrollo del proyecto, teniendo en cuenta tanto su probabilidad de ocurrencia como el impacto que podrían generar en el proyecto.

Es fundamental comprender y anticiparse a los posibles obstáculos que podrían surgir durante el proceso de desarrollo, ya que esto nos permitirá tomar medidas preventivas y establecer estrategias de mitigación adecuadas. Al identificar y analizar los riesgos, estaremos preparados para enfrentarlos de manera efectiva, minimizando cualquier impacto negativo en el proyecto.

En esta etapa, examinaremos diversos escenarios que son comúnmente considerados en el análisis de riesgos de proyectos, con el objetivo de abordarlos de manera proactiva y garantizar un desarrollo exitoso de la plataforma.

Escenarios de riesgo	Probabilidad de caso	Impacto de caso	Riesgo Inicial
Obtener requisitos poco claros o ambiguos	Alta	Medio	Alta
Planificar con dificultad	Media	Baja	Media
Conflictos entre compañeros de equipo	Baja	Baja	Baja
Mala asignación de recursos	Baja	Medio	Baja
Bajas laborales	Medio	Medio	Medio
Realizar un número de pruebas insuficientes	Baja	Alta	Medio
Ausencia injustificada de personal técnico	Baja	Baja	Baja
Aparición de costes imprevistos	Alto	Baja	Medio
Depender de terceros (ya sea por su tecnología o por sus activos)	Medio	Bajo (si tenemos en cuenta el apartado legal de antes)	Baja
Mal feedback sobre la plataforma	Medio	Alta	Alta
Discrepancias en los bocetos de diseño	Alta	Baja	Medio
Anadir nuevas características en cualquier punto del proyecto	Alta	Alta	Inaceptable
Problemas legales con otras empresas	Baja	Alta	Medio

Cuadro 3.21: Lista de posibles riesgos del proyecto

Es esencial llevar a cabo una exhaustiva planificación de los riesgos y establecer medidas de mitigación o contingencia para hacerles frente. Sin embargo, en ciertos casos críticos, es imperativo abordar de inmediato los problemas inaceptables antes de proseguir con

el proyecto. En nuestro caso, nos comprometemos inequívocamente a no añadir tareas adicionales más allá de las que se hayan planificado previamente.

En el caso de que surja el riesgo de recibir comentarios negativos sobre el proyecto, contamos con un plan de contingencia establecido. Este plan incluiría la contratación de terceros para crear diseños más atractivos para la comunidad y posponer los descansos del equipo.

Además, para reducir la presencia de requisitos poco claros o ambiguos, sería beneficioso realizar varias entrevistas con los clientes y preguntarles sobre los casos de uso específicos que desean abordar, con el objetivo de evitar malentendidos.

Mientras que para mitigar los riesgos descritos anteriormente de menor nivel, los trataríamos de la siguiente forma:

- Para evitar planificar con dificultad, en nuestro caso podemos acudir a la ayuda de los profesores que ya tienen experiencia en este campo.
- Para evitar o reducir el ausentismo laboral, se podría implementar un mantenimiento regular de los recursos utilizados por nuestro equipo, así como establecer períodos obligatorios de descanso lejos de las pantallas para permitir el descanso visual y relajación.
- Para evitar mal feedback ¹, se pueden realizar pruebas con usuarios en las fases de beta y que aunque el número pueda llegar a ser insuficiente, este pueda ayudar a realizar cambios antes.
- Para evitar costes imprevistos, se deberá de evaluar semanalmente los recursos usados y realizar una revisión del código.
- Si queremos evitar el uso de malos diseños, tendríamos que crear un grupo encargado de revisar que los bocetos presentan una calidad aceptable por el público.
- Para evitar posibles problemas legales, nos aseguraremos de utilizar contenido propio o recursos que sean públicamente reconocidos y legalmente utilizados por programadores o diseñadores en la industria. Esto garantizará que estemos utilizando materiales populares de forma legítima y evitaremos cualquier infracción de derechos de autor u otras cuestiones legales.

¹Es la información que se proporciona a una persona o entidad sobre su desempeño, comportamiento, trabajo realizado o producto creado

Capítulo 4

Desarrollo

4.1. Introducción

En este capítulo, daremos inicio al análisis detallado de los requisitos del proyecto, centrándonos en comprender las funcionalidades que este debe ofrecer y cómo se integran para brindar una experiencia completa al usuario.

Posteriormente, nos enfocaremos en el diseño del proyecto, donde definiremos la arquitectura general del sistema, donde también se incluirá la estructura de la interfaz de usuario, asegurándonos de que sea todo intuitivo y atractivo para el usuario final.

Una vez finalizado el análisis y diseño, comenzará la implementación donde se llevará a cabo la programación, creación e integración de los recursos del proyecto.

4.2. Análisis

En esta sección nos adentraremos en la fase de análisis, que marca el punto de partida para el desarrollo de nuestra plataforma. Durante esta etapa, se realizará un análisis detallado de los requisitos del proyecto y se establecerán las bases para su implementación.

En primer lugar, se elaborarán los casos de uso correspondientes a la plataforma, los cuales plasman los requisitos previamente explicados en la sección 3.2 de este capítulo. Estos casos de uso representan las distintas interacciones que los usuarios pueden tener con la plataforma y ayudan a comprender cómo se utilizará el sistema en la práctica. A través de estos casos de uso, se identificarán las principales funcionalidades y se establecerán los

escenarios de uso más relevantes.

Finalmente, se profundizará en el análisis a través de la elaboración de diagramas de actividad. Estos diagramas visualizan los procesos que los usuarios llevarán a cabo en la plataforma para cumplir con los distintos casos de uso identificados. Desde la interacción con la interfaz hasta el procesamiento de datos y la generación de resultados, los diagramas de actividad proporcionan una representación clara de la dinámica de uso de la plataforma.

4.2.1. Casos de uso

Para analizar nuestro proyecto, al igual que en asignaturas de otros cursos anteriores, comenzaremos por definir los casos de uso de manera gráfica y literal. Estos casos de uso describirán las funcionalidades principales que se alinean con los requisitos mencionados en el punto 3.2.

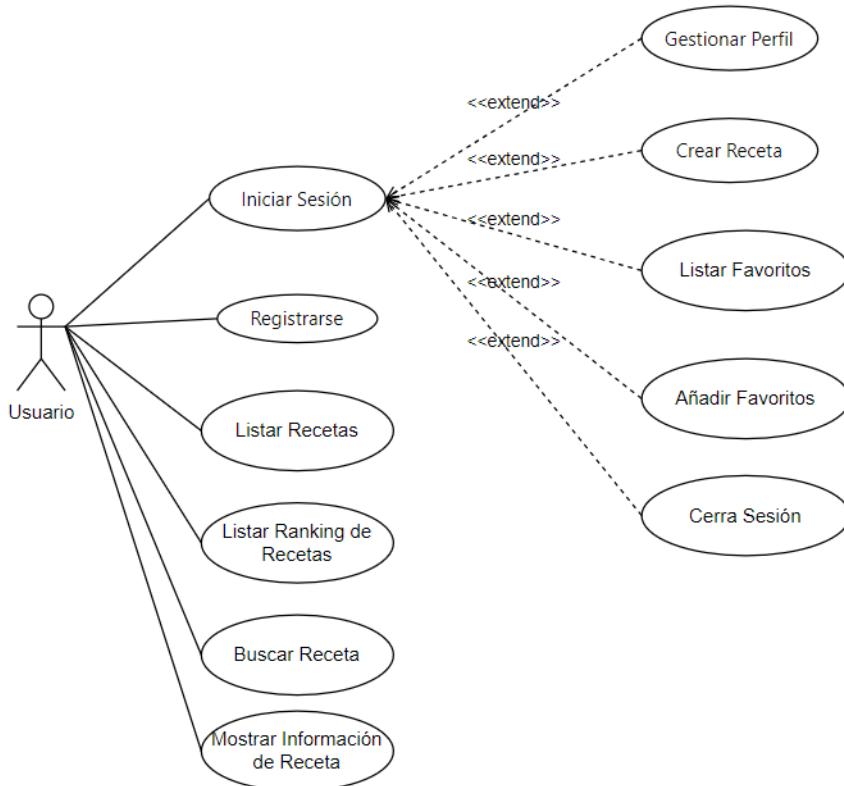


Figura 4.1: Diagrama de casos de uso de la aplicación móvil.

4.2. ANÁLISIS

61

ID Caso de Uso:	Iniciar Sesión
Actor principal:	Usuario
Actores secundarios:	
Breve descripción:	El usuario puede iniciar sesión con su cuenta
Precondiciones:	Necesita tener una cuenta creada para la plataforma y no tener iniciada la sesión
	Acciones del actor
	El usuario selecciona navegar a la página de su cuenta
	El sistema le devuelve un formulario para iniciar una cuenta
	El usuario inserta el correo y la contraseña
	El sistema comprueba que existe una cuenta con esos atributos
	Acciones del actor
	(Acción 3) El usuario inserta el correo y la contraseña
	El sistema devuelve la página con los datos de la cuenta mientras desbloquea funciones de la aplicación
Flujo de los eventos:	
	El sistema comprueba que existe una cuenta con esos atributos
Flujo alternativo	
(Cuenta inválida):	El sistema devuelve un mensaje de cuenta no válida
	Acciones del actor
	(Acción 5) El sistema devuelve la página con los datos de la cuenta mientras desbloquea funciones de la aplicación
Flujo alternativo	
(Gestionar la cuenta):	Véase CU2 "Gestionar Perfil"
	Acciones del actor
	(Acción 5) El sistema devuelve la página con los datos de la cuenta mientras desbloquea funciones de la aplicación
Flujo alternativo	
(Crear una nueva receta):	Véase CU3 "Crear Receta"
	Acciones del actor
	(Acción 5) El sistema devuelve la página con los datos de la cuenta mientras desbloquea funciones de la aplicación
Flujo alternativo	
(Mostrar la lista de favoritos de la cuenta):	Véase CU4 "Listar Favoritos"
	Acciones del actor
	(Acción 5) El sistema devuelve la página con los datos de la cuenta mientras desbloquea funciones de la aplicación
Flujo alternativo	
(Añadir una nueva receta a favoritos):	Véase CU5 "Añadir Favoritos"
	Acciones del actor
	(Acción 5) El sistema devuelve la página con los datos de la cuenta mientras desbloquea funciones de la aplicación
Flujo alternativo	
(Cerrar la sesión que se tiene):	Véase CU6 "Cerrar Sesión"
Postcondiciones:	—
Excepciones:	Cuando el usuario está iniciando sesión en una cuenta ya iniciada

Cuadro 4.1: Descripción del CU1 de "Iniciar Sesión".

ID Caso de Uso:	Gestionar Perfil
Actor principal:	Usuario
Actores secundarios:	
Breve descripción:	El usuario puede modificar información de su perfil
Precondiciones:	Necesita haber creado una cuenta previamente y estar iniciado en ella
	Acciones del actor
	El usuario selecciona un atributo a cambiar
	El sistema carga un formulario para cambiar ese atributo
	El usuario inserta el nuevo valor del atributo y la contraseña de su cuenta
	El sistema comprueba que la contraseña es correcta
	El sistema cambia el atributo de la cuenta en el backend
Flujo de los eventos:	
	Resuesta del Sistema
	Resposta del Sistema
	El sistema comprueba que la contraseña es correcta
	El sistema cambia el atributo de la cuenta en el backend
Flujo alternativo	
(contraseña incorrecta):	El sistema comprueba que la contraseña es correcta
	El sistema devuelve un mensaje de acción no permitida
Postcondiciones:	—
Excepciones:	—

Cuadro 4.2: Descripción del CU2 de "Gestionar Perfil".

ID Caso de Uso:	Crear Receta	
Actor principal:	Usuario	
Actores secundarios:	—	
Breve descripción:	El usuario puede crear una nueva receta para la plataforma	
Precondiciones:	Necesita haber creado una cuenta previamente y estar iniciado en ella	
	Acciones del actor	Respuesta del Sistema
	El usuario selecciona navegar a la página del listado de recetas	
	El usuario selecciona el botón para crear una nueva receta	
		El sistema carga un formulario para crear la nueva receta
	El usuario inserta los atributos de la nueva receta	
		El sistema comprueba que la receta se introdujo correctamente
		El sistema inserta en el backend la nueva receta
Flujo de los eventos:	—	
Flujo alternativo (receta inválida):	Acciones del actor (Acción 4) El usuario inserta los atributos de la nueva receta	Respuesta del Sistema El sistema comprueba que la receta se introdujo correctamente
		El sistema devuelve un mensaje de receta invalida mientras marca el campo incorrecto
Postcondiciones:	—	
Excepciones:	—	

Cuadro 4.3: Descripción del CU3 de “Crear Receta”.

ID Caso de Uso:	Listar Favoritos	
Actor principal:	Usuario	
Actores secundarios:	—	
Breve descripción:	El usuario puede visualizar su lista de recetas favoritas	
Precondiciones:	Necesita haber creado una cuenta previamente y estar iniciado en ella	
	Acciones del actor	Respuesta del Sistema
	El usuario selecciona navegar a la página del listado de favoritos	El sistema solicita al backend las recetas marcadas como favoritas por el usuario
		El sistema muestra la lista de las recetas
Flujo de los eventos:	—	
Flujo alternativo (no tiene favoritos):	Acciones del actor	Respuesta del Sistema (Acción 2) El sistema solicita al backend las recetas marcadas como favoritas por el usuario
		El sistema devuelve un mensaje de que no tiene favoritos
Postcondiciones:	—	
Excepciones:	—	

Cuadro 4.4: Descripción del CU4 de “Listar Favoritos”.

ID Caso de Uso:	Añadir Favoritos	
Actor principal:	Usuario	
Actores secundarios:	—	
Breve descripción:	El usuario puede añadir una receta a su lista de favoritos	
Precondiciones:	Necesita haber creado una cuenta previamente y estar iniciado en ella	
	Acciones del actor	Respuesta del Sistema
	Véase CU11 “Mostrar Información de Receta”	
Flujo de los eventos:	El usuario selecciona el botón de favoritos	El sistema guarda en el backend la receta como favorita
Flujo alternativo (ya está en favoritos):	Acciones del actor (Acción 2) El usuario selecciona el botón de favorito	Respuesta del Sistema
		El sistema quita en el backend la receta como favorita
Postcondiciones:	—	
Excepciones:	—	

Cuadro 4.5: Descripción del CU5 de “Añadir Favoritos”.

4.2. ANÁLISIS

63

ID Caso de Uso:	Cerrar Sesión	
Actor principal:	Usuario	
Actores secundarios:		
Breve descripción:	El usuario puede cerrar la sesión	
Precondiciones:	Necesita haber creado una cuenta previamente y estar iniciado en ella	
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El usuario selecciona el botón de cerrar sesión	El sistema cierra la sesión mientras devuelve el formulario de inicio de sesión
Postcondiciones:	-	
Excepciones:	-	

Cuadro 4.6: Descripción del CU6 de “Cerrar Sesión”.

ID Caso de Uso:	Registrarse	
Actor principal:	Usuario	
Actores secundarios:		
Breve descripción:	El usuario puede crear una cuenta en la aplicación	
Precondiciones:	Necesita de un correo válido y no estar iniciado en sesión	
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El usuario selecciona navegar a la página de su cuenta	El sistema le devuelve un formulario para crear una cuenta
	El usuario inserta su correo y repite 2 veces la contraseña que quiere	El sistema comprueba que el correo y las contraseñas son válidos
Flujo alternativo (campos introducidos inválidos o incorrectos):	Acciones del actor (Acción 3) El usuario inserta su correo y repite 2 veces la contraseña que quiere	El sistema crea la cuenta mientras cambia el estado de sesión
		El sistema comprueba que el correo y las contraseñas son válidos
		El sistema devuelve un mensaje indicando el campo que impide la creación de la cuenta
Postcondiciones:	-	
Excepciones:	Cuando un correo ya está en uso	

Cuadro 4.7: Descripción del CU7 de “Registrarse”.

ID Caso de Uso:	Listar Recetas	
Actor principal:	Usuario	
Actores secundarios:		
Breve descripción:	El usuario puede solicitar visualizar las recetas disponibles	
Precondiciones:		
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El usuario selecciona navegar a la página de las recetas	El sistema solicita al backend las recetas hasta la fecha
		El sistema muestra la lista de recetas recibido
Postcondiciones:	-	
Excepciones:	-	

Cuadro 4.8: Descripción del CU8 de “Listar Recetas”.

ID Caso de Uso:	Listar Ranking de Recetas	
Actor principal:	Usuario	
Actores secundarios:		
Breve descripción:	El usuario puede solicitar visualizar las recetas mejores valoradas	
Precondiciones:		
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El usuario selecciona navegar a la página de las recetas	El sistema solicita al backend las recetas hasta la fecha
		El sistema muestra las 10 primeras recetas ordenadas por valoración
Postcondiciones:	-	
Excepciones:	-	

Cuadro 4.9: Descripción del CU9 de “Listar Ranking de Recetas”.

ID Caso de Uso:	Buscar Receta	
Actor principal:	Usuario	
Actores secundarios:	-	
Breve descripción:	El usuario puede buscar las recetas por nombre	
Precondiciones:	-	
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El usuario selecciona navegar a la página de búsqueda	-
	El sistema solicita al backend las recetas hasta la fecha	-
	El usuario inserta el nombre de la receta en el buscador	El sistema filtra las recetas según solicito el usuario
Postcondiciones:	-	
Excepciones:	-	

Cuadro 4.10: Descripción del CU10 de “Buscar Receta“.

ID Caso de Uso:	Mostrar Información de Receta	
Actor principal:	Usuario	
Actores secundarios:	-	
Breve descripción:	El usuario puede ver todos los detalles de la receta	
Precondiciones:	-	
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El usuario selecciona una receta de las posibles listas	El sistema devuelve página con todos los detalles de la receta
Postcondiciones:	-	
Excepciones:	-	

Cuadro 4.11: Descripción del CU11 de “Mostrar Información de Receta“.

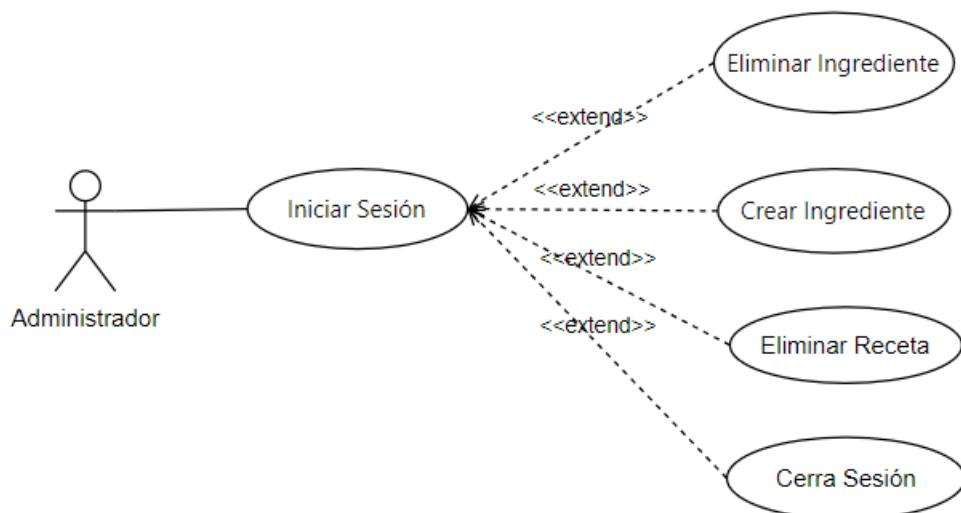


Figura 4.2: Diagrama de casos de uso de la página web.

4.2. ANÁLISIS

65

ID Caso de Uso:	Iniciar Sesión	
Actor principal:	Administrador	
Actores secundarios:	_____	
Breve descripción:	El administrador puede iniciar sesión con su cuenta	
Precondiciones:	Necesita tener una cuenta creada para la plataforma y no tener iniciada la sesión	
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El administrador selecciona el botón de inicio de sesión	El sistema le devuelve un formulario para iniciar una cuenta
	El administrador inserta el usuario y la contraseña	El sistema comprueba que existe una cuenta con esos atributos
		El sistema desbloquea funciones de la aplicación
Flujo alternativo (Cuenta inválida):	Acciones del actor	Respuesta del Sistema
	(Acción 3) El administrador inserta el usuario y la contraseña	El sistema comprueba que existe una cuenta con esos atributos
		El sistema devuelve un mensaje de cuenta no válida
Flujo alternativo (eliminar algún ingrediente):	Acciones del actor	Respuesta del Sistema
		(Acción 5) El sistema desbloquea funciones de la aplicación
	Véase CU2 “Eliminar Ingrediente”	
Flujo alternativo (crear un nuevo ingrediente):	Acciones del actor	Respuesta del Sistema
		(Acción 5) El sistema desbloquea funciones de la aplicación
	Véase CU3 “Crear Ingrediente”	
Flujo alternativo (Mostrar la lista de favoritos de la cuenta):	Acciones del actor	Respuesta del Sistema
		(Acción 5) El sistema desbloquea funciones de la aplicación
	Véase CU4 “Eliminar Receta”	
Flujo alternativo (Cerrar la sesión que está iniciada):	Acciones del actor	Respuesta del Sistema
		(Acción 5) El sistema desbloquea funciones de la aplicación
	Véase CU5 “Cerrar Sesión”	
Postcondiciones:	_____	
Excepciones:	Cuando el usuario esta iniciando sesión en una cuenta ya iniciada	

Cuadro 4.12: Descripción del CU1 de “Iniciar Sesión”.

ID Caso de Uso:	Eliminar Ingrediente	
Actor principal:	Administrador	
Actores secundarios:	_____	
Breve descripción:	El administrador puede borrar ingredientes de la aplicación	
Precondiciones:	Necesita estar iniciado en sesión como administrador	
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El administrador selecciona navegar a la página de gestión de ingredientes	El sistema le devuelve la página de gestión de ingredientes
	El administrador selecciona el ingrediente que quiere eliminar	El sistema envía un mensaje de borrar el ingrediente al backend
		El sistema devuelve un mensaje de acción realizada correctamente
Postcondiciones:	_____	
Excepciones:	_____	

Cuadro 4.13: Descripción del CU2 de “Eliminar Ingrediente”.

ID Caso de Uso:	Crear Ingrediente	
Actor principal:	Administrador	
Actores secundarios:	_____	
Breve descripción:	El administrador puede crear ingredientes en la aplicación	
Precondiciones:	Necesita estar iniciado en sesión como administrador	
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El administrador selecciona navegar a la página de gestión de ingredientes	El sistema le devuelve la página de gestión de ingredientes
	El administrador introduce la información del ingrediente	El sistema comprueba que el ingrediente es válido
		El sistema manda una petición al backend para crear el ingrediente
		El sistema devuelve un mensaje de acción realizada correctamente
Flujo alternativo (campos introducidos inválidos):	Acciones del actor	Respuesta del Sistema
	(Acción 3) El usuario introduce la información del ingrediente	El sistema comprueba que el ingrediente es válido
		El sistema devuelve un mensaje indicando el campo que impide la creación del ingrediente
Postcondiciones:	_____	
Excepciones:	_____	

Cuadro 4.14: Descripción del CU3 de “Crear Ingrediente”.

ID Caso de Uso:	Eliminar Receta	
Actor principal:	Administrador	
Actores secundarios:		
Breve descripción:	El administrador puede borrar recetas en la aplicación	
Precondiciones:	Necesita estar iniciado en sesión como administrador	
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El administrador selecciona navegar a la página de gestión de recetas	El sistema le devuelve la página de gestión de recetas
	El administrador selecciona la receta que quiere eliminar	El sistema envía un mensaje de borrar la receta al backend
		El sistema devuelve un mensaje de acción realizada correctamente
Postcondiciones:		
Excepciones:		

Cuadro 4.15: Descripción del CU4 de “Eliminar Receta”.

ID Caso de Uso:	Cerrar Sesión	
Actor principal:	Administrador	
Actores secundarios:		
Breve descripción:	El administrador puede cerrar la sesión	
Precondiciones:	Necesita tener una cuenta previamente y estar iniciado en ella	
Flujo de los eventos:	Acciones del actor	Respuesta del Sistema
	El administrador selecciona el botón de cerrar sesión	El sistema cierra la sesión mientras devuelve a la página de inicio
Postcondiciones:		
Excepciones:		

Cuadro 4.16: Descripción del CU5 de “Cerrar Sesión”.

4.2.2. Diagrama de estados

En esta subsección, normalmente nos adentraríamos en los diagramas de estado en el contexto de nuestro proyecto. Los diagramas de estado son una herramienta crucial para modelar el comportamiento dinámico de los elementos interactivos, proporcionando una representación visual de los diferentes estados en los que puede encontrarse un objeto o entidad, así como las transiciones entre dichos estados.

Sin embargo, dado que nuestra plataforma se centra en la gestión de recetas e ingredientes tanto en una aplicación móvil para usuarios como en una página web de administración, los estados dinámicos son bastante limitados. Básicamente, los estados relevantes para nuestros usuarios y administradores son "sesión iniciada" y "sesión no iniciada". Estas transiciones de estado son simples y no requieren una modelización compleja.

La principal funcionalidad de nuestra plataforma se basa en operaciones CRUD (Crear, Leer, Actualizar, Borrar) y la interacción con bases de datos para gestionar recetas e ingredientes, en lugar de un comportamiento dinámico y secuencial complejo que se encuentra en los videojuegos.

Por lo tanto, en el contexto de este proyecto, los diagramas de estado no son una herramienta necesaria para comprender o definir las reglas y mecánicas de la plataforma.

4.2.3. Diagrama de actividad

En esta última subsección del apartado de Análisis, desde la perspectiva de la Ingeniería del Software, presentamos una simulación de la prueba de funcionamiento de la plataforma tanto para la aplicación móvil como para la página web. Para ello, utilizamos un diagrama de actividades que representa el flujo de la mayoría de las funcionalidades del proyecto, brindando una visión completa de su ejecución.

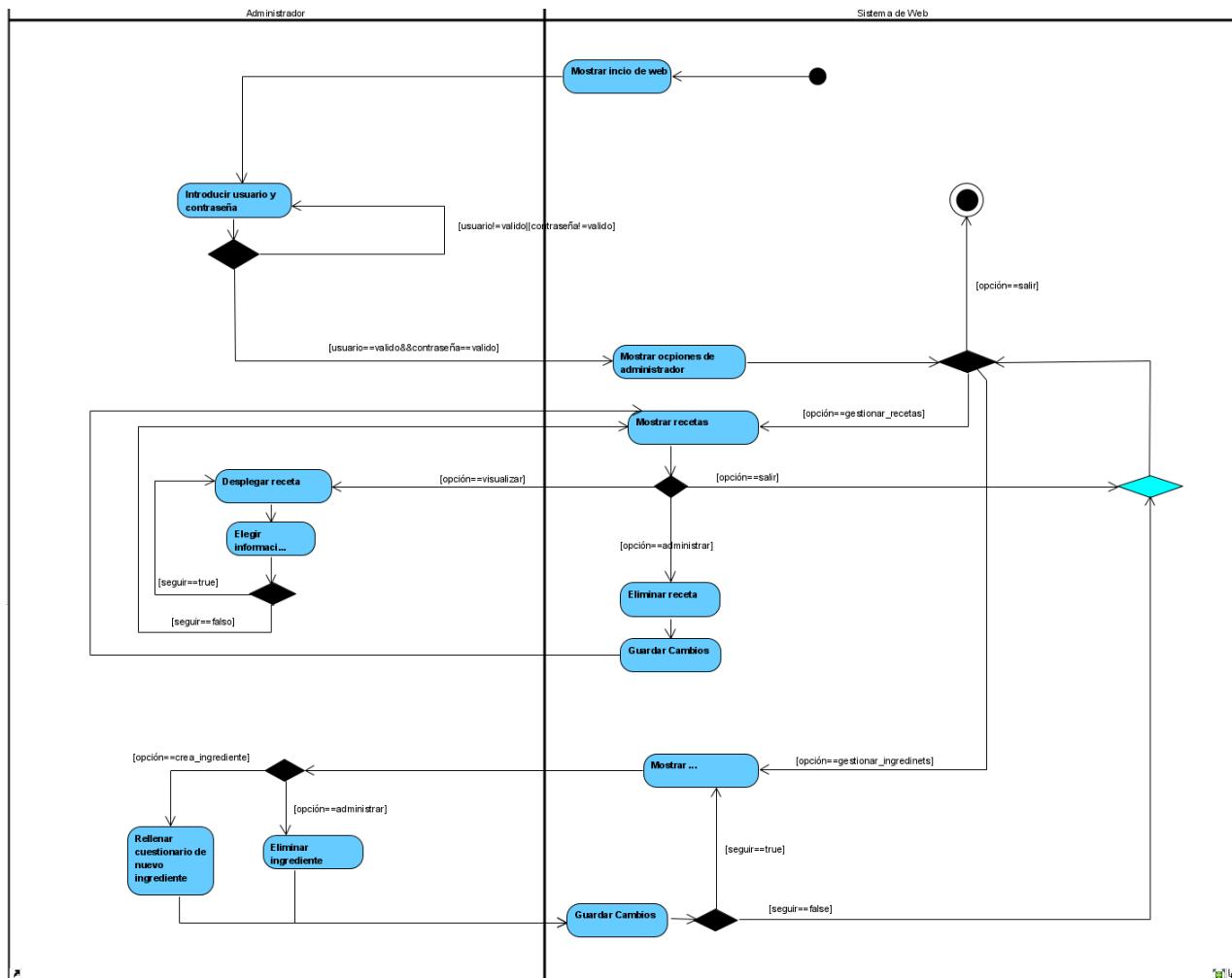


Figura 4.3: Diagrama de actividad del sistema de la aplicación web.

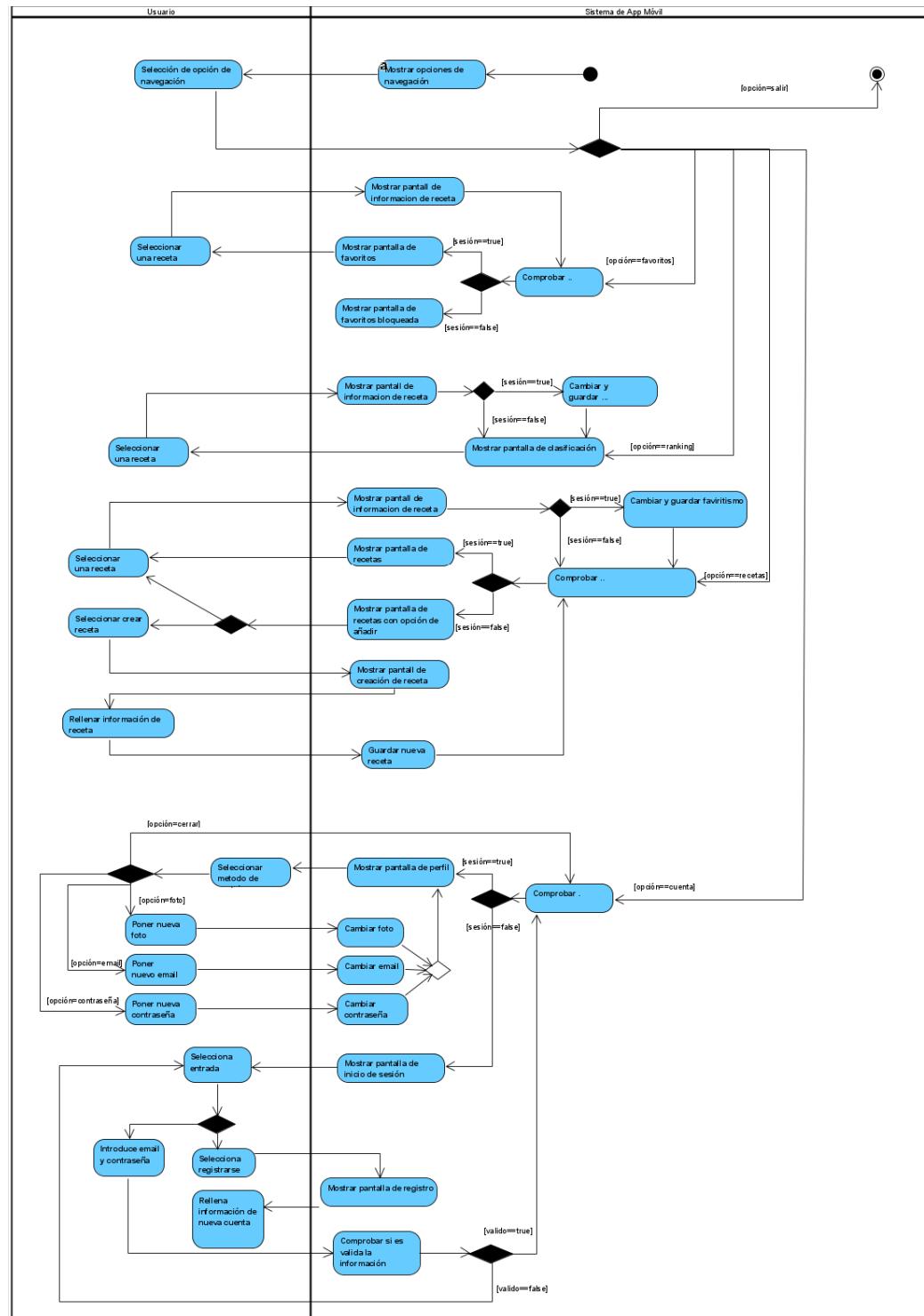


Figura 4.4: Diagrama de actividad de la aplicación móvil.

4.3. Diseño

En este apartado, se presenta el diseño del proyecto, abarcando aspectos como las interfaces de usuario tanto de la aplicación móvil como de la página web, el método de gestión de recetas e ingredientes, y las diferentes clases que conforman la base de datos del sistema en su totalidad.

Comenzaremos por explicar el diseño de la interfaz de usuario de la aplicación móvil, ya que desempeña un papel crucial para los usuarios finales. Para obtener más información detallada y comprender a fondo los detalles y la razón detrás de dicho diseño, se podrá observar en el punto ??.

A continuación, exploraremos el diseño de la interfaz de la página web destinada a los administradores, así como las consideraciones que se tuvieron en cuenta al diseñarla. Esta sección también incluirá una explicación sobre cómo los administradores pueden gestionar las recetas e ingredientes a través de la plataforma.

Además, examinaremos las diferentes clases que forman parte del sistema y su interacción para lograr el funcionamiento deseado. Esto incluirá el diseño de la base de datos y la estructura necesaria para soportar las funcionalidades tanto de la aplicación móvil como de la página web.

4.3.1. Diseño de interfaces de usuario de la aplicación móvil



Figura 4.5: Diagrama de UI del listado de recetas.

En la imagen se muestran las interfaces del listado de recetas, en ellas se muestran el panel de navegación inferior donde los iconos indican las diferentes páginas con las que un usuario puede interactuar en su móvil, aunque esta página tiene dos estilos diferentes. En el primero, donde el usuario no ha iniciado sesión, solo se muestra una lista de notas con recetas (incluyendo una imagen e información importante sobre cada una), con las que el usuario puede interactuar para visualizar más información y los pasos de las recetas. En el segundo estilo, donde el usuario ha iniciado sesión, se muestra un botón adicional que le permite crear más recetas para la plataforma.

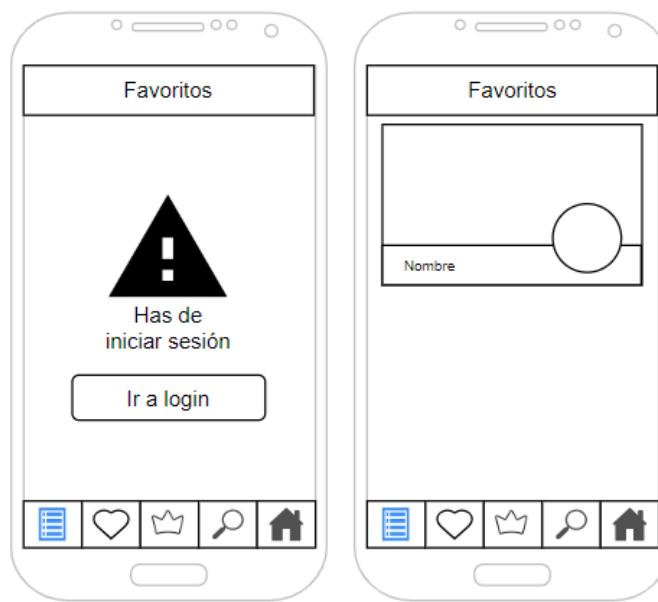


Figura 4.6: Diagrama de UI del listado de favoritos del usuario.

Esta página también presenta diferentes versiones dependiendo del estado de la sesión. En caso de que el usuario no se hubiera loggeado¹, mostrará un mensaje informativo donde le explicará que para acceder a esa función necesitará mostrar su cuenta y un botón con el que le mandará directamente a la pagina de iniciar sesión. Mientras que si ya hubiera iniciado sesión, se le mostraría una lista personalizado parecido al que teníamos con el listado de recetas, pero en este caso se muestra que son favoritas con un icono de favoritismo y con un acabado mas visual de la nota de la recetas para que sean mas reconocibles para el cliente.

¹En una cuenta es ingresar con credenciales para acceder a servicios personalizados y gestionar información personal en una plataforma.

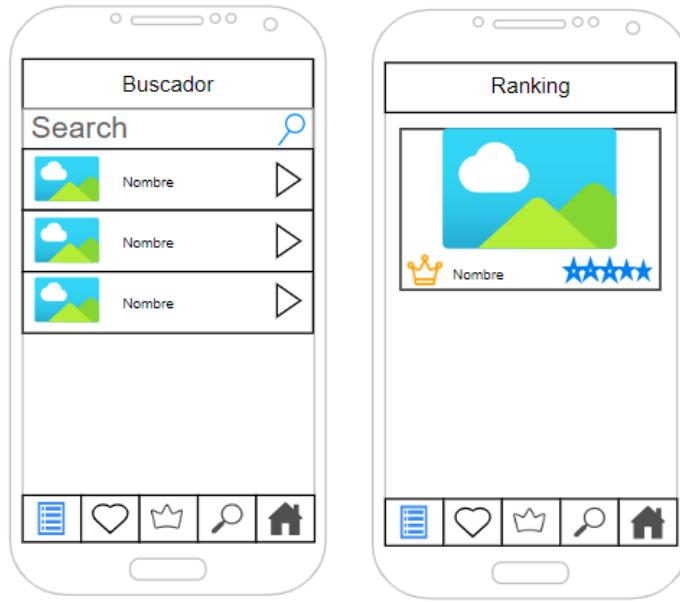


Figura 4.7: Diagrama de UI de *interfaz_{movil}favoritos. @*

En este panel se muestran dos páginas: el buscador de recetas y la clasificación de todas las recetas. En la primera, se encuentra un buscador y un listado simplificado de las recetas. En la segunda, se encontrarán las recetas clasificadas de mejor a peor con un listado parecido al de favoritos pero destacando las tres mejores con unos iconos únicos y en todas las recetas se estarán mostrando las valoraciones.

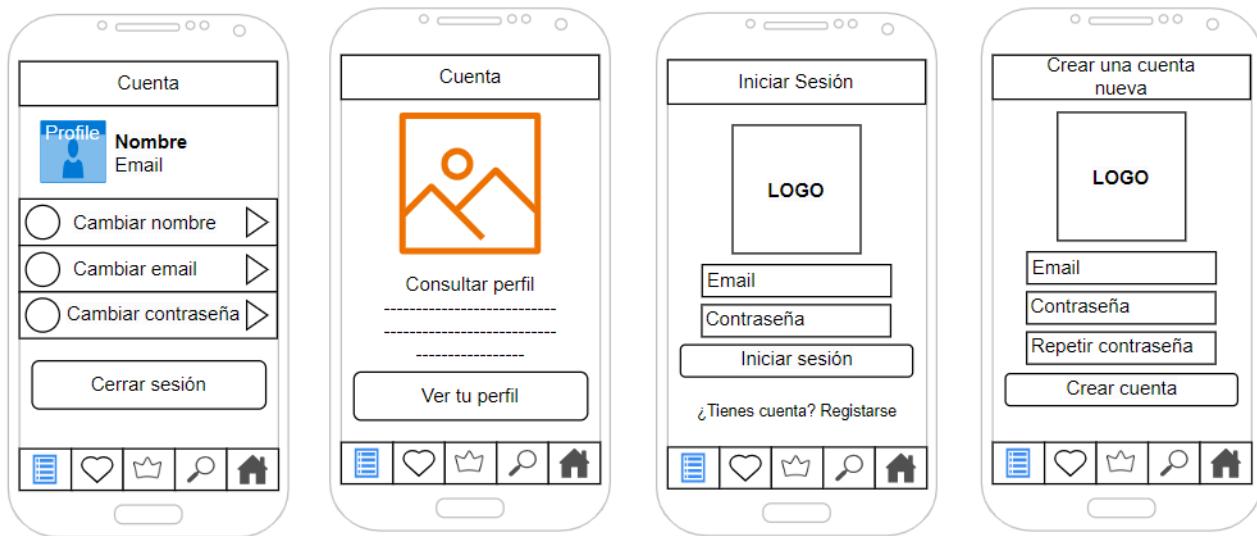


Figura 4.8: Diagrama de UI de gestión de la sesión.

En este panel se muestran las cuatro posibles visualizaciones de la página donde se

puede controlar la cuenta del usuario. La primera muestra las opciones que tiene el usuario tras iniciar sesión, donde puede cambiar cualquiera de los parámetro que quiera, hasta la propia foto de perfil. En la segunda se muestra una imagen y una explicación para mostrar como funciona esta parte de la aplicación para nuevos usuarios y un botón para poder desplazarse al inicio de sesión. El tercero es el propio formulario de inicio de sesión donde además se ve el icono de la plataforma y un enlace que lleva a nuestro usuario a la pantalla donde pueda crear una cuenta. Y por último, se muestra un formulario donde el usuario podrá crear una cuenta a partir de su correo y cuya acción también le iniciará de forma automática la sesión.



Figura 4.9: Diagrama de UI para crear una receta.

Esta página se despliega una vez que el usuario ha iniciado sesión. En ella, se presenta un formulario diseñado para facilitar al usuario la creación de nuevas recetas. Este formulario incluye campos para agregar pasos y una lista de ingredientes, ofreciendo una experiencia similar a la elaboración de una lista de compras o instrucciones detalladas para la preparación de platillos.

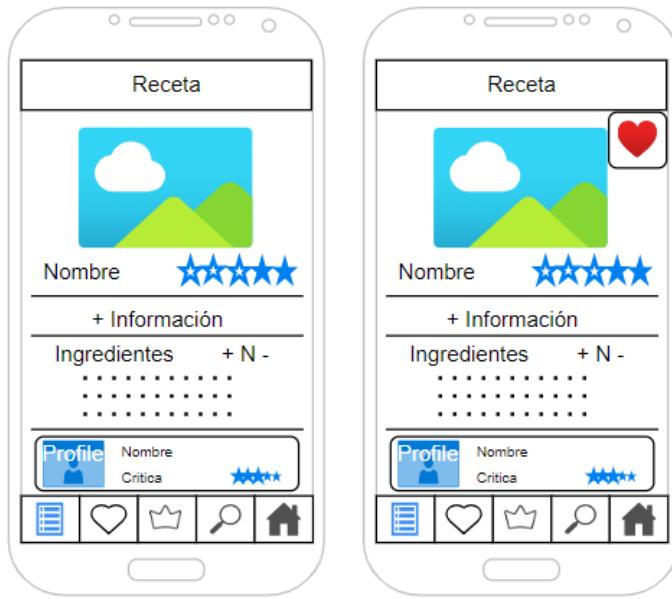


Figura 4.10: Diagrama de UI de Opciones.

En esta última pantalla de la aplicación móvil, se proporciona al usuario información completa sobre las recetas, incluidos los valores nutricionales y comentarios de otros usuarios. Si el usuario no ha iniciado sesión, no podrá dejar comentarios ni guardar la receta en favoritos, lo cual se indica con la ausencia de los respectivos botones.

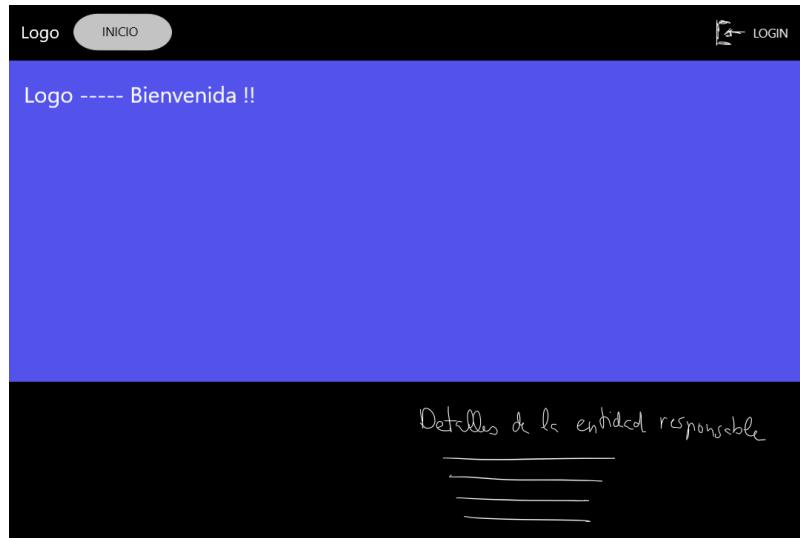


Figura 4.11: Diagrama de UI del inicio de la web sin iniciar sesión.

En esta interfaz de inicio, se muestra el navegador en la parte superior y los colores principales de los que constará nuestra página web. Esta servirá para dar la bienvenida al

administrador y mostrar los detalles de contacto de la empresa que tenga la propiedad de la plataforma



Figura 4.12: Diagrama de UI del inicio de la web .

Esta interfaz complementa a la anterior y en el se muestra que cuando nuestro usuario a iniciado sesión, en la navegación se le han desbloqueado a nuestro administrador 2 nuevas funcionalidades para poder gestionar la información de la plataforma.

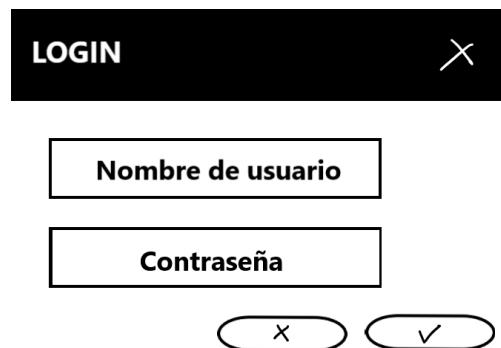


Figura 4.13: Diagrama de UI de inicio de sesión.

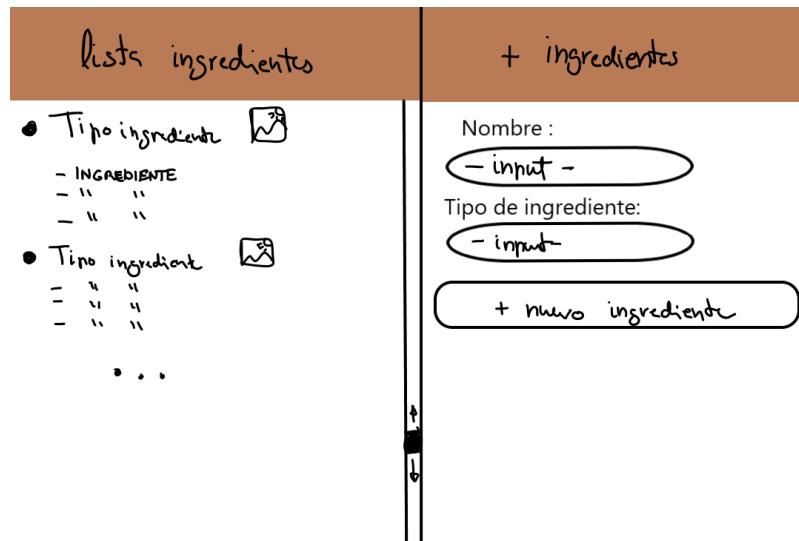


Figura 4.14: Diagrama de UI de gestión de ingredientes.

En esta interfaz, se muestran 2 ventanas: una para eliminar las ingredientes y otras donde se pueden generar nuevos ingredientes. En la izquierda se mostrara un listado de los ingredientes organizados por tipo de ingredientes y de los cuales se mostrara un botón para eliminar dichos ingredientes. Mientras que en la derecha, se mostrará un formulario donde el administrador elegira el nombre y seleccionara el tipo del nuevo ingrediente.

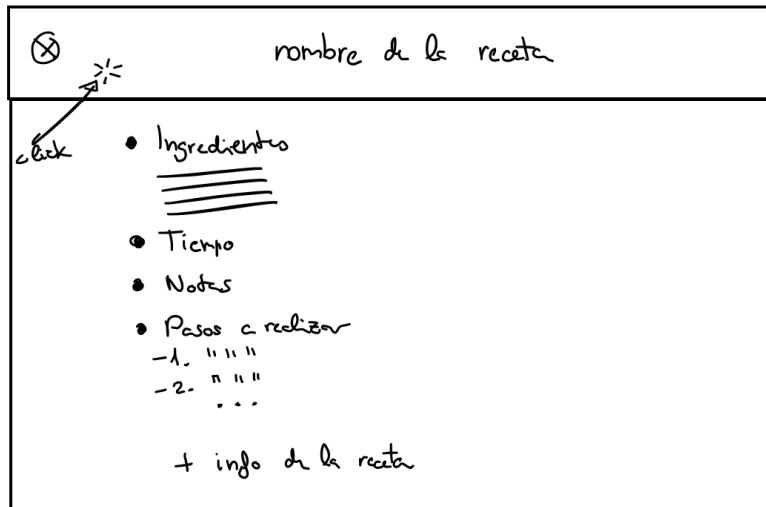


Figura 4.15: Diagrama de UI de gestión de recetas.

Y en esta ultima interfaz, se muestra la visualización de una de las recetas en la pagina de gestión de recetas cuando se despliega. En ella, se vera la informacion que ha introducido el cliente y los pasos listados, mientras en la esquina superior izquierda se

mostrar un botón con el que el administrador podrá considerar la recetas nociva para la plataforma y deshacerse de ella.

4.3.2. Diseño de base de datos

En el contexto de nuestro proyecto, es fundamental entender las diferentes formas de guardar y recuperar datos para asegurar la persistencia y consistencia de la información tanto en la aplicación móvil como en la página web de administración. Las principales formas de gestionar esta persistencia de datos son a través de bases de datos relacionales y no relacionales.

Bases de datos relacionales

Las bases de datos relacionales organizan la información en tablas que están relacionadas entre sí mediante claves primarias y foráneas. Este modelo es ideal para manejar datos estructurados y garantizar la integridad referencial. En nuestro proyecto, la aplicación web utilizará una base de datos relacional administrada a través de una API REST autogenerada con servlets. En esta base de datos se almacenará la información de los ingredientes.

La implementación de una base de datos relacional en la aplicación web sigue este procedimiento:

- **Creación de tablas:** Las tablas se diseñan para almacenar diferentes entidades, como ingredientes, con sus relaciones correspondientes.
- **Consultas SQL:** Se utilizan consultas SQL para realizar operaciones CRUD (Crear, Leer, Actualizar, Borrar) sobre las tablas.
- **API REST con servlets:** La API REST permite interactuar con la base de datos a través de endpoints definidos que manejan las solicitudes HTTP y ejecutan las consultas SQL.

Ejemplo de una consulta SQL en un servlet para recuperar ingredientes:

```
1 // Conexión a la base de datos
2 Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
3 // Consulta SQL
4 String sql = "SELECT * FROM ingredientes WHERE receta_id=?";
5 PreparedStatement pstmt = conn.prepareStatement(sql);
6 pstmt.setInt(1, recetaId);
7 ResultSet rs = pstmt.executeQuery();
8 // Procesar resultados
9 while (rs.next()) {
```

```

1|| } // Procesar cada ingrediente
11}

```

Código 4.1: Ejemplo de consulta SQL en un servlet

Bases de datos no relacionales

Las bases de datos no relacionales, también conocidas como NoSQL, almacenan datos en formatos más flexibles como documentos, gráficos o key-value pairs. Estas bases de datos son ideales para manejar grandes volúmenes de datos no estructurados y ofrecen escalabilidad horizontal. Para nuestra aplicación móvil, utilizaremos Firebase Firestore, una base de datos NoSQL que facilita el desarrollo móvil con sincronización en tiempo real y almacenamiento en la nube. En Firebase se almacenarán los datos del usuario, de las recetas y de la lista de favoritos de los usuarios.

La implementación de Firebase Firestore en la aplicación móvil sigue este procedimiento:

- **Estructura de documentos:** Los datos se organizan en colecciones de documentos, donde cada documento contiene pares clave-valor.
- **Operaciones con Firestore:** Se utilizan métodos proporcionados por el SDK de Firebase para realizar operaciones CRUD de manera eficiente.
- **Sincronización en tiempo real:** Firebase sincroniza los datos en tiempo real entre todos los dispositivos conectados, manteniendo la consistencia de los datos.

Ejemplo de guardar una receta en Firestore:

```

1// Crear una instancia de Firestore
2FirebaseFirestore db = FirebaseFirestore.getInstance();
3// Crear un nuevo documento de receta
4Map<String , Object> receta = new HashMap<>();
5receta.put("nombre", "Paella");
6receta.put("ingredientes", Arrays.asList("Arroz", "Mariscos", "Azafrán"));
7receta.put("instrucciones", "Cocinar a fuego medio");
8// Guardar el documento en la colección "recetas"
9db.collection("recetas").add(receta)
    .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
11        @Override
12        public void onSuccess(DocumentReference documentReference) {
13            Log.d(TAG, "Receta añadida con ID: " +
14                documentReference.getId());
15        }
16    })
    .addOnFailureListener(new OnFailureListener() {

```

```

17|     @Override
18|     public void onFailure(@NonNull Exception e) {
19|         Log.w(TAG, "Error al adiando receta", e);
20|     }
21| });

```

Código 4.2: Ejemplo de guardar una receta en Firestore

En resumen, nuestra plataforma utiliza una combinación de bases de datos relacionales y no relacionales para gestionar la información de manera eficiente y flexible. La aplicación móvil aprovecha las capacidades de Firebase Firestore para una sincronización en tiempo real y almacenamiento flexible de datos de usuarios, recetas y listas de favoritos, mientras que la aplicación web se beneficia de la estructura sólida y consultas robustas de una base de datos relacional a través de una API REST para la gestión de ingredientes.

A continuación, mostramos el diagrama del modelo de datos utilizado en ambos contextos.

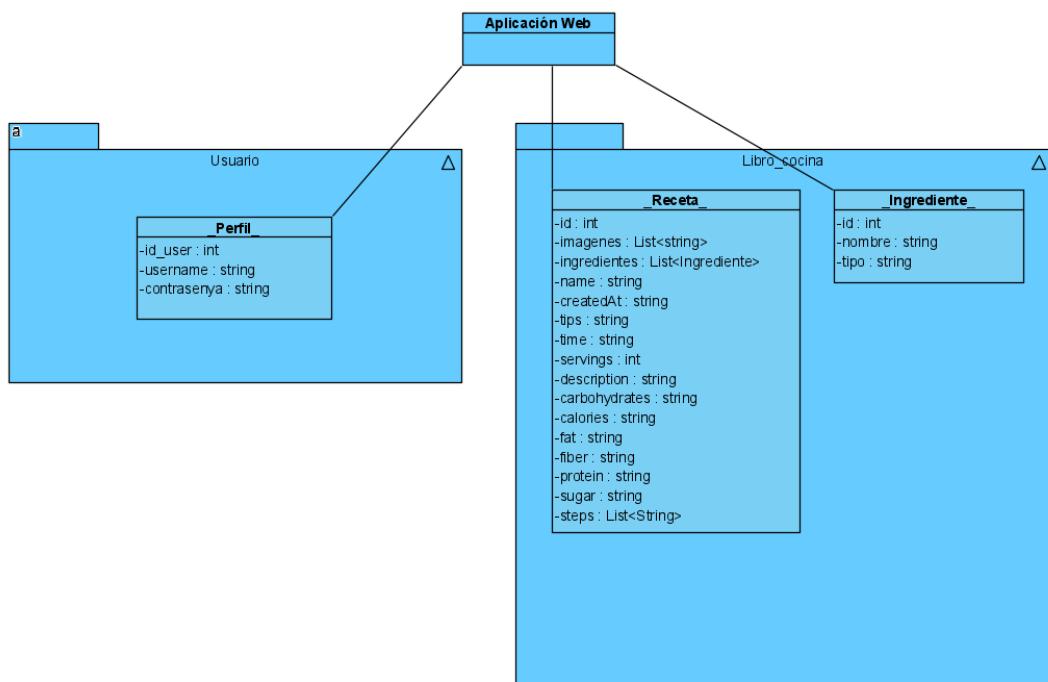


Figura 4.16: Diagrama del modelo de datos para la aplicación de web.

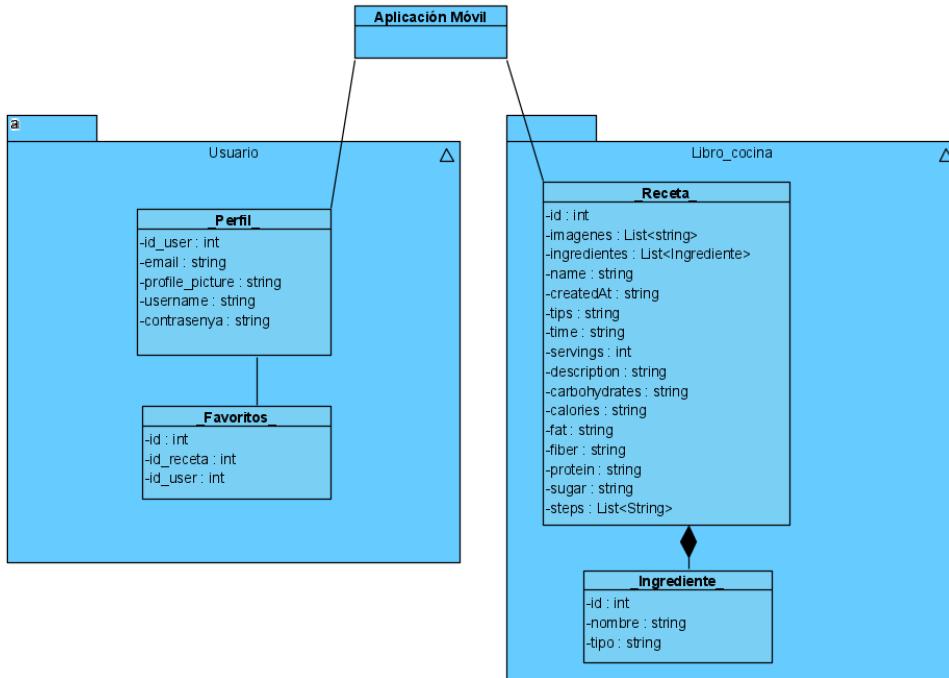


Figura 4.17: Diagrama del modelo de datos para la aplicación de móvil.

4.3.3. Diagrama de clases

Después de realizar el análisis de los datos que se utilizarían en nuestro proyecto y cómo se guardarían dentro de nuestra plataforma, identificamos las clases necesarias para su funcionamiento y las dependencias entre ellas. Estas son las clases que hemos definido:

Aplicación Móvil

- **AplicaciónMóvil:** Esta clase principal agrupa la mayoría de las clases de la aplicación móvil y maneja la lógica general del funcionamiento de la aplicación. Es responsable de iniciar y coordinar las diferentes pantallas y gestionar la navegación entre ellas.
- **Ingrediente:** Esta clase almacena la información relacionada con los ingredientes utilizados en las recetas, como el nombre, la cantidad y las unidades de medida.
- **Receta:** En esta clase se almacena la información completa de una receta, incluyendo el nombre, los ingredientes, los pasos de preparación y la información nutricional.

- **PantallaCrearReceta:** Esta clase maneja la pantalla donde los usuarios pueden crear nuevas recetas, incluyendo la adición de ingredientes y pasos.
- **PantallaInfoReceta:** Esta clase gestiona la pantalla que muestra la información detallada de una receta, como los ingredientes y los pasos de preparación.
- **PantallaListaRecetas:** En esta clase se maneja la pantalla que muestra la lista de todas las recetas disponibles en la aplicación.
- **PantallaRanking:** Esta clase administra la pantalla donde se visualizan las recetas mejor valoradas, organizadas de mayor a menor calificación.
- **PantallaBusqueda:** Esta clase administra la pantalla donde se puede buscar por nombre la receta de la plataforma.
- **PantallaListaFavoritos:** Esta clase gestiona la pantalla que muestra la lista de recetas favoritas del usuario, permitiendo una rápida visualización y acceso.
- **PantallaCuenta:** En esta clase se maneja la pantalla donde el usuario puede ver y editar su información de cuenta.
- **PantallaLoginMóvil:** Esta clase administra la pantalla de inicio de sesión para los usuarios de la aplicación móvil.
- **PantallaRegistro:** En esta clase se maneja la pantalla de registro para nuevos usuarios en la aplicación móvil.
- **UsuarioMovil:** Esta clase almacena la información del usuario de la aplicación móvil, como sus datos de inicio de sesión, recetas creadas y favoritas.
- **Favoritos:** Esta clase almacena la información de las recetas que el usuario ha marcado como favoritas, facilitando el acceso rápido y la gestión de estas recetas.
- **Comentario:** En esta clase se almacenan los comentarios que los usuarios dejan en las recetas, incluyendo el texto del comentario, la valoración, el autor y la fecha.

Aplicación Web

- **Ingrediente:** Similar a la aplicación móvil, esta clase almacena la información de los ingredientes, incluyendo nombre, cantidad y unidades de medida.
- **Receta:** Esta clase almacena la información completa de una receta, incluyendo los ingredientes, pasos de preparación y datos nutricionales.
- **AplicaciónWeb:** Esta clase principal agrupa las demás clases de la aplicación web y maneja la lógica general del funcionamiento de la plataforma. Coordina la navegación y la interacción entre las diferentes pantallas.
- **PantallaInicio:** Esta clase maneja la pantalla de inicio de la aplicación web, proporcionando un punto de entrada a las funcionalidades principales.
- **PantallaLoginWeb:** En esta clase se administra la pantalla de inicio de sesión para los usuarios de la aplicación web.
- **PantallaGestiónRecetas:** Esta clase gestiona la pantalla donde los administradores pueden ver, editar y eliminar recetas en la plataforma.
- **PantallaGestiónIngredientes:** En esta clase se maneja la pantalla de gestión de ingredientes, permitiendo a los administradores agregar, editar o eliminar ingredientes.
- **UsuarioWeb:** Esta clase almacena la información del usuario de la aplicación web, incluyendo sus datos de inicio de sesión y permisos de administración.

Estas clases y sus interacciones nos permiten mantener una estructura organizada y eficiente tanto en la aplicación móvil como en la página web de administración, facilitando la gestión de recetas, ingredientes y usuarios en nuestra plataforma.

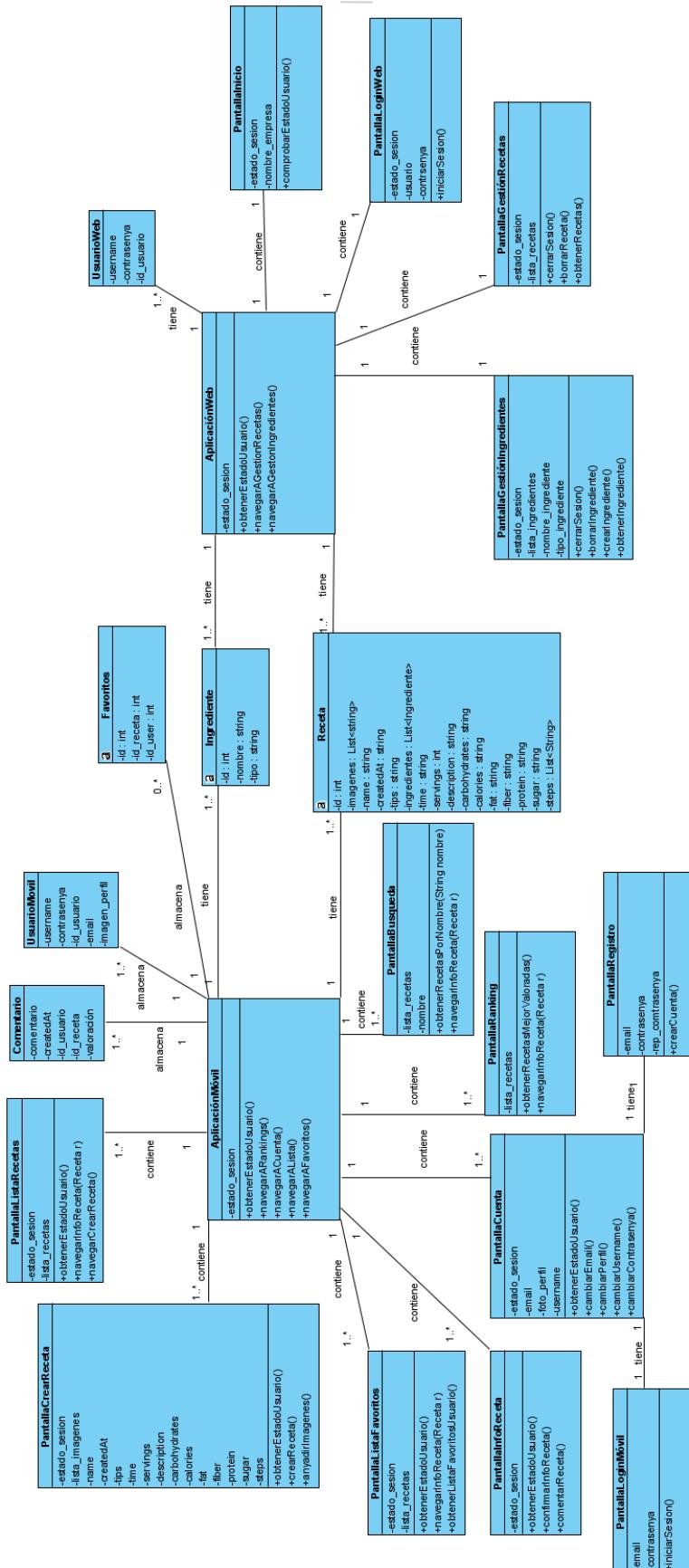


Figura 4.18: Diagrama de clases de la plataforma.

Aplicación Móvil

Descripción	
Gestiona la lógica y navegación de la aplicación móvil	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
Operaciones	
obtenerEstadoUsuario	Operación para obtener los datos de estado del usuario
navegarARanking	Operación para navegar hasta la pantalla de Ranking
navegarACuenta	Operación para navegar hasta la pantalla de Cuenta
navegarALista	Operación para navegar hasta la pantalla de Lista de Recetas
navegarAFavoritos	Operación para navegar hasta la pantalla de Favoritos
navegarABuscador	Operación para navegar hasta la pantalla de Busqueda

Cuadro 4.17: Descripción de la clase “Aplicación Móvil“.

Ingrediente

Descripción	
Almacena detalles de los ingredientes	
Atributos	
id	Variable que guarda el identificador del ingrediente
nombre	Variable que guarda el nombre del ingrediente
tipo	Variable que guarda el tipo de ingrediente

Cuadro 4.18: Descripción de la clase “Ingrediente“.

Receta

Descripción	
Contiene toda la información de una receta	
Atributos	
id	Variable que guarda el identificador de la receta
imagenes	Variable que guarda la lista de imágenes de la receta
name	Variable que guarda el nombre de la receta
createdAt	Variable que guarda la fecha de publicación
tips	Variable que guarda los consejos de la receta
ingredientes	Variable que guarda la lista de ingredientes y su cantidad
time	Variable que guarda el tiempo de cocinado
servings	Variable que guarda la cantidad de porciones que se consigue
description	Variable que guarda la descripción de la receta
carbohydrates	Variable que guarda la cantidad de carbohidratos
calories	Variable que guarda la cantidad de calorías
fat	Variable que guarda la cantidad de grasas saturadas
fiber	Variable que guarda la cantidad de fibra
protein	Variable que guarda la cantidad de proteína
sugar	Variable que guarda la cantidad de azúcares
steps	Variable que guarda las descripciones de cada paso de la receta

Cuadro 4.19: Descripción de la clase “Receta“.

PantallaCrearReceta

Descripción	
Maneja la pantalla para crear recetas	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
id	Variable que guarda el identificador de la receta
imagenes	Variable que guarda la lista de imágenes de la receta
name	Variable que guarda el nombre de la receta
createdAt	Variable que guarda la fecha de publicación
tips	Variable que guarda los consejos de la receta
ingredientes	Variable que guarda la lista de ingredientes y su cantidad
time	Variable que guarda el tiempo de cocinado
servings	Variable que guarda la cantidad de porciones que se consigue
description	Variable que guarda la descripción de la receta
carbohydrates	Variable que guarda la cantidad de carbohidratos
calories	Variable que guarda la cantidad de calorías
fat	Variable que guarda la cantidad de grasas saturadas
fiber	Variable que guarda la cantidad de fibra
protein	Variable que guarda la cantidad de proteína
sugar	Variable que guarda la cantidad de azúcares
steps	Variable que guarda las descripciones de cada paso de la receta
Operaciones	
obtenerEstadoUsuario	Operación para obtener los datos de estado del usuario
crearReceta	Operación para añadir una nueva receta a la base de datos
anyadirImagenes	Operación para acceder a la galería y obtener fotos para la receta

Cuadro 4.20: Descripción de la clase “PantallaCrearReceta“.

PantallaInfoReceta

Descripción	
Muestra información detallada de una receta	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
Operaciones	
obtenerEstadoUsuario	Operación para obtener los datos de estado del usuario
confirmaInfoReceta	Operación para confirmar que la receta navegada existe y obtener su información
comentarReceta	Operación para crear un comentario sobre una receta

Cuadro 4.21: Descripción de la clase “PantallaInfoReceta“.

PantallaListaRecetas

Descripción	
Gestiona la lista de recetas disponibles	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
lista_recetas	Variable que almacena la lista de recetas de la plataforma
Operaciones	
obtenerEstadoUsuario	Operación para obtener los datos de estado del usuario
navegarInfoReceta	Operación para navegar a la pantalla de información de la receta que se argumente
navegarCrearReceta	Operación para navegar a la pantalla de Crear Receta

Cuadro 4.22: Descripción de la clase “PantallaListaRecetas“.

PantallaRanking

Descripción	
Muestra recetas mejor valoradas	
Atributos	
lista_recetas	Variable que almacena la lista de recetas de la plataforma
Operaciones	
obtenerRecetasMejorValoradas	Operación para obtener la lista de recetas mejor valoradas en la plataforma ordenadas
navegarInfoReceta	Operación para navegar a la pantalla de información de la receta que se argumente

Cuadro 4.23: Descripción de la clase “PantallaRanking“.

PantallaBusqueda

Descripción	
Busca la receta que se haya especificado en el buscador sino te muestra todas las recetas	
Atributos	
lista_recetas	Variable que almacena la lista de recetas de la plataforma
nombre	Variable que almacena el nombre de la receta a buscar
Operaciones	
obtenerRecetasPorNombre	Operación para obtener la lista de recetas que comparten parte o completamente el nombre del buscador
navegarInfoReceta	Operación para navegar a la pantalla de información de la receta que se argumente

Cuadro 4.24: Descripción de la clase “PantallaRanking“.

PantallaListaFavoritos

Descripción	
Administra la lista de recetas favoritas	
Atributos	
lista_recetas	Variable que almacena la lista de recetas de la plataforma
estado_sesion	Variable que guarda el estado del usuario conectado
Operaciones	
obtenerEstadoUsuario	Operación para obtener los datos de estado del usuario
obtenerListaFavoritosUsuario	Operación para obtener la lista de recetas favoritas del usuario que inició sesión
navegarInfoReceta	Operación para navegar a la pantalla de información de la receta que se argumente

Cuadro 4.25: Descripción de la clase “PantallaListaFavoritos“.

PantallaCuenta

Descripción	
Permite ver y editar la información del usuario	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
email	Variable que guarda el email del usuario loggeado
foto_perfil	Variable que guarda la foto de perfil del usuario loggeado
username	Variable que guarda el nombre de usuario loggeado
Operaciones	
obtenerEstadoUsuario	Operación para obtener los datos de estado del usuario
cambiarEmail	Operación para cambiar el email del usuario loggeado
cambiarPerfil	Operación para cambiar la imagen de perfil del usuario loggeado
cambiarUsername	Operación para cambiar el nombre de usuario loggeado
cambiarContrasenya	Operación para cambiar la contraseña del usuario loggeado

Cuadro 4.26: Descripción de la clase “PantallaCuenta“.

PantallaLoginMóvil

Descripción	
Administra la pantalla de inicio de sesión en la aplicación móvil	
Atributos	
email	Variable que guarda el email del usuario
contrasenya	Variable que guarda la contraseña del usuario
Operaciones	
iniciarSesion	Operación para iniciar la cuenta del usuario que rellenó el formulario

Cuadro 4.27: Descripción de la clase “PantallaLoginMóvil“.

PantallaRegistro

Descripción	
Maneja la pantalla de registro de nuevos usuarios en la aplicación móvil	
Atributos	
email	Variable que guarda el email del usuario
contrasenya	Variable que guarda la contraseña del usuario
rep_contraseña	Variable que verifica que la contraseña sea la correcta
Operaciones	
crearCuenta	Operación para crear la cuenta del usuario que llenó el formulario e iniciar la sesión

Cuadro 4.28: Descripción de la clase “PantallaRegistro“.

UsuarioMovil

Descripción	
Almacena información del usuario cliente de la aplicación móvil	
Atributos	
email	Variable que guarda el email del usuario
contrasenya	Variable que guarda la contraseña del usuario
id_usuario	Variable que guarda el identificador del usuario
username	Variable que guarda el nombre del usuario
imagen_perfil	Variable que guarda la imagen de perfil del usuario

Cuadro 4.29: Descripción de la clase “UsuarioMovil“.

Favoritos

Descripción	
Almacena las recetas favoritas de los usuarios	
Atributos	
id	Variable que guarda el identificador del favorito
id_usuario	Variable que guarda el identificador del usuario
id_receta	Variable que guarda el identificador de la receta

Cuadro 4.30: Descripción de la clase “Favoritos“.

Comentario

Descripción	
Contiene comentarios de usuarios sobre las recetas	
Atributos	
comentario	Variable que guarda el texto del comentario
id_usuario	Variable que guarda el identificador del usuario
id_receta	Variable que guarda el identificador de la receta
createdAt	Variable que guarda la fecha de publicación del comentario
valoración	Variable que guarda la cantidad de estrellas de la receta

Cuadro 4.31: Descripción de la clase “Comentario”.

AplicaciónWeb

Descripción	
Gestiona la lógica y navegación de la aplicación web	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
Operaciones	
obtenerEstadoUsuario	Operación para obtener los datos de estado del usuario
navegarAGestionRecetas	Operación para navegar hasta la pantalla de gestión de las recetas
navegarAGestionIngredientes	Operación para navegar hasta la pantalla de gestión de los ingredientes

Cuadro 4.32: Descripción de la clase “AplicaciónWeb”.

PantallaInicio

Descripción	
Maneja la pantalla de inicio de la web	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
nombre_empresa	Variable que guarda el nombre de la empresa que debe saludar
Operaciones	
comprobarEstadoUsuario	Operación para obtener los datos de estado del usuario conectado

Cuadro 4.33: Descripción de la clase “PantallaInicio”.

PantallaLoginWeb

Descripción	
Administra la pantalla de inicio de sesión web	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
usuario	Variable que guarda el nombre del usuario a loggearse
contrasenya	Variable que guarda la contraseña del usuario a loggearse
Operaciones	
iniciarSesión	Operación para iniciar sesión en una cuenta admin

Cuadro 4.34: Descripción de la clase “PantallaLoginWeb“.

PantallaGestionRecetas

Descripción	
Gestiona la edición de recetas	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
lista_recetas	Variable que guarda la lista de recetas de la plataforma
Operaciones	
cerrarSesión	Operación para cerrar la sesión iniciada
borrarReceta	Operación para borrar la información de una receta
obtenerRecetas	Operación para obtener la lista de recetas de la base de datos

Cuadro 4.35: Descripción de la clase “PantallaGestionRecetas“.

PantallaGestionIngredientes

Descripción	
Administra la gestión de ingredientes	
Atributos	
estado_sesion	Variable que guarda el estado del usuario conectado
lista_ingredientes	Variable que guarda la lista de ingredientes de la plataforma
nombre_ingrediente	Variable que guarda el nombre del ingrediente a crear
tipo_ingrediente	Variable que guarda el tipo del ingrediente a crear
Operaciones	
cerrarSesión	Operación para cerrar la sesión iniciada
borrarIngrediente	Operación para borrar la información de un ingrediente
obtenerIngredientes	Operación para obtener la lista de ingredientes de la base de datos
crearIngrediente	Operación para añadir un nuevo ingrediente a la base de datos

Cuadro 4.36: Descripción de la clase “PantallaGestionIngredientes“.

UsuarioWeb

Descripción	
Almacena información del usuario web	
Atributos	
username	Variable que guarda el nombre del administrador
id_usuario	Variable que guarda el identificador del administrador
contrasenya	Variable que guarda la contraseña del administrador

Cuadro 4.37: Descripción de la clase “UsuarioWeb“.

4.3.4. Diagramas de secuencia

Después de revisar las descripciones de las clases que forman parte de nuestro proyecto de gestión de recetas e ingredientes, presentaremos ahora, para cada uno de los casos de uso mencionados anteriormente, un diagrama de secuencia que representa las principales acciones que el usuario puede realizar en nuestra aplicación móvil y página web.

Cabe destacar que todas estas acciones se centran tanto en la plataforma móvil como en la web, por lo que al explicar las interacciones del usuario con el sistema, tendremos en cuenta estos factores y cómo el sistema responderá en consecuencia.

A continuación, se muestran los diagramas de secuencia que ilustran las interacciones entre el usuario y el sistema en diferentes escenarios.

Para el caso de uso “Iniciar Sesión“ (Móvil):

La secuencia de Ínicio de Sesión.^{en} este caso es para la aplicación móvil, esta se puede adaptar a la estructura del diagrama de flujo de la imagen de la siguiente manera:

- Al utilizar “iniciarSesion(nombre, contrasena)“, se requiere que el usuario introduzca su nombre de usuario y contraseña correspondientes.
- Al iniciar sesión puede utilizar “opciones()“, el usuario puede elegir entre las siguientes acciones:
 - **Modificar perfil:** El usuario puede modificar su perfil, como su imagen de usuario, email o contraseña.
 - Al utilizar “modificarElemento(caracteristica, nuevoValor)“, el usuario introduce el nuevo valor del elemento que desea modificar y el sistema actualiza la información del perfil del usuario.
 - **Acceder a recursos:** El usuario puede acceder a los recursos del sistema, como sus recetas favoritas o las recetas creadas por otros usuarios.
 - Al utilizar “nuevaReceta(receta)“, el usuario puede añadir una nueva receta a la base de datos.
 - Al utilizar “obtenerFavoritos()“, el sistema recupera la lista de recetas favoritas del usuario y las muestra en el frontend.
 - Al utilizar “añadirFavorito(receta)“, el usuario puede añadir una receta a su lista de favoritos.
 - **Cerrar sesión:** El sistema elimina la información de la cuenta del usuario de la memoria del frontend, envía un mensaje para advertir al backend de la acción y muestra la pantalla de inicio de sesión.

- Al utilizar “cerrarSesion()“, el sistema cierra la sesión del usuario.

Las demás opciones se detallan claramente en el diagrama o en las referencias correspondientes.

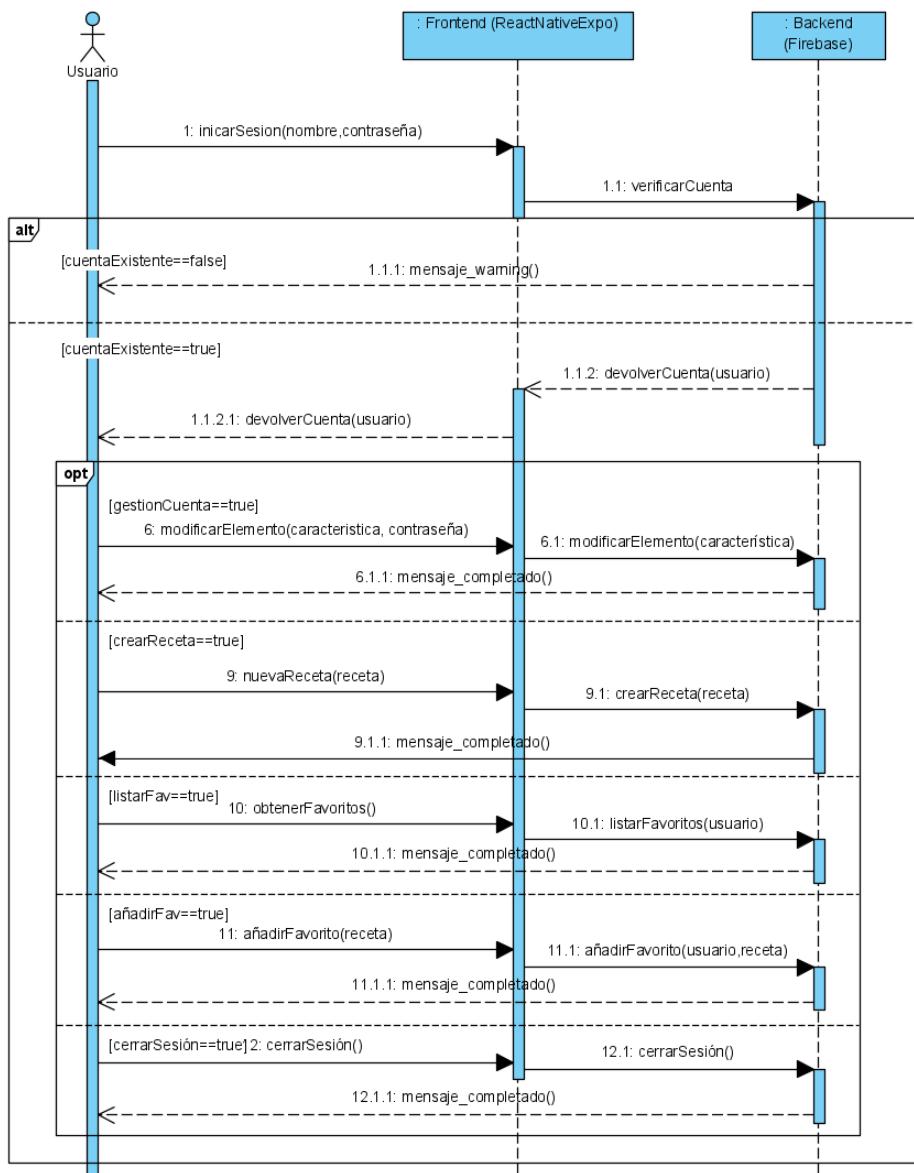


Figura 4.19: Diagrama de secuencia de “Iniciar Sesión“en Móvil.

Para el caso de uso “Registrarse“:

Este diagrama describe la forma en la que el usuario puede registrarse en la aplicación móvil. El usuario tiene la posibilidad de ingresar a la sección de registro para crear una

nueva cuenta. Una vez dentro de esta sección, el usuario debe proporcionar su correo electrónico, contraseña y repetir la contraseña para validación. Si todos los datos son correctos y la cuenta es válida, el registro se completa exitosamente y el usuario puede iniciar sesión.

El diagrama de secuencia correspondiente ilustrará de manera más detallada las interacciones y pasos necesarios para realizar esta acción en la aplicación móvil.

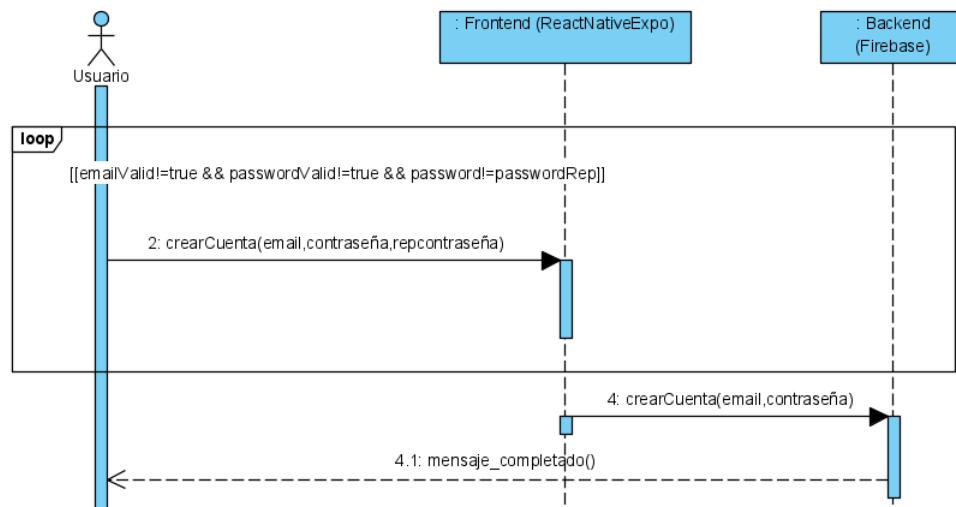


Figura 4.20: Diagrama de secuencia de “Registrarse”.

Para el caso de uso “Listar Recetas“:

Este diagrama describe la forma en la que el usuario puede acceder a la lista de recetas disponibles en la aplicación. El usuario tiene la posibilidad de ingresar a la sección de recetas para ver todas las recetas disponibles en la plataforma. Una vez dentro de esta sección, el usuario puede explorar las diferentes recetas, ver sus detalles, y posiblemente filtrarlas según ciertos criterios.

El diagrama de secuencia correspondiente ilustrará de manera más detallada las interacciones y pasos necesarios para realizar esta acción en la aplicación.

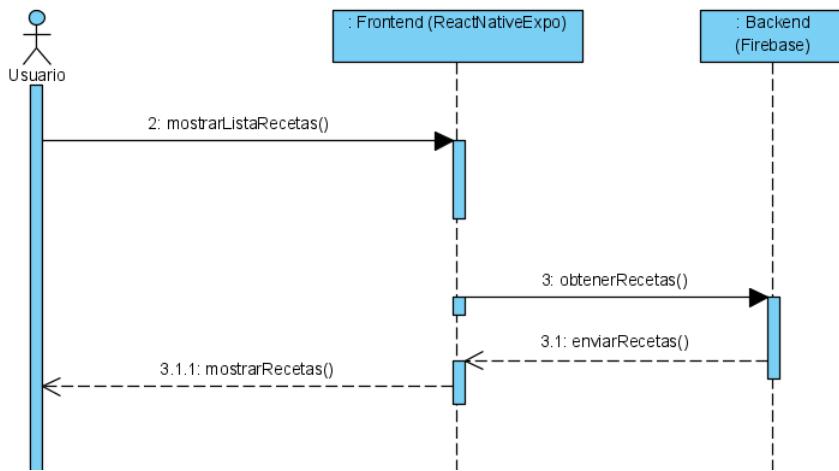


Figura 4.21: Diagrama de secuencia de “Listar Recetas“.

Para el caso de uso “Listar Ranking de Recetas“:

Este diagrama describe la forma en la que el usuario puede acceder al ranking de las recetas mejor valoradas en la aplicación. El usuario tiene la posibilidad de ingresar a la sección de ranking para ver las recetas que han obtenido las mejores valoraciones de otros usuarios. Una vez dentro de esta sección, el usuario puede explorar las recetas mejor valoradas y ver detalles adicionales sobre cada una de ellas.

El diagrama de secuencia correspondiente ilustrará de manera más detallada las interacciones y pasos necesarios para realizar esta acción en la aplicación.

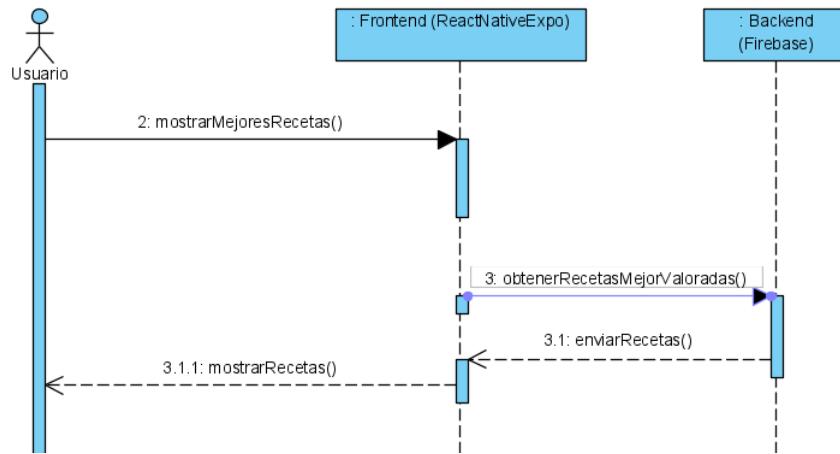


Figura 4.22: Diagrama de secuencia de “Listar Ranking de Recetas“.

Para el caso de uso “Buscar Receta“:

Este diagrama describe la forma en la que el usuario puede buscar una receta específica en la aplicación en base a su nombre. Una vez que el usuario ingresa los criterios de búsqueda, el sistema mostrará una lista de recetas que coinciden con esos criterios, permitiendo al usuario explorar y ver detalles adicionales sobre cada una de ellas.

El diagrama de secuencia correspondiente ilustrará de manera más detallada las interacciones y pasos necesarios para realizar esta acción en la aplicación.

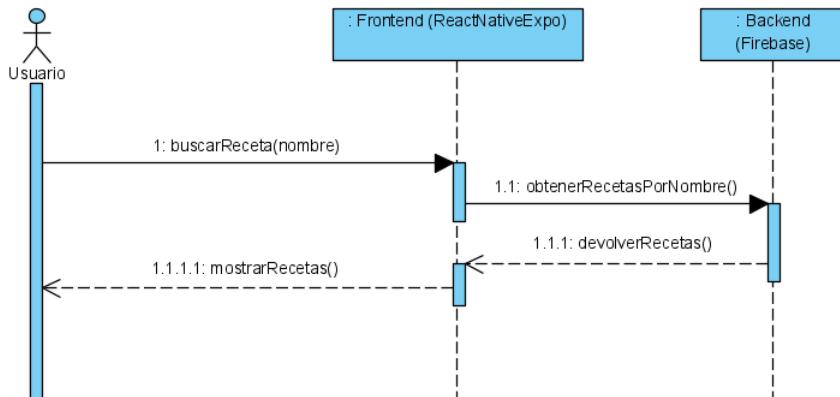


Figura 4.23: Diagrama de secuencia de “Buscar Receta“.

Para el caso de uso “Mostrar Información Receta“:

Este diagrama describe la forma en la que el usuario puede acceder a la información detallada de una receta en la aplicación. El usuario tiene la posibilidad de seleccionar una

receta específica desde la lista de recetas disponibles. Una vez que el usuario selecciona una receta, el sistema mostrará información detallada sobre la misma, incluyendo ingredientes, pasos de preparación, tiempo de cocción, valoraciones y comentarios de otros usuarios.

El diagrama de secuencia correspondiente ilustrará de manera más detallada las interacciones y pasos necesarios para realizar esta acción en la aplicación.

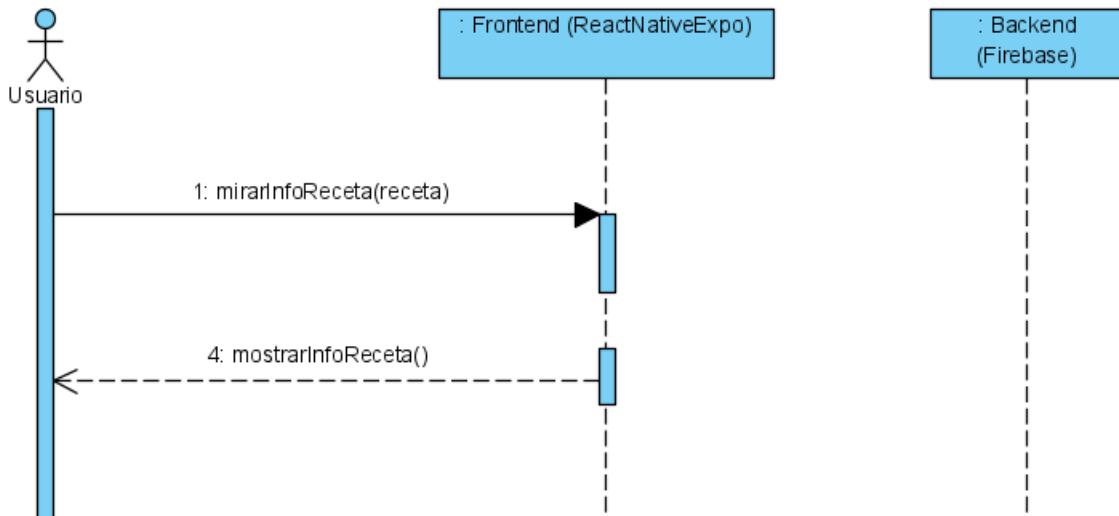


Figura 4.24: Diagrama de secuencia de “Mostrar Información Receta“.

Para el caso de uso “Iniciar Sesión“ (Web):

La secuencia de Ínicio de Sesión.^{en} este caso es relativa a la aplicación web, esta se puede adaptar a la estructura del diagrama de flujo de la imagen de la siguiente manera:

- Al utilizar “iniciarSesion(nombre, contraseña)“, se requiere que el administrador introduzca su nombre de usuario y contraseña correspondientes para que se puedan cargar sus permisos administrativos.
- Al utilizar “opciones()“, el usuario puede elegir entre las siguientes acciones:
 - **Acceder y gestionar a recursos:** El usuario puede acceder a los recursos del sistema, como son las recetas creadas por los usuarios y los ingredientes generados por los administradores.
 - Al utilizar “eliminarIngrediente(ingrediente)“, el administrador puede seleccionar uno de los ingredientes de una lista para ser eliminado del almacenamiento.
 - Al utilizar “nuevoIngrediente(nombre, tipo)“, el administrador puede introducir en un formulario el ingrediente que desea añadir y el sistema lo generará en el backend.

- Al utilizar “eliminarReceta(receta)“, el administrador puede seleccionar uno de las recetas que visualice o despliegue de una lista, para ser eliminado del almacenamiento de la base de datos de la aplicación móvil.
- **Cerrar sesión:** El sistema elimina la información de la cuenta del usuario de la memoria y muestra la pantalla de inicio de sesión.
 - Al utilizar “cerrarSesion()“, el sistema cierra la sesión del administrador.

Las demás opciones como en anteriores apartados, se detallan claramente en el diagrama o en las referencias correspondientes.

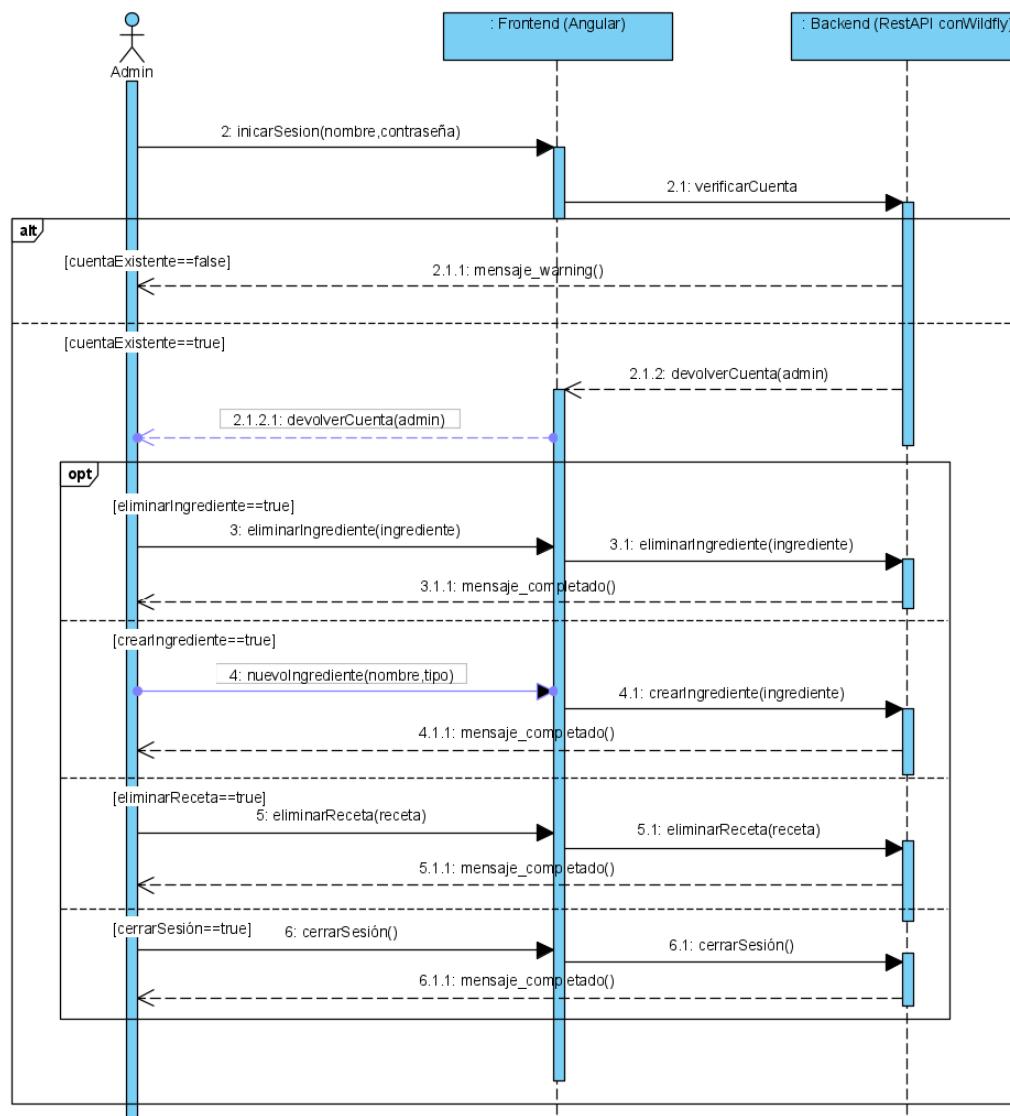


Figura 4.25: Diagrama de secuencia de “Iniciar Sesión” en Web.

4.3.5. Herramientas de desarrollo

En este subapartado, se brinda información detallada sobre las herramientas seleccionadas, incluyendo su propósito, funcionalidades y ventajas específicas que las hacen adecuadas para el desarrollo de nuestra plataforma de gestión de recetas e ingredientes.

A continuación, se describen las herramientas utilizadas en nuestro proyecto, detallando sus características y cómo se integran en el desarrollo de la aplicación móvil y la página web de administración.

Visual Studio Code + Angular

Visual Studio Code es el entorno de desarrollo elegido para escribir y editar los scripts del frontend de la página web de administración, utilizando Angular como framework. Visual Studio Code ofrece una interfaz intuitiva y personalizable, junto con funcionalidades como resaltado de sintaxis, autocompletado y depuración integrada, que facilitan el proceso de desarrollo.

Angular es un framework robusto y eficiente para el desarrollo de aplicaciones web, proporcionando herramientas y estructuras que permiten crear interfaces de usuario dinámicas y responsivas. Con Angular, hemos desarrollado las distintas pantallas de gestión de recetas e ingredientes, así como las funcionalidades de inicio de sesión para nuestros administradores. Además añadimos comprobaciones en cada consulta que se realiza, para evitar enviar peticiones erróneas o nocivas nuestro backend.

Visual Studio Code + React Native Expo

Para el desarrollo del frontend de la aplicación móvil, hemos utilizado Visual Studio Code en combinación con React Native Expo. React Native Expo facilita la creación de aplicaciones móviles multiplataforma, permitiendo desarrollar tanto para iOS como para Android con un solo código base.

React Native Expo proporciona una amplia gama de componentes y APIs que simplifican el desarrollo de nuestra aplicación móvil, desde la navegación hasta la gestión de estado y el acceso a servicios nativos del dispositivo. Nosotros utilizamos estas herramientas para crear las distintas pantallas de la aplicación móvil, como la creación y visualización de recetas, listas de favoritos, y la gestión de la cuenta del usuario, en las cuales hicimos gran uso de esas características.

Visual Studio Code + Java

Para el backend de la página web de administración, hemos utilizado Visual Studio Code junto con Java para crear un API REST de ingredientes. Este API REST maneja todas las operaciones relacionadas con los ingredientes, desde la creación y actualización hasta la eliminación y consulta de datos.

Hemos creado JavaBeans para los ingredientes y los administradores, permitiendo una gestión estructurada y eficiente de estas entidades. Además, implementamos el patrón de diseño Factory para gestionar el inicio de sesión y las peticiones de gestión de recetas, utilizando un inicializador para configurar el acceso a Firebase al inicio de las peticiones.

Las solicitudes provenientes de Angular se manejan mediante servlets en Java, que procesan las peticiones HTTP y se comunican con Firebase para realizar operaciones en Firestore, incluyendo la gestión de autenticación y sincronización de datos. Este enfoque aprovecha las ventajas de Java y Visual Studio Code para asegurar un backend escalable y mantenable que cumple con las necesidades de la aplicación web y móvil.

Wildfly

Wildfly es el servidor de aplicaciones elegido para desplegar nuestro servicio de API REST. Wildfly es conocido por su alta rendimiento y capacidad de manejar aplicaciones empresariales complejas, lo que lo hace ideal para nuestro proyecto en caso de que escalará.

Al utilizar Wildfly, aseguramos que nuestro servicio de API REST esté disponible para la creación de recetas en la aplicación móvil y para la administración de recetas en la página web, proporcionando una infraestructura confiable y escalable para la gestión de los datos de nuestra plataforma.

Firebase

Firebase es la plataforma utilizada para almacenar los datos relacionados con la aplicación móvil, incluyendo usuarios, recetas, comentarios y favoritos. Firebase ofrece una base de datos NoSQL en tiempo real que facilita el almacenamiento y sincronización de datos entre los usuarios y la aplicación.

Con Firebase, podemos gestionar la autenticación de usuarios y proporcionar una experiencia de usuario fluida y responsive, gracias a su capacidad de manejar grandes volúmenes de datos en tiempo real. Además, nos permite escalar, en base a un presupuesto, fácilmente nuestra aplicación a medida que crece la base de usuarios, garantizando un rendimiento consistente.

Visual Paradigm

Visual Paradigm es la herramienta de modelado visual utilizada para desarrollar todos los diagramas necesarios en las etapas de análisis y diseño de nuestro proyecto. Esta herramienta nos permite crear diagramas de casos de uso, diagramas de clases, diagramas de secuencia, y otros modelos visuales que ayudan a representar y entender los distintos aspectos del sistema.

Con Visual Paradigm, podemos comunicar de manera clara y precisa la estructura y el funcionamiento de nuestra plataforma, facilitando la colaboración entre los miembros del equipo de desarrollo y los interesados en el proyecto, como se está realizando a lo largo de esta memoria.

4.3.6. Lenguaje de programación

En este proyecto, hemos empleado distintos lenguajes de programación según las necesidades específicas de cada componente. Para el desarrollo del frontend de la aplicación web, se optó por TypeScript junto con Angular, lo que garantizó una estructura robusta y de fácil mantenimiento. En el caso del frontend de la aplicación móvil, se implementó en JavaScript utilizando React Native Expo, lo que simplificó el desarrollo multiplataforma. En cuanto al backend, se utilizó Java para crear un API REST encargado de gestionar las operaciones relacionadas con los ingredientes y la administración de recetas de manera eficiente. Además, Firebase se empleó para la autenticación de usuarios y el almacenamiento de datos en tiempo real, lo que mejoró la sincronización y la gestión de datos dinámicos. Es importante destacar que Firebase utiliza JavaScript para todos sus servicios, excepto para el almacenamiento de datos, que se realiza con un lenguaje NoSQL.

4.3.7. Componentes externos

En nuestro proyecto, hemos aprovechado la integración de componentes externos para enriquecer las tecnologías utilizadas. Estos componentes externos son herramientas, bibliotecas o servicios desarrollados por terceros que nos permiten agregar funcionalidades adicionales y mejorar la experiencia de nuestros usuarios. En el caso específico de la aplicación móvil desarrollada con React Native Expo, hemos utilizado varias librerías de componentes externos para mejorar la experiencia del usuario y facilitar la programación de los diferentes pantallas dentro del proyecto.

Librería @react-navigation/native

La librería @react-navigation/native es una de las principales herramientas que hemos utilizado para gestionar la navegación en nuestra aplicación móvil. Esta librería nos permite crear rutas y gestionar la transición entre diferentes pantallas de manera sencilla y eficiente. Su uso es esencial para la estructuración de la aplicación, facilitando una navegación fluida y coherente entre las distintas vistas que el usuario puede visitar. Como ejemplos de su utilización, se encuentra el reenvío de los usuarios a la pantalla de la cuenta para que inicien sesión y desbloqueen ciertos servicios, así como la propia navegabilidad por "Stacks"² del tabulador inferior de la aplicación.

Librería expo-image-picker

La librería expo-image-picker se ha integrado para permitir a los usuarios seleccionar y subir imágenes desde su dispositivo. Esta funcionalidad es crucial en la sección de creación y edición de recetas, donde los usuarios pueden agregar imágenes de los platos que han preparado. La facilidad de uso y la integración nativa con Expo hacen que el proceso de selección y carga de imágenes sea rápido y sin complicaciones. Además, esta librería también resulta útil en la edición del perfil del usuario logueado en la aplicación.

Librería react-native-toast-message

Para mejorar la experiencia del usuario con notificaciones y mensajes en pantalla, hemos implementado la librería react-native-toast-message. Esta librería nos permite mostrar mensajes temporales y personalizables que informan al usuario sobre el estado de sus acciones, como confirmaciones de guardado, errores, o cualquier otro tipo de feedback necesario. Su uso es recurrente en toda la aplicación para asegurar que los usuarios estén siempre informados sobre el resultado de sus interacciones.

Librería react-native-vector-icons

La librería react-native-vector-icons se ha utilizado para incorporar íconos atractivos y funcionales en nuestra aplicación móvil. Esta librería proporciona una amplia variedad de íconos personalizables que mejoran la interfaz de usuario, haciéndola más intuitiva y visualmente agradable. Los íconos se han integrado en diferentes partes de la aplicación,

²Conjunto de pantallas que se apilan una sobre otra. Cuando el usuario navega hacia una nueva pantalla, esta se coloca en la cima del stack, y al retroceder, se elimina la pantalla superior, mostrando la anterior. Este modelo permite una navegación lineal y ordenada entre vistas.

como botones de acción, menús y barras de navegación, aportando a la estética y usabilidad del proyecto.

Librería react-native-elements

Finalmente, la librería react-native-elements ha sido fundamental para construir componentes de interfaz de usuario reutilizables y estilizados. Esta librería ofrece una colección de componentes preconstruidos que hemos utilizado para crear botones, tarjetas, formularios y otros elementos interactivos. Su integración nos ha permitido mantener una apariencia consistente y profesional en toda la aplicación, reduciendo el tiempo de desarrollo y asegurando una experiencia de usuario coherente. Esta librería ha sido la más utilizada durante el desarrollo, debido a que ofrecía un amplio conjunto de opciones en los componentes, superando a las opciones básicas proporcionadas por Expo.

La inclusión de estas librerías externas no solo ha enriquecido la funcionalidad de nuestra aplicación, sino que también ha facilitado el desarrollo y mejorado la experiencia del usuario. Dada su importancia y el valor añadido que aportan, estas librerías destacan como componentes externos clave en nuestro proyecto.

4.3.8. Desarrollo de la conexión entre la aplicación móvil y la página web

En esta sección, detallaremos cómo se ha implementado la conexión entre la aplicación móvil y la página web para obtener y gestionar los datos de los ingredientes, así como para observar y manipular los datos almacenados en Firebase.

Para que la aplicación móvil pueda obtener los datos de los ingredientes, se realiza una petición HTTP utilizando la siguiente función:

```

1 const obtenerIngredientes = async () => {
2   try {
3     // Guardar los ingredientes en el estado local
4     const response = await axios.get(
5       "http://192.168.1.44:8080/plc-pls-mps-tutorial/ingredientes.html"
6     );
7     const transformedIngredients = response.data.map((ingrediente) => ({
8       value: ingrediente.nombre,
9       label: ingrediente.nombre,
10      image: {
11        uri: `${ingrediente.tipo}.png`,
12      },
13      category: ingrediente.tipo,
14      cantidad: "0",
15    })
16   } catch (error) {
17     console.error(error);
18   }
19 }
```

```

15});  
    // Guardar los ingredientes transformados en el estado local  
17    setData(transformedIngredients);  
} catch (error) {  
19    console.error("Error al obtener los ingredientes:", error);  
}  
21};  


```

Código 4.3: Función de obtención de ingredientes

Esta función hace una solicitud GET a la URL especificada , obtiene los datos de los ingredientes y los transforma en un formato adecuado para ser utilizados en la aplicación. Los ingredientes transformados se almacenan en el estado local de la aplicación. En este caso, nuestras peticiones dependen mucho de que se encuentre en la misma red porque no utilizamos una IP publica, sino que usamos la propia IP del dispositivo sobre el que se este lanzando Wildfly.

Por otro lado, para que la página web pueda observar y gestionar los datos creados por la aplicación móvil que están almacenados en Firebase, el proceso se simplifica al tratarse de un servicio en la nube proporcionado por terceros a través de una URL pública. Para acceder a ella, es necesario inicializar Firebase en el backend. Esto se logra con la siguiente clase:

```

1 package es.uv.etse.twcam.backend.apirest;  
  
3 import com.google.auth.oauth2.GoogleCredentials;  
import com.google.firebase.FirebaseApp;  
5 import com.google.firebase.FirebaseOptions;  
import java.io.IOException;  
7 import java.io.InputStream;  
  
9 public class FirebaseInitializer {  
private static boolean initialized = false;  
11 public static void initialize() throws IOException {  
    if (!initialized) {  
        InputStream serviceAccount =  
InitServlet.class.getClassLoader().getResourceAsStream("cookingV2.json");  
15        FirebaseOptions options = new FirebaseOptions.Builder()  
            .setCredentials(GoogleCredentials.fromStream(serviceAccount))  
17            .build();  
  
19        FirebaseApp.initializeApp(options);  
21        initialized = true;  
    }  
23}  


```

Código 4.4: Inicialización de Firebase

Esta clase asegura que Firebase se inicialice correctamente utilizando las credenciales especificadas en el archivo cookingV2.json, que generamos desde nuestro repositorio.

Una vez inicializado Firebase, el siguiente servlet gestiona las peticiones realizadas desde Angular para interactuar con los datos almacenados en Firebase:

```
1 package es.uv.etse.twcam.backend.apirest;
2
3 import com.google.api.core.ApiFuture;
4 import com.google.cloud.firestore.CollectionReference;
5 import com.google.cloud.firestore.DocumentReference;
6 import com.google.cloud.firestore.Firestore;
7 import com.google.cloud.firestore.Queryimport
8     com.google.cloud.firestore.QueryDocumentSnapshot;
9 import com.google.cloud.firestore.QuerySnapshot;
10 import com.google.cloud.firestore.WriteResult;
11 import com.google.firebase.cloud.FirestoreClient;
12 import com.google.gson.Gson;
13
14 import es.uv.etse.twcam.backend.business.javabeans.Receta;
15 import java.io.IOException;
16 import java.util.ArrayList;
17 import java.util.List;
18 import java.util.concurrent.ExecutionException;
19 import javax.servlet.ServletException;
20 import javax.servlet.annotation.WebServlet;
21 import javax.servlet.http.HttpServlet;
22 import javax.servlet.http.HttpServletRequest;
23 import javax.servlet.http.HttpServletResponse;
24 import org.apache.logging.log4j.*;
25
26 @WebServlet("/recetas")
27 public class RecetasServlet extends HttpServlet {private static final
28     Logger logger = LogManager.getLogger(RecetasServlet.class.getName());
29
30     @Override
31     protected void doGet(HttpServletRequest request, HttpServletResponse
32         response)
33         throws ServletException, IOException {
34         try {
35             addCORSHeaders(response);
36
37             FirebaseInitializer.initialize();
38
39             // Accede a Firestore
40             Firestore db = FirestoreClient.getFirestore();
41
42             // Accede a la colección "recetas"
43             CollectionReference recetasCollection = db.collection("recetas");
44
45             ApiFuture<QuerySnapshot> future = recetasCollection.get();
46             QuerySnapshot querySnapshot = future.get();
```

```
    List<QueryDocumentSnapshot> documents =
46    querySnapshot.getDocuments();
47
48    // Construye la lista de objetos Receta
49    List<Receta> recetas = new ArrayList<>();
50    for (QueryDocumentSnapshot document : documents) {
51        System.out.println(document);
52        Receta receta = document.toObject(Receta.class);
53        recetas.add(receta);
54    }
55
56    // Configura la respuesta HTTP
57    response.setContentType("application/json");
58    response.setCharacterEncoding("UTF-8");
59
60    // Convierte la lista de recetas a JSON y env a la respuesta
61    response.getWriter().write(new Gson().toJson(recetas));
62 } catch (InterruptedException | ExecutionException e) {
63     e.printStackTrace();
64     response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
65 }
66
67 @Override
68 protected void doDelete(HttpServletRequest request, HttpServletResponse
69     response)
70     throws ServletException, IOException {
71     try {
72
73         addCORSHeaders(response);
74
75         FirebaseInitializer.initialize();
76
77         // Accede a Firestore
78         Firestore db = FirestoreClient.getFirestore();
79
80         // Paso 0: Obtiene el ID de la receta a eliminar del par metro de
81         // la solicitud
82         String recetaId = request.getParameter("id");
83
84         if (recetaId != null && !recetaId.isEmpty()) {
85
86             // Paso 1: Eliminar todos los favoritos encontrados
87             DocumentReference recetaRef =
88             db.collection("recetas").document(recetaId);
89             ApiFuture<WriteResult> future = recetaRef.delete();
90             future.get(); // Espera a que se complete la eliminaci n
91
92             CollectionReference comentariosCollection =
93             db.collection("comentarios");
94             Query query = comentariosCollection.whereEqualTo("idRecipe",
95             recetaId);
96             ApiFuture<QuerySnapshot> querySnapshotFuture = query.get();
97             QuerySnapshot querySnapshot = querySnapshotFuture.get();
98 }
```

```

List<QueryDocumentSnapshot> comentarios =
querySnapshot.getDocuments();

// Paso 2: Eliminar todos los comentarios encontrados
for (QueryDocumentSnapshot comentario : comentarios) {
    // Obtiene la referencia al documento del comentario
    DocumentReference comentarioRef = comentario.getReference();
    // Elimina el comentario
    ApiFuture<WriteResult> deleteFuture =
comentarioRef.delete();
    // Espera a que se complete la eliminación
    deleteFuture.get();
}

comentariosCollection = db.collection("favoritos");
query = comentariosCollection.whereEqualTo("idRecipe",
recetaId);
querySnapshotFuture = query.get();
querySnapshot = querySnapshotFuture.get();
List<QueryDocumentSnapshot> favoritos =
querySnapshot.getDocuments();

// Paso 3: Eliminar todos los favoritos encontrados
for (QueryDocumentSnapshot favorito : favoritos) {
    // Obtiene la referencia al documento del favorito
    DocumentReference favoritoRef = favorito.getReference();
    // Elimina el favorito
    ApiFuture<WriteResult> deleteFuture = favoritoRef.delete();
    // Espera a que se complete la eliminación
    deleteFuture.get();
}

response.setStatus(HttpServletResponse.SC_OK);
} else {
    response.setStatus(HttpServletResponse.SC_BAD_REQUEST);
}

} catch (InterruptedException | ExecutionException e) {
e.printStackTrace();
response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
}
}

@Override
protected void doOptions(HttpServletRequest request, HttpServletResponse
response) {
addCORSHeaders(response);

try {
super.doOptions(request, response);
} catch (ServletException se) {
logger.error("Error genérico en la clase padre");
} catch (IOException ioe) {
}
}

```

```
142     logger.error("Error gen ríco de salida la clase padre");
143 }
144 }

145 private void addCORSHeaders(HttpServletRequest response) {
146     response.addHeader("Content-Type", "application/json");
147     response.setCharacterEncoding("UTF-8");
148     response.addHeader("Access-Control-Allow-Credentials", "true");
149     response.addHeader("Access-Control-Allow-Methods", "OPTIONS, HEAD,
150     DELETE, GET");
151     response.addHeader("Access-Control-Allow-Headers",
152     "authorization, content-type");
153     response.addHeader("Access-Control-Allow-Origin", "*");
154 }
```

Código 4.5: Servlet de gestión de recetas

Este servlet maneja las solicitudes HTTP GET y DELETE para la colección de recetas en Firebase. La función doGet() obtiene todas las recetas y las envía como una respuesta JSON, mientras que la función doDelete() elimina una receta específica y sus comentarios y favoritos asociados, para limpiar todo contenido relacionado a esa receta desde nuestro backend de Firebase y dejar mas espacio a otras nuevas recetas.

4.4. Desarrollo de la Aplicación Móvil

En esta sección, se detallará el desarrollo de la aplicación móvil, centrándose en las diferentes interfaces y los componentes que las componen. Se describirán las funcionalidades principales implementadas en cada pantalla, proporcionando una visión completa de cómo se ha estructurado y diseñado la aplicación para ofrecer una experiencia de usuario óptima. A través de esta exploración, se destacarán las herramientas y técnicas utilizadas para integrar las diversas funcionalidades necesarias para cumplir con los objetivos del proyecto.

4.4.1. Interfaces de listado de recetas

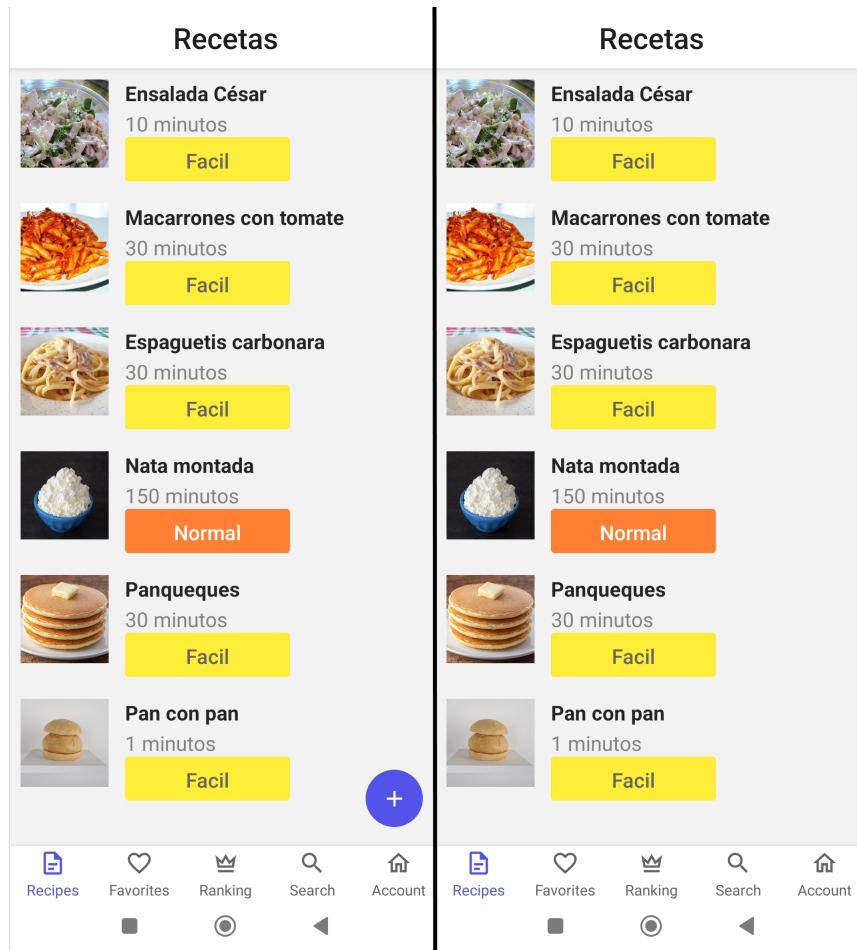


Figura 4.26: Diseño final de interfaces de listado de recetas

En estas interfaces se muestra una lista de recetas creada a partir de un componente "Flatlist", en el cual se ha integrado un panel interactivo que contiene una imagen de la receta, el nombre de la receta, el tiempo de preparación y la dificultad de la misma, representada con un botón que indica el nivel de dificultad mediante un nombre y un color distintivo (amarillo para fácil, naranja para normal y rojo para difícil).

La única diferencia entre las dos pantallas es el inicio de usuario en la segunda pantalla. En esta, se muestra un ícono de suma interactivo que redirige al usuario hacia la página de creación de recetas a partir del nombre de su página.

La función más peculiar de esta página es el cálculo de la dificultad de la receta, el cual se realiza mediante un sistema de puntuación que depende de la cantidad de ingredientes, la cantidad de pasos y el tiempo de preparación.

4.4.2. Interfaz de creación de receta

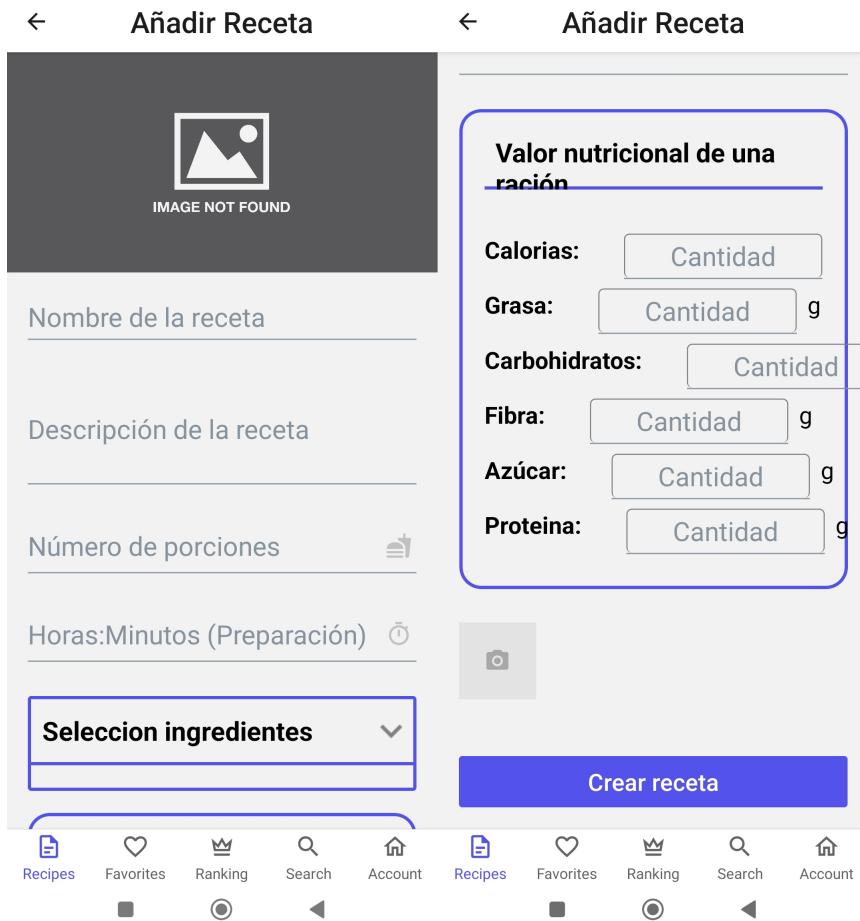


Figura 4.27: Diseño final de interfaz de información de creación de receta.

En esta interfaz se muestra un formulario de creación de una receta. En el encabezado, se presenta una imagen por defecto en caso de que no se haya escogido ninguna imagen mediante el componente ImagePicker.^a Al final del formulario. Si se han escogido varias fotos, la imagen por defecto se sustituirá por la primera seleccionada.

A continuación, se muestran varias entradas obligatorias como el nombre, la descripción, el número de porciones que se obtienen, el tiempo de preparación, la selección de ingredientes y los pasos a seguir. También hay entradas opcionales como los consejos y los valores nutricionales de la receta.

Al presionar el botón "Crear Receta", se verificará que el usuario haya ingresado la información pertinente mediante la validación de Formik. En caso de error, se utilizará el componente "Toast" para mostrar un mensaje en pantalla indicando que no se ha realizado correctamente y se marcará la entrada con color rojo.

La función más particular de este subapartado es el componente "ImagePicker", que permite al usuario de cualquier plataforma acceder a su galería y seleccionar las fotos que representen la receta. Si el usuario desea deshacer su elección, solo deberá pulsar sobre la imagen seleccionada.

4.4.3. Interfaz de información de receta

Receta

Ensalada César ★★★★★

La mejor ensalada para verano

🕒 0 horas y 10 minutos

Ingredientes para - 2 +

- Lechuga : 1 unidad
- Salsa César : 20 ml
- Picatostes : 100 gramos

Calorías: 167
Grasas: 10 gramos
Carbohidratos: 9.8 gramos
Fibra: 2 gramos
Azúcares: 1 gramo
Proteínas: 8.4 gramos

Escribe una opinión

Comentario

Normal

★★★★★

Titulo

Comentario

Enviar review

Recipes Favorites Ranking Search Account

Figura 4.28: Diseño final de interfaz de información de receta.

En la interfaz de la izquierda se muestra toda la información que se ha completado en la interfaz de creación de una nueva receta 4.4.2. Además, incluye un apartado para la valoración con estrellas, otro para añadir a favoritos (disponible si la sesión está iniciada), y al final, un área donde se pueden crear comentarios sobre la receta junto con un listado de los demás comentarios.

También se han implementado cambios dinámicos, como la modificación del número de porciones para visualizar la cantidad de ingredientes necesarios, y la inclusión de imágenes

gráficas que describen el tipo de alimento de cada ingrediente.

Para acceder al formulario en la interfaz de la derecha, debes estar iniciado sesión, igual que para añadir la receta a favoritos. Si no estás iniciado sesión, se mostrará un botón de navegación a la página de inicio de sesión. Si ya estás iniciado sesión, podrás acceder a la sección interactiva de comentarios.

En la sección de comentarios, deberás seleccionar la cantidad de estrellas, insertar el título del comentario y escribir tu opinión antes de enviarlo a Firebase para su almacenamiento.

La función más destacada de este subapartado es la creación de comentarios. Detecta si el usuario ha iniciado sesión o si ya ha creado un comentario anteriormente, permitiendo la creación de un nuevo comentario que modifique la valoración de la receta.

4.4.4. Interfaces de favoritos

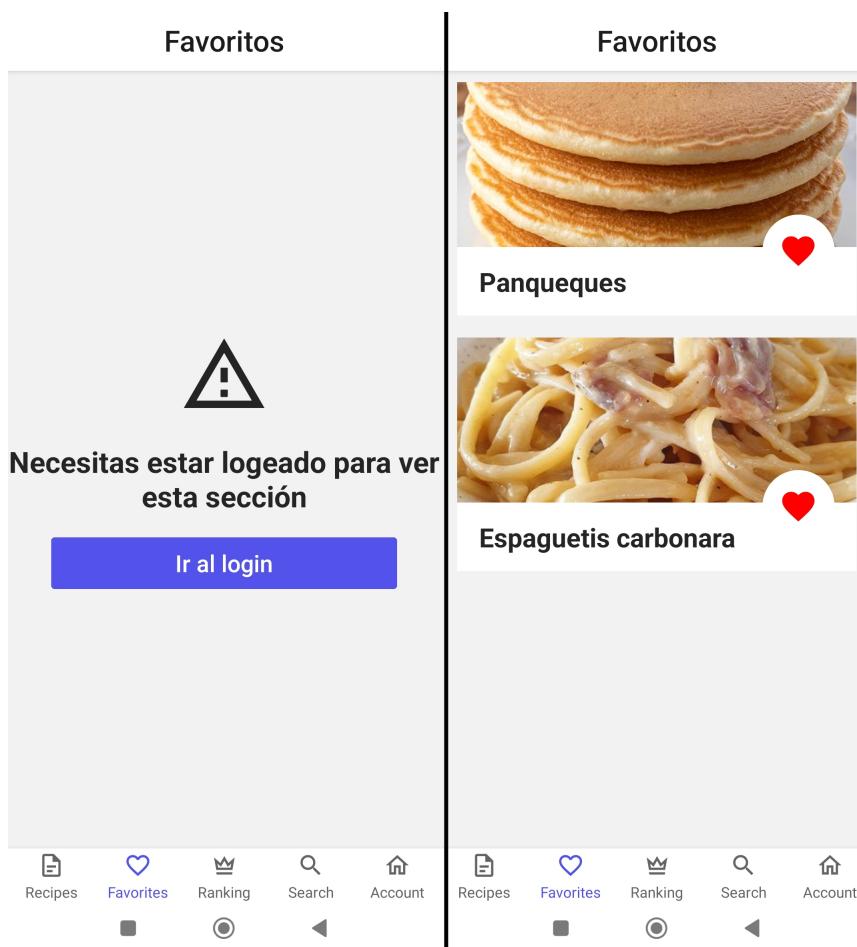


Figura 4.29: Diseño final de interfaces de favoritos

En la interfaz de la izquierda, se muestra el estado cuando el usuario no se ha autenticado. Centrada en la pantalla, aparece una imagen de advertencia acompañada de un texto explicando por qué no puede acceder a esta función. Además, hay un botón con el texto "Ir a login" que, al interactuar, redirige al usuario a la página de inicio de sesión de la aplicación.

En la interfaz de la derecha, se muestra un listado de recetas obtenidas de manera similar a la interfaz de listado de recetas 4.4.1, pero con algunas diferencias. En este caso, el panel interactivo presenta una imagen más grande de la receta, muestra únicamente el nombre de la receta, y utiliza el ícono de corazón empleado para añadir a favoritos en la interfaz de información de la receta 4.4.3.

La función más destacada de esta interfaz es la obtención de la lista de favoritos. Aunque es un proceso sencillo, consiste en recuperar las recetas asociadas en la tabla de favoritos junto con el identificador del usuario que ha iniciado sesión.

4.4.5. Interfaz de clasificación

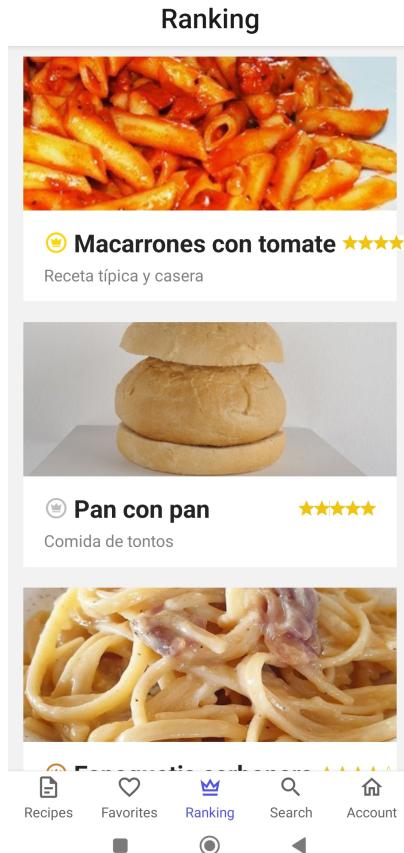


Figura 4.30: Diseño final de interfaz de clasificación

Esta es una de las interfaces más simples, que combina la visibilidad de la interfaz de favoritos 4.4.4 con las funciones de obtención de recetas de la interfaz de lista de recetas 4.4.1. La única diferencia en la visibilidad es que, en lugar de un ícono de corazón, se muestra la valoración de la receta y la descripción asociada a esta. Aunque la obtención de recetas se realiza de la misma manera, esta lista está ordenada y limitada a las 10 recetas mejor valoradas, destacando las tres primeras con iconos olímpicos.

En esta interfaz, lo que más resalta no es la funcionalidad, sino la presentación de las recetas.

4.4.6. Interfaz de búsqueda

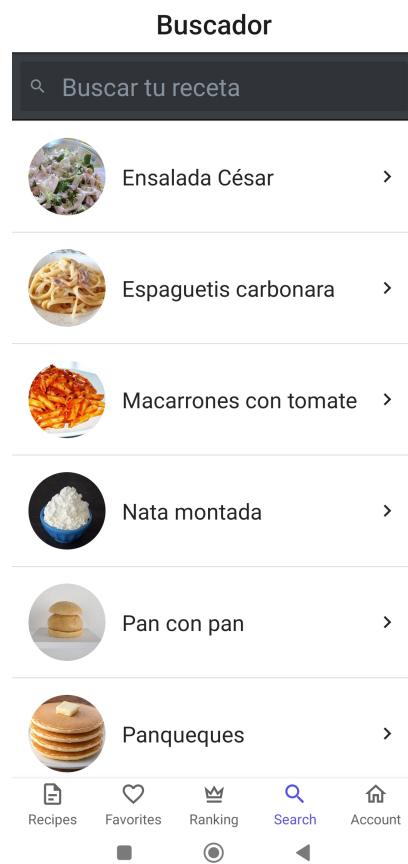


Figura 4.31: Diseño final de interfaz de búsqueda

Esta es una de las interfaces más sencillas del proyecto, un buscador minimalista que cuenta con las funciones de obtención de recetas de la interfaz de lista de recetas 4.4.1.

La obtención de recetas se realiza de la misma manera, aunque en este caso se filtran

según el contenido introducido en el buscador ubicado en la parte superior. Solo se muestra la foto de la receta y su nombre en el elemento interactivo, el cual redirige a la interfaz de información sobre la receta 4.4.3.

A diferencia de la interfaz de lista de recetas, una de las características más destacadas de esta interfaz es la funcionalidad de realizar peticiones específicas a la base de datos de Firebase mediante código. Esto se logra utilizando funcionalidades de una biblioteca externa que facilita la obtención de información del backend sin necesidad de conocer los comandos específicos para acceder a estos datos.

4.4.7. Interfaces de cuenta

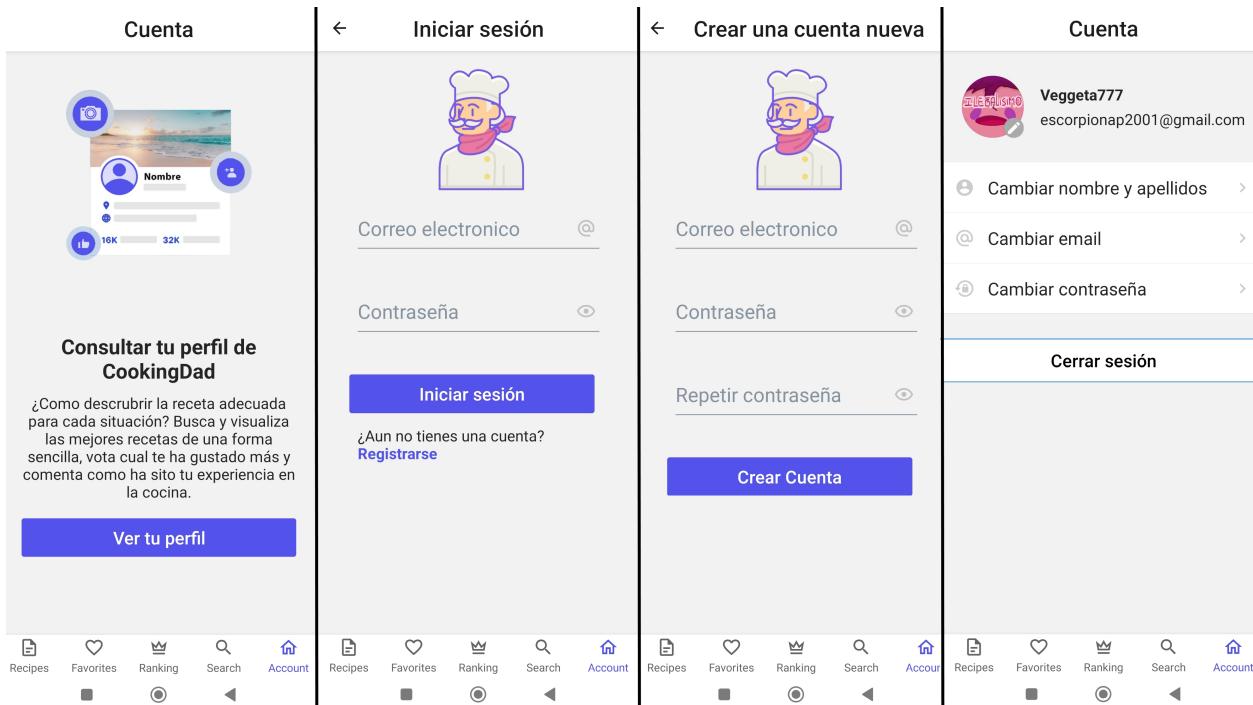


Figura 4.32: Diseño final de interfaces de cuenta

En esta imagen se presentan cuatro interfaces relacionadas con la pestaña de cuenta, ordenadas según el recorrido del usuario.

En la primera interfaz, se muestra una imagen del perfil de un usuario junto con una breve explicación de cómo se puede consultar el perfil del propio usuario. Debajo de todo esto, hay un botón con el texto "Ver perfil" que redirige al usuario a la siguiente interfaz.

La segunda interfaz muestra un formulario típico de inicio de sesión donde se valida automáticamente que los datos enviados a Firebase para la autenticación cumplan con las

condiciones básicas (más de 6 caracteres, un correo válido y sin campos vacíos). Si no se cumplen estas condiciones, se muestra un mensaje de error en pantalla o se resalta en rojo el campo con el fallo. Si el usuario no tiene una cuenta creada, puede pulsar sobre el texto interactivo "Registrarse" para navegar a la página de creación de una nueva cuenta.

En la tercera interfaz, se encuentra el mismo formulario que en la interfaz anterior, pero con un campo adicional para repetir la contraseña y verificar que el usuario la recuerde correctamente y no contenga errores.

Si no hay ningún error en las dos interfaces anteriores, el usuario es redirigido a esta última interfaz, donde se muestra la imagen de perfil, el nombre del usuario y el correo vinculado a la cuenta. Si el usuario desea cambiar alguno de estos valores, puede hacerlo utilizando el "ImagePicker" para cambiar la imagen sin restricciones o pulsando cualquiera de los enlaces interactivos ("Cambiar nombre y apellidos", "Cambiar email", "Cambiar contraseña"), introduciendo su contraseña actual para que el backend permita el cambio. Al final de todo, si el usuario desea cerrar sesión, puede hacerlo pulsando el botón "Cerrar sesión", lo que cerrará la sesión y redirigirá al usuario a la primera interfaz.

La funcionalidad más destacada de estas interfaces, desde el punto de vista del programador, es la que permite verificar el contenido escrito en las contraseñas en los formularios de inicio de sesión o de registro al pulsar el ícono del ojo, el cual permite ocultar y mostrar las contraseñas de forma dinámica.

4.5. Desarrollo de la Página Web

En esta sección, se detallará el desarrollo de la página web, centrándose en las diferentes interfaces y los componentes que las componen. Se describirán las funcionalidades principales implementadas en cada pantalla, proporcionando una visión completa de cómo se ha estructurado y diseñado la aplicación para ofrecer una experiencia de usuario óptima. A través de esta exploración, se destinarán las herramientas y técnicas utilizadas para integrar las diversas funcionalidades necesarias para cumplir con los objetivos del proyecto.

4.5.1. Interfaz de inicio

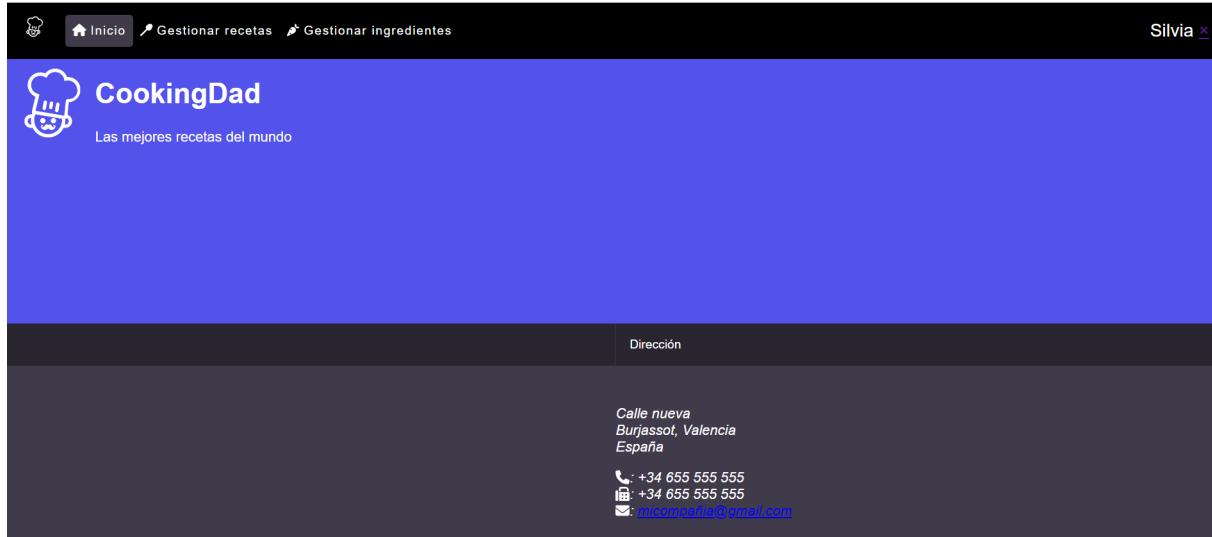


Figura 4.33: Diseño final de interfaz de inicio

En esta interfaz se muestra una cabecera compuesta por un navegador que incluye el ícono de la plataforma junto con las opciones de navegación. Las opciones de navegación permiten cambiar de página y destacan en gris claro la página actual en la que se encuentra el usuario. A la derecha de la cabecera, hay un botón que permite acceder a la interfaz de inicio de sesión de la web 4.5.2.

En el cuerpo de la página, se muestra el ícono de la plataforma acompañado de un mensaje de saludo.

Finalmente, en el pie de página, se proporciona información de contacto de la empresa encargada de la aplicación web y un enlace que puede redirigir a la web oficial de esa empresa en caso de que tenga.

4.5.2. Interfaz de inicio de sesión

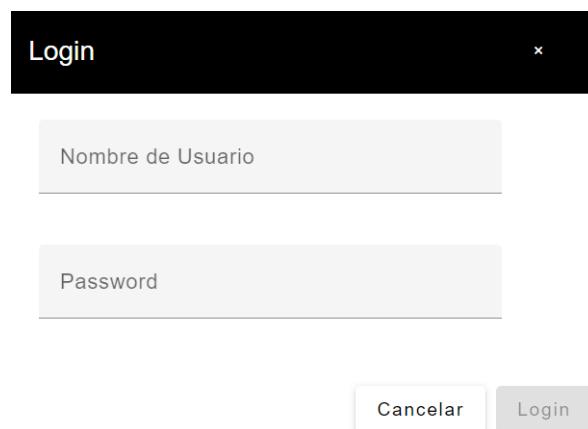


Figura 4.34: Diseño final de interfaz de inicio de sesión

En esta interfaz se muestra el formulario de inicio de sesión, el cual se sobrepone a la interfaz de inicio 4.5.1 utilizando los diálogos de Angular. Este formulario cuenta con una validación dinámica de las entradas, resaltando en rojo cualquier entrada inválida y mostrando un mensaje de alerta sobre el error antes de hacer uso del servicio que accede al backend.

Para cerrar este panel de inicio de sesión, el usuario puede ingresar a su cuenta, hacer clic en el botón de cancelar, pulsar el ícono de la cruz o hacer clic fuera del panel.

4.5.3. Interfaz de gestión de ingredientes

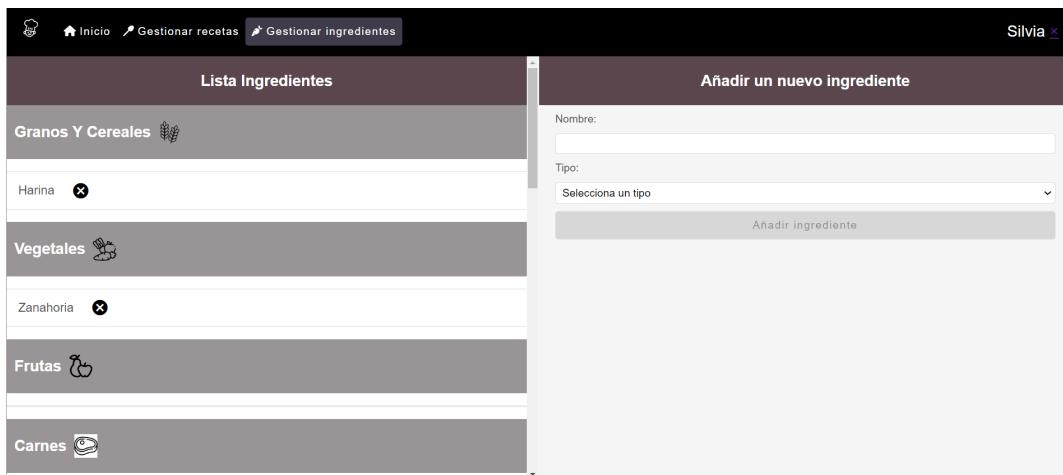


Figura 4.35: Diseño final de interfaz de gestión de ingredientes

El cuerpo de la interfaz de gestión de ingredientes está dividido en dos secciones: la izquierda, destinada a eliminar ingredientes de nuestra API REST, y la derecha, utilizada para agregar nuevos ingredientes a la misma API.

En la sección izquierda, se presentan divisiones que muestran una imagen y un texto del tipo de ingredientes, con un listado de los ingredientes pertenecientes a esa categoría debajo. Cada ingrediente tiene un ícono interactuable que permite eliminarlo permanentemente.

La sección derecha contiene un formulario para agregar un nuevo ingrediente. Este formulario incluye una validación dinámica para asegurarse de que todos los campos se hayan completado o seleccionado antes de utilizar el servicio de agregación de ingredientes.

4.5.4. Interfaz de gestión de recetas

Recetas

Espaguetis carbonara

Descripción:
Deliciosa receta familiar

Porciones pensadas para la receta:
4

Tiempo necesario:
30 minutos

Consejos:
Cuando emplates puedes añadir más queso a los espaguetis para que sepan mejor

Valoración:
4

Figura 4.36: Diseño final de interfaz de gestión de recetas

En el cuerpo de esta interfaz web se encuentra un listado de tarjetas, cada una almacenando una receta de Firebase. A primera vista, cada tarjeta muestra un ícono para eliminar la receta mediante un servicio a nuestra API REST, así como el nombre de la receta.

Cuando el administrador hace clic en una tarjeta, se despliega una lista con la información completada de la receta. Si se ha completado también la información nutricional, esta se encuentra dentro de un desplegable adicional para evitar la sobrecarga de texto en la página.

4.5.5. Apirest

En esta subsección se describen los diferentes endpoints del servicio API REST que se utilizan para gestionar recetas e ingredientes. A continuación, se detallan los endpoints disponibles:

- **GET /**: Este endpoint de nuestra Apirest proporciona un listado con la explicación breve de cada uno de los endpoints que se están explicando sobre esta memoria.
- **GET /recetas**: Este endpoint se utiliza para obtener el listado completo de todas las recetas disponibles en el sistema. Al realizar una solicitud GET a esta ruta, el servidor responde con un array de objetos JSON, donde cada objeto representa una receta con sus respectivos atributos, como el nombre, los ingredientes, las instrucciones, etc.

- **DELETE /recetas?id=X:** Este endpoint permite eliminar una receta específica del sistema. La solicitud DELETE debe incluir un parámetro de consulta (**id**) que indique el identificador único de la receta que se desea eliminar. Por ejemplo, para eliminar la receta con id igual a 5, se enviaría una solicitud a `DELETE /recetas?id=5`. El servidor responde con un mensaje de confirmación de eliminación.
- **GET /ingredientes:** A través de este endpoint, se puede obtener el listado completo de todos los ingredientes disponibles en el sistema. Al realizar una solicitud GET a esta ruta, el servidor responde con un array de objetos JSON, donde cada objeto representa un ingrediente con sus respectivos atributos, como el nombre y la cantidad.
- **POST /ingredientes:** Este endpoint se utiliza para añadir un nuevo ingrediente al listado de ingredientes. La solicitud POST debe incluir en el cuerpo de la solicitud un objeto JSON que contenga la información del nuevo ingrediente, como el nombre y la cantidad. El servidor procesa la solicitud y añade el nuevo ingrediente al sistema, respondiendo con los detalles del ingrediente añadido.
- **DELETE /ingredientes?id=X:** Similar al endpoint de eliminación de recetas, este endpoint permite eliminar un ingrediente específico del sistema. La solicitud DELETE debe incluir un parámetro de consulta (**id**) que indique el identificador único del ingrediente que se desea eliminar. Por ejemplo, para eliminar el ingrediente con id igual a 3, se enviaría una solicitud a `DELETE /ingredientes?id=3`. El servidor responde con un mensaje de confirmación de eliminación.

Estos endpoints proporcionan la funcionalidad básica para la gestión de recetas e ingredientes en el sistema, permitiendo obtener listados, añadir nuevos elementos y eliminar elementos existentes según sea necesario.

Capítulo 5

Pruebas y resultados

5.1. Introducción

En este capítulo se describen detalladamente los métodos empleados para realizar las pruebas de la plataforma, la recopilación y análisis de datos, y la interpretación de los resultados obtenidos. Se examinan los aspectos técnicos de la plataforma, sus limitaciones, y se discuten las implicaciones de los resultados junto con posibles mejoras para futuras versiones. El objetivo principal es validar la funcionalidad y el rendimiento de la plataforma desarrollada, utilizando las tecnologías mencionadas en la sección de nuestro estado del arte 2.5.



Figura 5.1: Capturas de los distintos finales del juego.

5.2. Prueba de rendimiento

Para la página web, se llevaron a cabo pruebas de rendimiento utilizando Selenium en conjunto con JMeter. Estas pruebas se realizaron con diferentes cantidades de usuarios concurrentes, variando desde 2 hasta 512 usuarios, con el objetivo de evaluar el rendimiento del sistema bajo diferentes cargas de trabajo. Para la aplicación móvil, las pruebas se centraron en evaluar el desempeño y la eficiencia de las principales funcionalidades de la aplicación. Se realizaron mediciones de tiempo desde el inicio hasta el final de estas funcionalidades clave.

Las pruebas nos proporcionaron información sobre la capacidad de los sistemas y ayudaron a identificar posibles cuellos de botella o áreas de mejora en el rendimiento tanto de la página web como de la aplicación móvil.

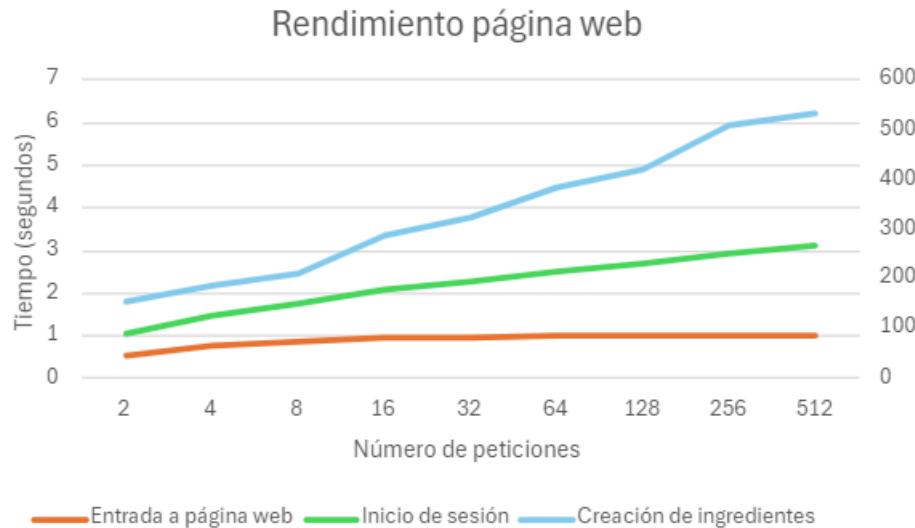


Figura 5.2: Gráfica del rendimiento de la página web.

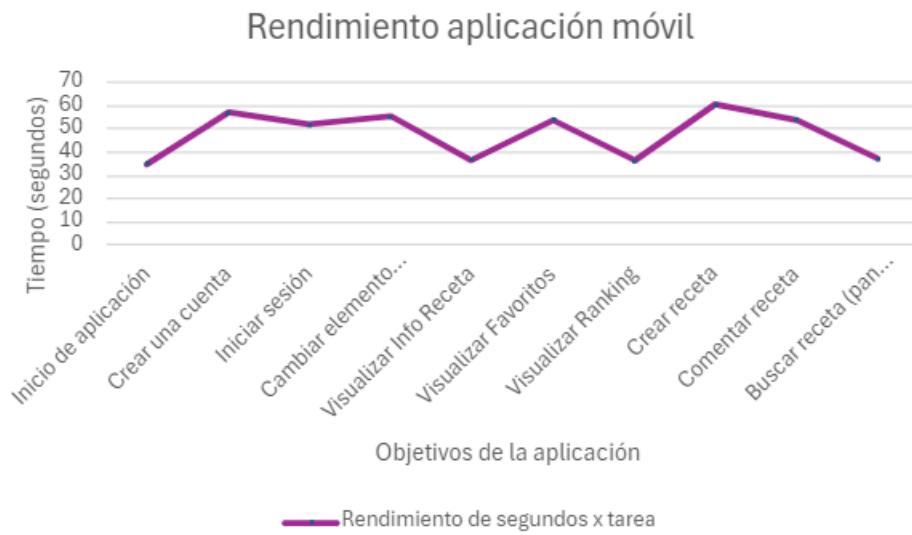


Figura 5.3: Gráfica del rendimiento de la aplicación móvil.

Para la página web, las pruebas indicaron que las operaciones más intensivas, como la creación de ingredientes, sufren un aumento significativo en el tiempo de respuesta bajo cargas altas, mientras que las operaciones básicas como entrar a la página de inicio mantienen tiempos aceptables. En cuanto a la aplicación móvil, los tiempos de respuesta

para todas las funcionalidades son considerablemente altos, especialmente para el inicio de la aplicación y la autenticación, pero esto se debe a la construcción de la aplicación en React Native. En conjunto, estos resultados resaltan áreas críticas que requerirían mejoras para garantizar un rendimiento eficiente y una experiencia de usuario satisfactoria en ambas plataformas. No obstante, dentro del contexto de nuestro proyecto, estos resultados son esperados y se consideran aceptables.

5.3. Prueba de funcionalidad con usuarios

La prueba de funcionalidad de la plataforma se lleva a cabo mediante la participación de usuarios. El objetivo principal de esta prueba es verificar que la página web y la aplicación móvil funcionen de manera adecuada y cumplan con las expectativas y necesidades de los usuarios. A través de esta prueba, se busca evaluar la intuición, dinamismo y aspecto visual de nuestra plataforma, para garantizando una experiencia de usuario excepcional.

Para la realización de estas pruebas, se contó con la colaboración de 4 usuarios con diferentes niveles de experiencia e intuición en el uso de sitios web y aplicaciones móviles:

- Usuario 1: Persona experimentada en sitios web y aplicaciones móviles.
- Usuario 2: Persona experimentada en sitios web y aplicaciones móviles.
- Usuario 3: Persona con poca experiencia en sitios web y aplicaciones móviles.
- Usuario 4: Persona con poca experiencia en sitios web y aplicaciones móviles.

Estos usuarios fueron seleccionados con el objetivo de obtener una variedad de perspectivas y opiniones sobre la plataforma durante las pruebas.

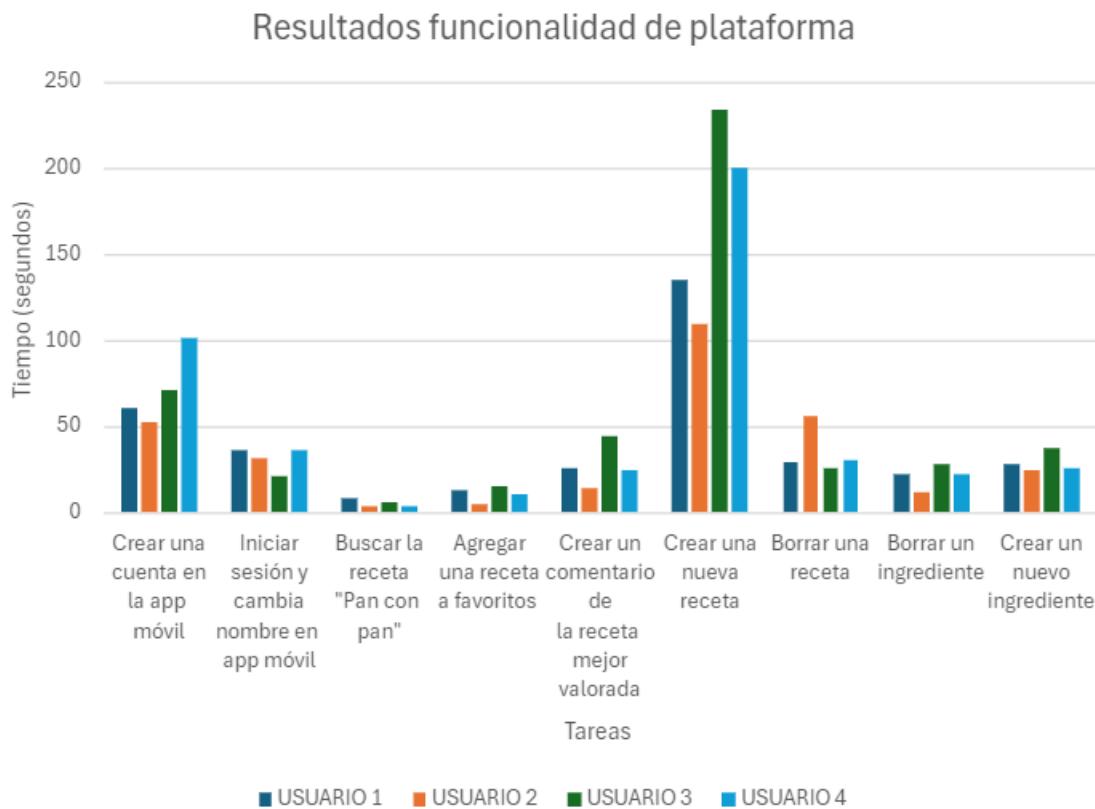


Figura 5.4: Gráfica de resultado de las pruebas de funcionalidad.

En este gráfico se muestran los tiempos promedio que tardan los usuarios de prueba en realizar las diferentes funcionalidades principales de la plataforma, tanto para la aplicación móvil como para el sitio web. En varias tareas, los tiempos de finalización son similares, como al gestionar recetas o navegar por el sitio web. Sin embargo, en la tarea de "Crear una cuenta en la app móvil" y "Crear una nueva receta" se nota una gran discrepancia entre los usuarios con experiencia y aquellos sin experiencia. Esta diferencia notable no se debe a la ambigüedad de las instrucciones, sino al desconocimiento del usuario sobre cómo realizar la tarea o a la complejidad de la receta que los usuarios sin experiencia querían crear con nuestra aplicación.

Aun así, con esta gráfica se puede observar que el uso de esta plataforma no depende significativamente de la experiencia previa, y sus funcionalidades resultan bastante intuitivas para los nuevos usuarios.

5.4. Test SUS

La fase final implica la aplicación de un Test SUS adaptado a nuestro proyecto para evaluar la experiencia de los usuarios con la plataforma desarrollada. Este test proporcionará datos tanto cuantitativos como cualitativos sobre la eficacia del sistema y la satisfacción de los usuarios. Se utilizará el mismo grupo de usuarios que participaron en las pruebas anteriores para garantizar la consistencia en la evaluación. El test constará de 10 preguntas, divididas equitativamente entre afirmaciones positivas y negativas sobre la plataforma, las cuales serán calificadas en una escala del 1 al 5 por los participantes.

Las preguntas formuladas a los usuarios fueron las siguientes:

- La plataforma es intuitiva de usar.
- Hay demasiada inconsistencia en la plataforma.
- Se aprende rápidamente a usar la plataforma.
- Se necesita la ayuda de un experto para poder usar la plataforma.
- Se notifica de cada uno de las acciones correctas e incorrectas del usuario.
- Tuve dificultades para entender si se realizaba correctamente mis acciones.
- Las funciones principales de la aplicación móvil son evidentes.
- Las funciones principales de la aplicación móvil son confusas y no se llegan a entender.
- La web facilita la gestión de las recetas y de los ingredientes.
- La página web complica la gestión de las recetas y de los ingredientes.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Resultado Final
USUARIO 1	5	1	4	2	5	2	5	1	4	2	87.5
USUARIO 2	5	2	5	2	5	1	4	1	4	2	87.5
USUARIO 3	5	1	5	1	5	2	4	1	4	2	90
USUARIO 4	4	2	5	2	5	2	5	1	5	1	90
TOTAL											88.75

Cuadro 5.1: Resultados de la Prueba SUS

A partir de los cálculos realizados en la tabla, podemos concluir que obtener una puntuación mayor a 71 indica que la plataforma proporciona una experiencia satisfactoria tanto en la aplicación móvil como en el sitio web. Si la puntuación se encuentra entre 50 y 70, la plataforma está en el margen y puede requerir algunas mejoras para optimizar la

experiencia del usuario. Sin embargo, una puntuación menor a 50 sería inaceptable, indicando la necesidad de realizar correcciones significativas. En nuestro caso, los resultados obtenidos con el test de SUS muestran que la plataforma ha proporcionado una gran satisfacción a los usuarios, independientemente de su nivel de experiencia en el uso de sitios web y aplicaciones móviles.

Capítulo 6

Conclusiones

6.1. Conclusiones

La motivación principal de este proyecto ha sido desarrollar una plataforma gastronómica que permite a los usuarios con conocimiento básicos o inexistentes gestionar recetas y navegar por la aplicación móvil mientras un administrador gestiona la información desde el sitio web. Nuestro objetivo ha sido proporcionar una herramienta intuitiva y dinámica que brinde una gran experiencia a los usuarios, independientemente de su nivel de experiencia en el uso de tecnologías y del nivel de cocina que posean.

En cuanto a los subobjetivos establecidos en la sección 1.2, hemos logrado implementar la gran mayoría de ellos de manera completa en nuestro proyecto, abarcando tanto las partes visibles para el usuario como la funcionalidad y la lógica interna del programa.

Durante el análisis realizado durante la creación del proyecto, hemos obtenido información relevante sobre los costes y tiempos asociados a proyectos en este sector, así como las limitaciones y ventajas que podrían surgir. También hemos realizado un estudio para determinar el precio de venta del producto, con el objetivo de obtener beneficios en caso de que los inversores buscaran esa meta con nuestra plataforma.

Respecto al diseño e implementación, hemos aplicado los conocimientos adquiridos en el ámbito universitario, tales como la gestión de proyectos, el desarrollo de aplicaciones móviles y la programación tanto del lado del servidor como del cliente. Este proyecto nos ha brindado la oportunidad de enfrentarnos a diversos desafíos y encontrar soluciones a problemas relacionados con el desarrollo completo de aplicaciones web y móviles.

Durante el proceso, hemos comprendido la importancia de realizar copias de seguridad periódicas para evitar pérdidas de datos, así como guardar el proyecto en un disco externo como medida adicional de seguridad. Además, hemos aprendido la importancia

de organizar adecuadamente los recursos, dado que los proyectos completos requieren una gran cantidad de página y componentes, por lo que al finalizar un proyecto puede llegar a resultar caótico a la hora de pulir o corregir errores posteriormente.

También hemos experimentado el valioso apoyo proporcionado por la comunidad de desarrolladores en este sector. En momentos de dificultad o al enfrentarnos a problemas específicos, hemos encontrado respuestas, consejos y soluciones a través de la interacción con otros desarrolladores y participando en comunidades en línea o con la ayuda de compañeros que tuvieron experiencia en los ámbitos de esta plataforma.

En base a los resultados obtenidos, hemos observado un gran interés generado por la plataforma entre diversos grupos de usuarios. Han mostrado aprecio por la facilidad de uso y la experiencia intuitiva tanto en la aplicación móvil como en el sitio web, incluso para aquellos con poca experiencia en el uso de tecnologías.

Esto nos indica que hemos logrado nuestro objetivo de brindar una experiencia atractiva y satisfactoria para una amplia gama de usuarios. La capacidad de la plataforma para adaptarse a diferentes niveles de experiencia ha sido bien recibida y ha generado un mayor nivel de disfrute y compromiso por parte de los usuarios.

En resumen, este proyecto ha cumplido con los objetivos planteados, brindando una herramienta en el ámbito gastronómico que permite una experiencia personalizada y atractiva para diferentes tipos de usuarios. Además, nos ha proporcionado un amplio conocimiento y experiencia en el desarrollo de proyectos de esta naturaleza, así como la oportunidad de aplicar y expandir nuestras habilidades en campos relacionados con nuestra carrera como son la informática y el diseño de interfaces.

6.2. Trabajo Futuro

En cuanto al trabajo futuro, este proyecto podría mejorarse para ofrecer una experiencia de usuario más dinámica y personalizada. Una de las mejoras clave sería la visualización de las recetas, haciendo que sea más interactiva y utilizando componentes personalizados en la interfaz que se asemejen a objetos de la vida cotidiana, facilitando así el proceso de creación y edición de recetas.

Para lograr este objetivo en la página web, sería útil crear más ventanas que permitan editar las recetas y los ingredientes ya existentes, o incorporar estas opciones dentro de las ventanas de gestión actuales. Además, se podría añadir una función de búsqueda en cada pantalla para facilitar la localización de información a medida que la aplicación crezca. Mientras que en la aplicación móvil, se podrían crear componentes personalizados para llenar más cómodamente el formulario de las recetas.

Otra mejora importante sería el almacenamiento persistente y la publicación web del backend. Actualmente, el backend se encuentra en local, pero para publicar la aplicación móvil se necesitará una IP pública que permita realizar peticiones. Además, el backend debería tener un sistema de almacenamiento de datos robusto para garantizar la persistencia de información en caso de fallos del sistema.

Con estas mejoras, se mejoraría significativamente la experiencia del usuario y las capacidades de gestión de los administradores. También se podría publicar la aplicación mediante Expo para que cualquier usuario pueda acceder a la plataforma.

Si quisiéramos competir con otras aplicaciones, podríamos incorporar inteligencia artificial en el buscador, permitiendo búsquedas por ingredientes y sugiriendo recetas basadas en el historial de búsqueda del usuario, mejorando así la personalización de la experiencia.

En resumen, la implementación de estas mejoras no solo aumentaría la satisfacción de los usuarios y la eficiencia de los administradores, sino que también elevaría el nivel competitivo de nuestra plataforma en el mercado de aplicaciones gastronómicas.

Bibliografía

- [1] Ivan Caraballo. Gastronomía social: La cocina tiene el poder de mejorar la sociedad. <https://es.linkedin.com/pulse/gastronom%C3%ADA-social-la-cocina-tiene-el-poder-de-ivan-caraballo>.
- [2] Isabel Conde. ¿por qué triunfa el fenómeno tasty y las vídeo-recetas de un minuto en youtube? <https://www.lavanguardia.com/comer/tendencias/20160506/401603500506/fenomeno-tasty-recetas-virales.html>.
- [3] Ministerio de Trabajo y Economía Social. Resolución de 27 de febrero de 2023, de la dirección general de trabajo, por la que se registra y publica el xx convenio colectivo nacional de empresas de ingeniería; oficinas de estudios técnicos; inspección, supervisión y control técnico y de calidad. https://www.boe.es/diario_boe/txt.php?id=BOE-A-2023-6346.
- [4] Nestle España. Página oficial. <https://www.nestlecocina.es/>.
- [5] Google. Aprenda los fundamentos de firebase. <https://firebase.google.com/docs/guides?hl=es>.
- [6] Emilio Martínez. El secreto alicantino de la cocina de cookpad, la gigantesca red social de recetas nacida en japon. https://www.elespanol.com/alicante/vivir/gastronomia/20211006/secreto-alicantino-cocina-cookpad-gigantesca-recetas-japon/617188865_0.html.
- [7] Fernando Ontiveros. ¿por qué desarrollar apps de react native en expo? <https://medium.com/leopark-lab/por-qu%C3%A9-desarrollar-apps-de-react-native-en-expo-2e1b83e4d00a>.
- [8] El pespunte. ¿cómo compartir tus conocimientos por internet? <https://www.elpespunte.es/compartir-tus-conocimientos-internet/>.
- [9] Carlos Rodríguez-Hoyos Teresa Susinos Rada. La educación inclusiva hoy. reconocer al otro y crear comunidad a través del diálogo y la participación. pages 1–17, 2011–2022.

- [10] Cetina Volkan. ¿cuáles son las ventajas de angular como framework front-end? <https://mx.devoteam.com/expert-view/cuales-son-las-ventajas-de-angular-como-framework-front-end/>.
- [11] Wildfly. Wildfly documentation. <https://docs.wildfly.org/>.
- [12] Zoom. App de recetas de cocina: Oportunidad de negocio. <https://zoomarketingdigital.com/app-de-recetas-de-cocina-oportunidad-de-negocio/>.