

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE UNA APLICACIÓN DE GESTIÓN
DE RECETAS DE COCINA**

APLICACIÓN MOVIL

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

KEVIN JOEL MORENO GARCIA

DIRECTOR: IVONNE FERNANDA MALDONADO SOLIZ

DMQ, marzo 2024

CERTIFICACIONES

Yo, Kevin Joel Moreno Garcia declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Kevin Joel Moreno Garcia

kevin.moreno01@epn.edu.ec

konkermoreno@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Kevin Joel Moreno Garcia, bajo mi supervisión.

Ivonne Fernanda Maldonado Soliz

DIRECTOR

ivonne.maldonadof@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Kevin Joel Moreno Garcia

DEDICATORIA

A mis queridos familiares, amigos y seres cercanos, quienes han estado a mi lado brindándome un apoyo incondicional a lo largo de este arduo camino. La paciencia y el aliento que me han brindado han sido mi mayor fuente de motivación para alcanzar esta meta. Cada logro obtenido es también suyo, ya que cada paso dado ha sido posible gracias a su inquebrantable apoyo y contención. Agradezco profundamente el papel fundamental que cada uno de ustedes ha desempeñado en este logro, y les dedico este éxito con todo mi cariño y gratitud. Sin su presencia y respaldo, este camino habría sido mucho más difícil. Con amor y agradecimiento, Kevin.

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a Ivonne Fernanda Maldonado Soliz, por su orientación experta, paciencia y apoyo constante a lo largo de este proyecto. Agradezco profundamente a mis amigos y compañeros por su ánimo y comprensión durante este período. Por último, pero no menos importante, agradezco a mi familia por su amor incondicional, su apoyo inquebrantable y su fe en mí. Este logro no habría sido posible sin ustedes. ¡Gracias!

Kevin.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1. Objetivo general	2
1.2. Objetivos específicos	2
1.3. Alcance	2
1.4. Marco Teórico.....	4
2 METODOLOGÍA	5
2.1. Metodología de Desarrollo	6
Roles.....	6
Artefactos	8
2.2. Diseño de interfaces.....	9
FIGMA.....	9
2.3. Diseño de la arquitectura	10
Arquitectura de Datos.....	10
Patrón arquitectónico.....	11
2.4. Herramientas de desarrollo	12
3 RESULTADOS.....	13
3.1. <i>Sprint 0</i> . Configuración del ambiente de desarrollo	13
Definición de roles	14
Recopilación de requerimientos	14
Diseño de la Base de Datos	16
Preparación y configuración de las herramientas (ambiente de desarrollo).....	16
3.2. <i>Sprint 1</i> . Autenticación y registro	18
Registro de Usuario y <i>Login</i>	18
Cierre de sesión y recuperar contraseña	22
3.3. <i>Sprint 2</i> . Creación, Edición y Eliminación de receta	24

Crear recetas	24
Editar recetas	26
Eliminar recetas.....	27
3.4. <i>Sprint 3</i> . Visualización de las recetas (Usuario).....	27
Visualizar lista de recetas	27
Visualizar detalles de la receta	28
3.5. <i>Sprint 4</i> . Sistema de búsqueda y selección de recetas	29
Búsqueda de receta por nombre	29
Selección de receta (tipo, tiempo, vegetariana).....	30
3.6. <i>Sprint 5</i> . Votos receta favorita y ventana de home	30
Creación del apartado de votos para las recetas	30
Mostrar recetas más votadas	31
3.7. <i>Sprint 6</i> . Pruebas y Despliegue.....	32
Pruebas de Funcionalidad.....	32
Pruebas de Compatibilidad.....	36
Pruebas de Aceptación:	38
Despliegue:.....	42
4 Conclusiones.....	45
5 Recomendaciones	46
6 REFERENCIAS BIBLIOGRÁFICAS	48
7 ANEXOS	50
ANEXO I.....	51
ANEXO II	52
ANEXO III	92
ANEXO IV	93

RESUMEN

El proyecto se centra en el desarrollo de una Aplicación de Gestión de Recetas, utilizando la tecnología React Native y respaldado por Firebase como *backend*. El objetivo principal es mejorar la experiencia de los usuarios, ofreciendo un entorno intuitivo y eficiente para la creación, visualización y gestión de recetas culinarias.

La aplicación permite a los usuarios registrarse y gestionar sus cuentas, diferenciando entre roles de Chef y Usuario. Los Chefs tienen la capacidad de agregar, editar y mostrar recetas, mientras que los Usuarios pueden visualizar las recetas, calificarlas y realizar búsquedas por nombre y tiempo de preparación. La interfaz ha sido diseñada para ser atractiva y fácil de usar, optimizada para diversos dispositivos y versiones de Android.

La integración con Firebase abarca aspectos clave como la autenticación de usuarios, gestión de bases de datos con Firestore y almacenamiento de archivos, garantizando un manejo seguro y eficiente de la información. Además, se ha prestado especial atención a la seguridad, rendimiento y modularidad del código, siguiendo las mejores prácticas de desarrollo.

El proyecto se basa en una arquitectura escalable para futuras expansiones y mejoras. Se han aplicado pruebas rigurosas para asegurar la funcionalidad y compatibilidad en diversas situaciones. La documentación clara y actualizada acompaña el código, facilitando el mantenimiento y comprensión de la aplicación. Es así que este proyecto representa una solución integral y moderna para entusiastas de la cocina, proporcionando una plataforma robusta y amigable que se adapta a las necesidades de Chefs y Usuarios por igual.

Este documento sigue el siguiente orden: sección 1, descripción del componente, los objetivos y el marco teórico. Sección 2, metodología Scrum, arquitectura y herramientas de desarrollo. Sección 3, resultados por *Sprint*. Y Secciones 4 y 5 conclusiones y recomendaciones obtenidas del desarrollo del proyecto.

PALABRAS CLAVE: Gestión de Recetas, Firebase, Usuarios, Chefs, Firestore, Seguridad, Rendimiento

ABSTRACT

The project focuses on the development of a Recipe Management Application, using React Native technology and supported by Firebase as a backend. The main objective is to improve the user experience, offering an intuitive and efficient environment for the creation, visualization and management of culinary recipes.

The application allows users to register and manage their accounts, differentiating between Chef and User roles. Chefs have the ability to add, edit and display recipes, while Users can view recipes, rate them and search by name and preparation time. The interface has been designed to be attractive and easy to use, optimized for various devices and versions of Android.

The integration with Firebase covers key aspects such as user authentication, database management with Firestore and file storage, guaranteeing secure and efficient handling of information. In addition, special attention has been paid to the security, performance and modularity of the code, following best development practices.

The project is based on a scalable architecture for future expansions and improvements. Rigorous testing has been applied to ensure functionality and compatibility in various situations. Clear and up-to-date documentation accompanies the code, making it easier to maintain and understand the application. Thus, this project represents a comprehensive and modern solution for cooking enthusiasts, providing a robust and friendly platform that adapts to the needs of Chefs and Users alike.

This document follows the following order: section 1, description of the component, objectives, and theoretical framework. Section 2, Scrum methodology, architecture, and development tools. Section 3, Sprint results. AND Sections 4 and 5 conclusions and recommendations obtained from the development of the project.

KEYWORDS: Recipe Management, Firebase, Users, Chefs, Firestore, Security, Performance.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el contexto actual, el uso de dispositivos móviles ha dejado de ser un privilegio para convertirse en una necesidad importante. Estos dispositivos se han integrado de manera fundamental en la vida, facilitando la ejecución de diversas actividades diarias. En este escenario, el empleo de aplicaciones específicas se ha vuelto esencial, sobre todo teniendo en cuenta la gran presencia de los teléfonos móviles en la sociedad contemporánea.

En respuesta a esta creciente dependencia de la tecnología móvil, ha surgido una demanda significativa de aplicaciones que simplifiquen y optimicen las tareas cotidianas. Una de las áreas en las que esto se evidencia de manera destacada es en la gestión de la alimentación y la preparación de recetas de cocina. La creación de una aplicación diseñada para mantener un registro continuo de los hábitos alimenticios y recetas se erige como una respuesta inteligente a esta necesidad.

Esta aplicación no solo actúa como un almacén digital de recetas, sino que también ofrece una lista en la cual se puede realizar una retroalimentación con la búsqueda o la interacción con las diferentes recetas. Por ejemplo, permite búsqueda de recetas por tiempo estimado para la preparación, cosa que resulta importante con el trajín del día a día. Esta funcionalidad promueve la eficiencia en la planificación de comidas.

La arquitectura de la aplicación se sustenta en una sólida base de datos con información detallada sobre diversas recetas, considerando varios objetivos y preferencias. La búsqueda de recetas se ha simplificado mediante parámetros personalizables, como el tipo de plato o el tiempo de preparación. Además, se ha incluido una sección dedicada a aquellos usuarios que siguen una dieta vegetariana, reconociendo la diversidad de necesidades alimenticias.

La interfaz de usuario se ha diseñado para combinar la estética agradable con la funcionalidad sencilla. La accesibilidad ha sido una prioridad, permitiendo que la aplicación sea fácilmente utilizada por un público amplio, independientemente de su nivel de familiaridad con la tecnología.

Una característica distintiva de la aplicación es la adopción de un lenguaje neutro al nombrar las recetas. Dada la variabilidad de los nombres de recetas según la región o los ingredientes disponibles, este enfoque asegura que la información sea clara y comprensible para usuarios de diferentes contextos culturales.

En resumen, esta aplicación pretende posicionarse como una herramienta práctica para la gestión de recetas y hábitos alimenticios, teniéndola como un aliado integral en el fomento de un estilo de vida más consciente y equilibrado.

1.1. Objetivo general

Desarrollar una aplicación para la gestión de recetas de cocina.

1.2. Objetivos específicos

1. Determinar los requerimientos para desarrollar la aplicación móvil.
2. Diseñar los prototipos de cada módulo de la aplicación móvil.
3. Codificar cada módulo de la aplicación móvil.
4. Verificar el correcto funcionamiento de la aplicación móvil.
5. Desplegar la aplicación móvil.

1.3. Alcance

En este proyecto se ha desarrollado una aplicación que simplifique la experiencia de búsqueda de recetas de cocina para usuarios de todos los niveles de habilidad culinaria y adaptada a su tiempo y disponibilidad, de ahí que esté disponible para cualquier persona.

La aplicación cuenta con dos roles: chef y usuario. Así mismo, se compone por el inicio de sesión, el perfil de usuario y la pantalla principal. Dependiendo del rol se tienen diferentes funcionalidades, mismas que han sido desarrolladas en base a los requerimientos levantados con la guía de Scrum, herramientas de diseño, codificación y el consumo *endpoints*, logrando de manera exitosa cumplir con los objetivos del presente proyecto.

A continuación, se presenta los roles y funcionalidades con las que la aplicación cuenta.

Roles de la aplicación móvil:

- Chef
- Usuario

El rol Chef puede:

- Registrarse.
- Autenticarse.
- Reestablecer contraseña.
- Crear recetas.
- Visualizar las diferentes recetas.
- Eliminar sus propias recetas.
- Recibir comentarios positivos o negativos de su receta.
- Recibir *Likes* en su receta

El rol Usuario puede:

- Registrarse.
- Autenticarse.
- Reestablecer contraseña.
- Visualizar las diferentes recetas.
- Ver comentarios de las recetas
- Ver las recetas más populares
- Ver instrucciones de la receta

1.4. Marco Teórico

Aplicación Móvil

Son concebidas para funcionar en un dispositivo móvil específico y requieren descarga e instalación. Se define una aplicación (App) como un software instalado en dispositivos móviles o tabletas para ayudar al usuario en tareas específicas (profesionales o de ocio y entretenimiento). Su propósito es facilitar la realización de una tarea específica o asistir en operaciones y gestiones cotidianas [1].

Base de datos no relacional

Una base de datos no relacional es un sistema de almacenamiento de datos que presenta características distintivas (almacenar datos en varios formatos, como documentos, pares clave-valor, columnares o basados en grafos, además de no requerir un esquema fijo ya que se puede agregar campos o columnas sin tener que modificar toda la base de datos, proporcionando flexibilidad para manejar datos no estructurados o semi-estructurados) que la separan del otro grupo principal de bases de datos, es decir, las relacionales [2].

Firestore Realtime Database

Es un sistema de gestión de datos NoSQL alojado en la nube que permite almacenar y sincronizar datos en tiempo real entre los usuarios [1].

Scrum

Es un modelo de gestión de proyectos basado en la metodología ágil, diseñado para ayudar a los equipos en la organización y supervisión de sus actividades mediante la aplicación de un conjunto de valores, principios y prácticas específicas [3].

React Native

Es un marco de desarrollo de código abierto desarrollado por Meta Platforms, Inc. que facilita la creación de aplicaciones para diversas plataformas, incluyendo Android, Android TV, iOS, macOS, tvOS, Web, Windows y UWP. Este marco permite a los desarrolladores utilizar React e integrar las características nativas de cada plataforma [4].

2 METODOLOGÍA

La metodología utilizada para desarrollar la aplicación móvil esta basa en un enfoque exploratorio y descriptivo, para comprender y describir las funcionalidades necesarias para que chefs y usuarios puedan visualizar recetas eficientemente y efectivas.

Tipo de trabajo

Exploración de las necesidades y expectativas de los usuarios (chefs y usuarios), mediante entrevistas para comprender las necesidades y expectativas de los interesados en cuanto a recetas. Identificando así, qué buscan en una herramienta de creación de recetas, qué funcionalidades consideran importantes y cómo les gustaría interactuar con la plataforma.

- Descripción detallada de las funcionalidades del componente de creación de recetas, que incluye creación de recetas, acciones que ofrece los usuarios y la forma en que interactúan entre sí para proporcionar una experiencia completa y satisfactoria.

Técnica aplicada

Para desarrollar el componente de creación de recetas, se ha empleado una combinación de tecnologías centradas en aplicaciones móviles, como React Native y la integración con la plataforma de servicios en la nube Firebase. A continuación, se detallan las etapas y actividades realizadas:

- Definición de Requisitos y Funcionalidades:
 - Trabajo conjunto entre todos los interesados para identificar los requisitos y funcionalidades clave.
 - Uso de técnicas ágiles, como historias de usuario, para capturar los detalles de las características esperadas.
- Selección de Tecnologías:
 - Se ha evaluado distintas tecnologías, optando por React Native por su eficiencia en el desarrollo multiplataforma.

- Firebase elegido como el *backend* por su fácil integración, escalabilidad y herramientas como autenticación, base de datos en tiempo real y almacenamiento de archivos.

2.1. Metodología de Desarrollo

En el marco de desarrollo del proyecto de recetas, se ha adoptado metodologías ágiles, caracterizadas por su enfoque flexible, colaborativo y adaptable. Estas metodologías ponen énfasis en la entrega temprana y continua de software funcional, así como en la capacidad de respuesta ágil a los cambios y a la retroalimentación del cliente.

Para estructurar el desarrollo del proyecto, se ha seleccionado Scrum como el marco de trabajo ágil principal. Scrum se especializa en la entrega iterativa e incremental de un producto, fomentando la colaboración, la adaptabilidad y una respuesta rápida a los cambios. Facilita una visión global del progreso del proyecto, permitiendo la inspección constante para tomar decisiones informadas y evitar posibles obstáculos graves durante el desarrollo.

Roles

Dentro del marco de trabajo Scrum, se definen tres roles clave para el desarrollo del proyecto de creación de recetas:

Product Owner

- Responsable: Kevin Moreno
- Funciones:
 - Definir y priorizar requisitos y funcionalidades en el *Product Backlog*.
 - Mantener una comunicación constante con el equipo de desarrollo para comprender las necesidades y expectativas.
 - Tomar decisiones sobre qué funcionalidades se deben desarrollar en cada *Sprint*, basándose en el valor y la importancia para el cliente.
- Establecer criterios de aceptación claros para cada funcionalidad.

Scrum Master

- Responsable: Ing. Ivonne Maldonado
- Funciones:
 - Garantizar que el equipo de desarrollo entienda y aplique correctamente los principios y prácticas de Scrum.
 - Eliminar obstáculos y problemas que puedan afectar el progreso del equipo.
 - Facilitar reuniones de Scrum, promoviendo un ambiente de trabajo colaborativo y de mejora continua.
 - Ayudar al equipo a definir y alcanzar sus objetivos, fomentando la autonomía y autogestión.

Development Team

- Responsable: Kevin Moreno
- Funciones:
 - Implementar las funcionalidades de la aplicación móvil de manera colaborativa.
 - Planificar y ejecutar el trabajo necesario para entregar las funcionalidades definidas en el *Sprint*.
 - Participar activamente en reuniones de Scrum.
 - Aclarar requisitos y validar las funcionalidades desarrolladas.

A continuación, la **Tabla 2.1.** muestra los roles asignados en el proyecto.

Tabla 2.1 Roles designados

ROLES	NOMBRES
<i>Product Owner</i>	Sr. Kevin Moreno
<i>Scrum Master</i>	Ing. Ivonne Maldonado
<i>Development Team</i>	Sr. Kevin Moreno

Artefactos

Son todas las herramientas que permiten dar un seguimiento y respaldar el cumplimiento del proyecto. En los artefactos se dejan claros los requisitos, las prioridades y el progreso en si del proyecto, lo que facilita la colaboración y la toma de decisiones por parte del equipo Scrum.

Recopilación de Requerimientos

Está basada en la comunicación constante y la colaboración entre el *Product Owner* y el *Development Team*. Sigue un proceso iterativo y colaborativo que se enfoca en identificar las necesidades clave del cliente y transformarlas en elementos del *Product Backlog*. El listado de los requerimientos que se han levantado puede observarse en él **ANEXO II**.

Historias de Usuario

Es el artefacto principal para capturar y entender de manera clara los requisitos y necesidades del usuario. La **Tabla 2.2** muestra un ejemplo de Historia de Usuario, las restantes se encuentran en él **ANEXO II**.

Tabla 2.2 Formato de Historias de Usuario – RR001

HISTORIA DE USUARIO	
Identificador (ID): RR001	Usuario: Chef, Usuario
Nombre Historia: Registrar cuenta (selección de tipo de usuario)	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	

Descripción: El usuario requiere de la creación de su cuenta para poder ingresar de manera correcta dentro de la aplicación al igual que su selección del tipo de usuario que requiere (Chef o Usuario)
--

Observación: El usuario tendrá diferentes tipos de pantallas y acciones al momento de seleccionar su tipo de usuario.
--

Product Backlog

Es el artefacto que permite gestionar los requisitos y funcionalidades del proyecto, en este se prioriza el valor de los elementos en relación con las necesidades del negocio. En el **Product Backlog** se puede observar el *Product Backlog* de este proyecto.

Sprint Backlog

Este artefacto permite tener organizado y planificado el trabajo durante un *Sprint* específico. Incluye una lista de tareas seleccionadas del *Product Backlog* para abordar durante cada *Sprint*, ayudando a mantener el enfoque en la entrega de los elementos planificados. En el **Sprint Backlog** se puede observar el *Sprint Backlog* de este proyecto.

2.2. Diseño de interfaces

Su construcción sirve para que el *Product Owner* tengan una mejor claridad sobre las necesidades convertidas en funcionalidades del software.

FIGMA

Es una herramienta de prototipado y editor de gráficos vectoriales, se trabaja y almacena los diseños en web. Por su facilidad de empleo ha sido utilizada para prototipar cada una de las pantallas de la aplicación. La **Figura 2.1 - Ejemplo de diseño** – presenta un ejemplo de prototipo de pantalla. En el **Diseño de Interfaces** se puede observar todos los diseños realizados.



Figura 2.1 - Ejemplo de diseño – *Login*

2.3. Diseño de la arquitectura

La planificación del software, un término originado en la década de los 60, implica el uso de modelos, patrones y conceptos abstractos como parte del proceso de desarrollo de software de cierta complejidad. Este enfoque se considera como un paso inicial antes de la implementación del software [5].

Arquitectura de Datos

La arquitectura de datos describe el cómo los datos son gestionados dentro de un software. Ayuda a entender la forma en que los datos fluyen en la Base de Datos del sistema.

La **Figura 2.2** muestra las colecciones que se han utilizado como modelo de Base de Datos de para la aplicación móvil.

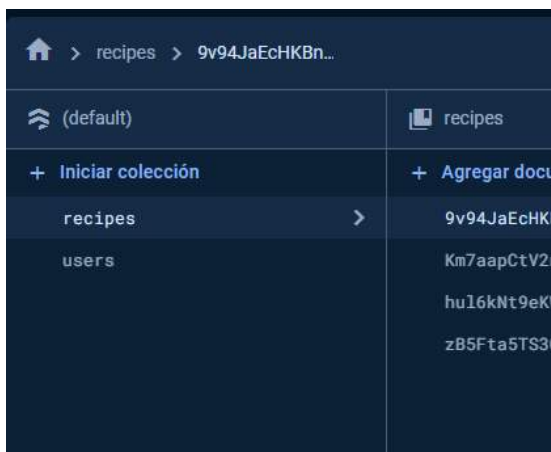


Figura 2.2 - Colecciones base de datos

Patrón arquitectónico

Un modelo arquitectónico es una solución genérica y reutilizable para un problema común en la arquitectura de software en un contexto específico. Estos modelos son similares a los patrones de diseño de software, pero abarcan un espectro más amplio [6].

Modelo-Vista-Controlador (MVC):

- **Modelo:** Este componente administra la información necesaria para el funcionamiento del software, incluyendo el manejo de accesos y consultas a bases de datos. Envía a la “vista” la información requerida en cada momento para su visualización, generalmente al usuario.
- **Vista:** Presenta el “modelo” (información y lógica de negocio) en un formato adecuado para interactuar, generalmente la interfaz de usuario.
- **Controlador:** Responde a eventos, normalmente acciones del usuario, y realiza solicitudes al “modelo” cuando se requiere información (por ejemplo, inicio de sesión). También envía comandos a la “vista” asociada si se solicita un cambio en la presentación del “modelo”, actuando como intermediario entre la “vista” y el “modelo”.

El patrón arquitectónico de la aplicación móvil se desarrolla con MVC ya que se realiza peticiones por parte del usuario para lograr la comunicación entre la base

de datos, el modelo y la vista, y así poder terminar y empezar con el mismo usuario (Figura 2.3).

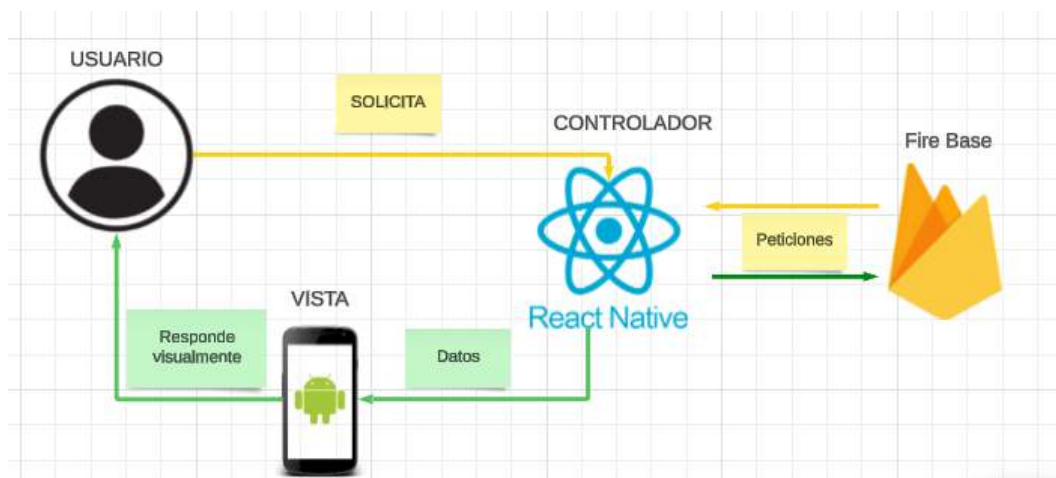


Figura 2.3 - Patrón arquitectónico – Aplicación móvil

2.4. Herramientas de desarrollo

Las herramientas que se han seleccionado se ajustan a las necesidades del proyecto y el conocimiento del equipo de desarrollo. La **Tabla 2.3 Herramientas para el desarrollo de la Aplicación** presenta las herramientas que se han utilizado en este proyecto.

Tabla 2.3 Herramientas para el desarrollo de la Aplicación

Herramienta	Justificación
React Native	Es una elección fundamental para el desarrollo móvil, permite crear aplicaciones nativas para iOS y Android desde un único código base en JavaScript. Esto mejora la eficiencia del desarrollo y garantiza una experiencia de usuario nativa. Su eficiencia en el desarrollo ha permitido compartir un código base entre plataformas, resultando en una reducción de costos y tiempo de desarrollo.
Firebase	Ofrece una amplia variedad de servicios en la nube que son esenciales para el desarrollo de aplicaciones móviles. La autenticación, la base de datos en tiempo real, el almacenamiento

	en la nube y otras características simplifican el <i>backend</i> , permitiendo una implementación más rápida y escalable.
React Navigation	Proporciona una solución de navegación intuitiva. Con soporte para navegación en pila (<i>Stack Navigation</i>), facilita la transición entre pantallas y la gestión del estado de la aplicación. Ha facilitado la creación de pantallas ofreciendo una navegación clara y sencilla entre ellas.
React Context API	Permite compartir estado entre componentes sin necesidad de pasar explícitamente props a través de múltiples niveles de anidación. Facilita la gestión global del estado de la aplicación, siendo útil en la aplicación donde múltiples componentes pueden compartir y actualizar el estado de manera eficiente.
React Native Image Picker	Para la carga de imágenes, React Native Image Picker es una herramienta esencial. Permite a los usuarios seleccionar imágenes desde su dispositivo, facilitando la subida de imágenes, mejorando la interactividad y la participación con la aplicación.
React Native Text Input	La entrada de texto es fundamental en las aplicaciones, React Native Text Input ha proporcionado componentes de entrada de texto personalizables facilitando la recopilación de información textual de los usuarios.
Dependencias Básicas (Axios, etc.)	La inclusión de dependencias básicas como Axios simplifica las solicitudes HTTP. Estas herramientas básicas son fundamentales para mejorar la eficiencia y la funcionalidad general de la aplicación.

3 RESULTADOS

A continuación, en este apartado se presenta los resultados del desarrollo de cada *Sprint* de la aplicación móvil.

3.1. *Sprint* 0. Configuración del ambiente de desarrollo

Para este *sprint* se tiene los siguientes resultados:

- Definición de roles
- Recopilación de requerimientos
- Diseño de la Base de datos

- Preparación y configuración de las herramientas (ambiente de desarrollo)

Definición de roles

Se han definido dos roles de usuario: el Chef, quien tiene la creación de recetas al igual que su propia edición y la eliminación de esta, si así lo desea y el Usuario, quien puede realizar una búsqueda o vista de diferentes tipos de recetas, al igual que el poder votar por su receta favorita (ver **Figura 3.1**).

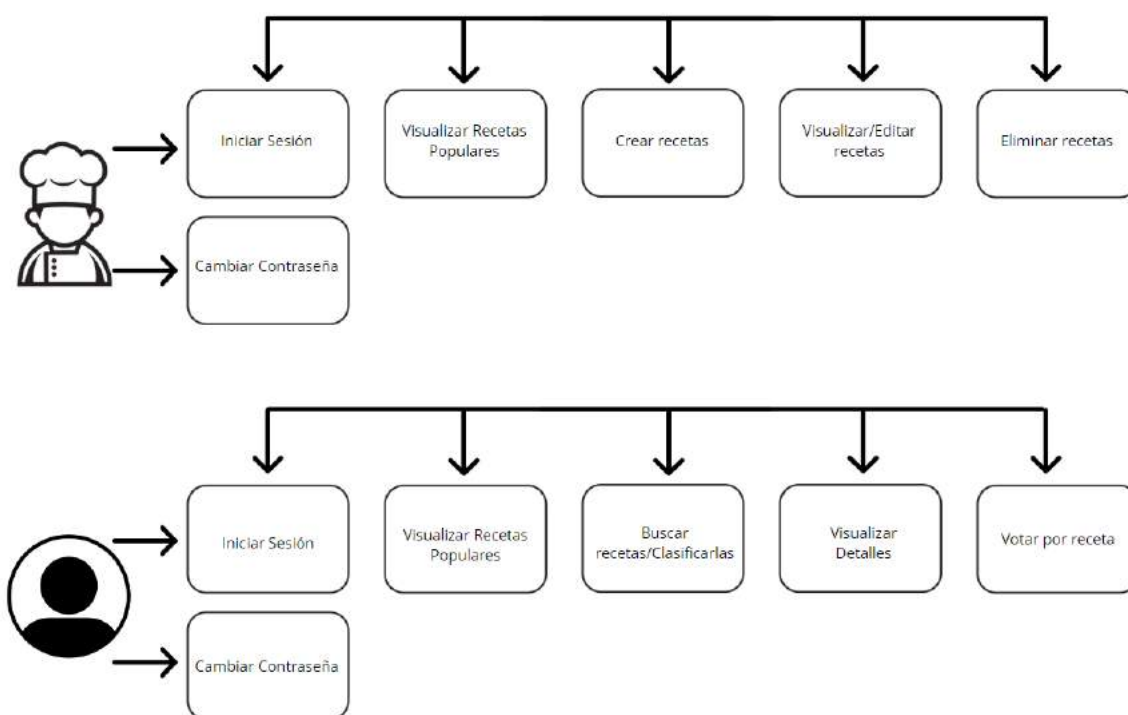


Figura 3.1 – Roles

Recopilación de requerimientos

Registro de usuario

Se permite registrar a un usuario con la opción de seleccionar que tipo de usuario desea ser (chef o usuario).

Iniciar y cerrar sesión.

El inicio de sesión permite ingresar a la aplicación con el usuario creado al igual que con el rol seleccionado al momento del registro. Así también se cuenta con la funcionalidad de cerrar sesión para terminar el uso de la aplicación por un determinado usuario.

Creación, edición y eliminación de recetas

El usuario con rol de Chef tiene permitido la creación de su propia receta, llenando diferentes características, como por ejemplo el título de la receta, su descripción, pasos, tiempo (15 min, 15 min, - 15 min) e ingredientes. Así también en cualquier momento el usuario puede realizar una edición o eliminación de la receta si así lo desea.

Visualización de recetas (Chef)

El Chef cuenta con la funcionalidad de visualización de sus propias recetas, así también tiene la opción de observar las recetas que puedan ser las “votadas”.

Visualización de recetas (Usuario)

El Usuario cuenta con la funcionalidad de visualizar las diferentes recetas ingresadas por los diferentes chefs para así poder encontrar alguna que sea de su agrado.

Calificación de las recetas.

Al Usuario se le permite marcar como “Favorita” una o varias recetas, siendo parte de la estadística para lograr el cálculo de recetas más votadas en la ventana de *home*.

Visualización de las recetas más votadas.

El Chef y el Usuario pueden visualizar las recetas más votadas, en la ventana de *home* para así poder acceder a estos de manera mucho más rápida y flexible.

Visualización de recetas por su diferente tipo.

El Usuario puede realizar una búsqueda dependiendo del tipo de plato que desee (Plato fuerte, Entrada, Postre).

Visualización de recetas vegetarianas

Se cuenta con una división de las recetas, en las cuales solo se muestran las recetas que son vegetarianas, así el Usuario puede acceder a estas con mayor facilidad, mediante una búsqueda por el nombre.

Visualización de receta por tiempo de preparación

Se cuenta con una división de las recetas en funcional tiempo de preparación.

Diseño de la Base de Datos

Con la ayuda de Firebase se ha creado y configurado la base de datos que soporta al proyecto. La **Figura 3.2** muestra dicho diseño.

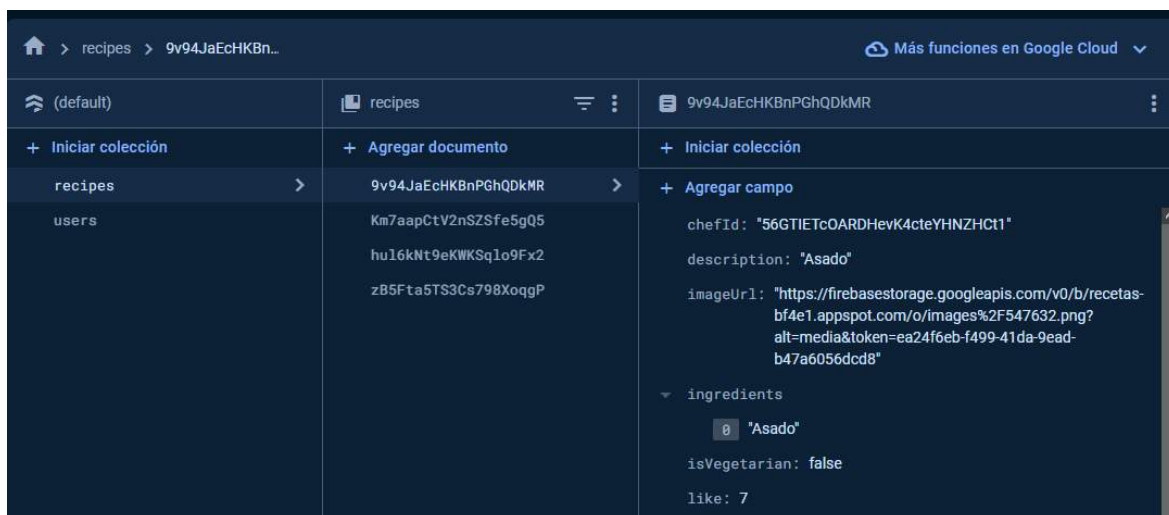


Figura 3.2 - Diseño de la Base de Datos

Preparación y configuración de las herramientas (ambiente de desarrollo)

Es de suma importancia el poder realizar y entender la instalación de las diferentes herramientas que han sido utilizadas durante la etapa de desarrollo.

- Descarga e Instalación JDK (**Figura 3.3**).

```
C:\Users\CLEAR MINDS>java --version
java 11.0.21 2023-10-17 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.21+9-LTS-193)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.21+9-LTS-193, mixed mode)
C:\Users\CLEAR MINDS>
```

Figura 3.3 - JDK versión

- Descarga e instalación de Node (**Figura 3.4**).

```
C:\Users\CLEAR MINDS>npm --version
10.2.4

C:\Users\CLEAR MINDS>
```

Figura 3.4 - Node versión

- Descarga e instalación de Android Studio (**Figura 3.5**).

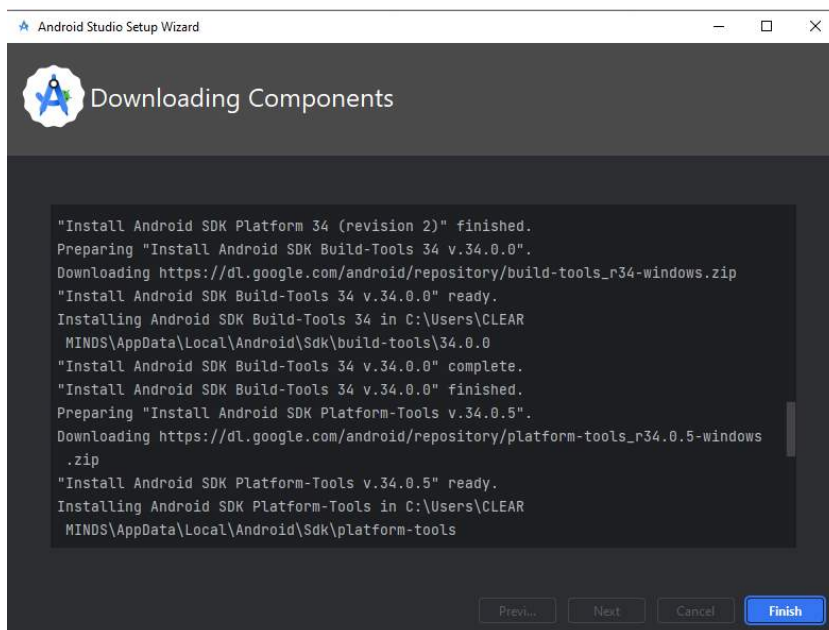


Figura 3.5 - Instalación Android Studio

- Configuración de Variables de entorno, configuración Path (**Figura 3.6**).

```
C:\Program Files\nodejs\
C:\Program Files\Java\jdk-11\bin
%ANDROID_HOME%\tools
%ANDROID_HOME%\platform-tools
```

Figura 3.6 – Path del sistema

- Configuración Variables del sistema (**Figura 3.7**).

Variable	Valor
ANDROID_HOME	C:\Users\CLEAR MINDS\AppData\Local\Android\Sdk
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Java\jdk-11\
NUMBER_OF_PROCESSORS	12

Figura 3.7 – Configuración variables del sistema

3.2. *Sprint 1. Autenticación y registro*

Para este *sprint* se tiene los siguientes resultados:

- Registro de Usuario y *Login*
- Cierre de sesión y recuperar contraseña

Registro de Usuario y *Login*

La funcionalidad de registro de usuario se genera mediante dependencias de uso de Firebase en el entorno de desarrollo de React Native, por lo cual se genera una colección llamada “users”. La **Figura 3.8** muestra la colección donde se guarda la información del usuario, como el tipo de rol que maneja, así también se guarda en la ventana de Autenticación de Firebase (ver **Figura 3.9**).

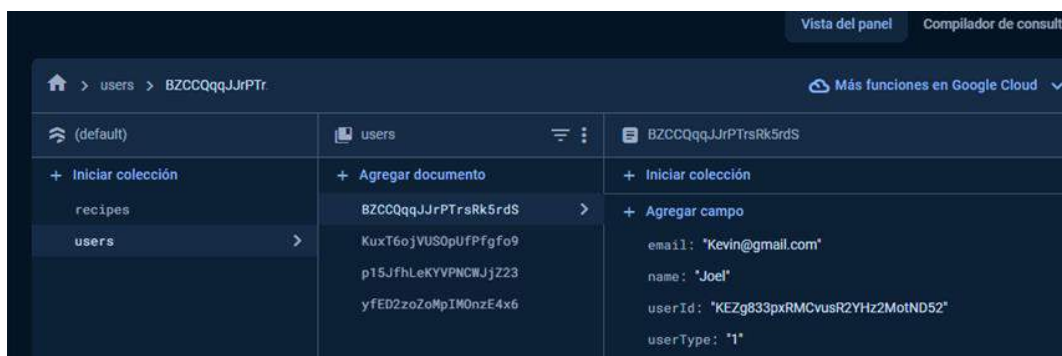


Figura 3.8 - Colección users

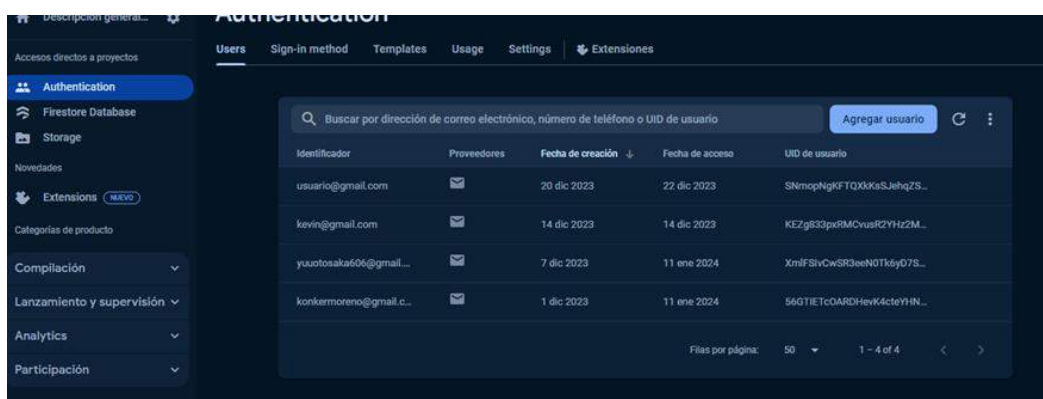


Figura 3.9 - Authentication Firebase

Para la creación de los usuarios y la selección del tipo, se ha hecho uso de la importación proporcionada por la biblioteca fire/auth (ver **Figura 3.10**).

```
import { firebaseConfig } from '../utils/firebase-config';
import { createUserWithEmailAndPassword, getAuth, signInWithEmailAndPassword, } from 'firebase/auth';
import { Alert } from 'react-native';
import { SegmentedButtons } from 'react-native-paper';
```

Figura 3.10 – Importación biblioteca fire/auth

Se crean las diferentes variables (ver **Figura 3.11**) a usar para realizar el registro del usuario de manera que se pueda guardar en la base de datos (colección users y propiedades de Firebase) sin ningún problema.

```
//variables para el login
const [email, setEmail] = React.useState('');
const [password, setPassword] = React.useState('');
const [confirmPassword, setConfirmPassword] = React.useState('');
const [nombre, setNombre] = React.useState('');
const [accType, setAccType] = React.useState('');
const [showPassword, setShowPassword] = React.useState(false); // Nuevo estado

const [emailError, setEmailError] = useState('');
const [passwordError, setPasswordError] = useState('');
const [confirmPasswordError, setConfirmPasswordError] = useState('');
const [nombreError, setNombreError] = useState('');
const [accTypeError, setAccTypeError] = useState('');

const togglePasswordVisibility = () => {
  // Cambia el estado para mostrar u ocultar la contraseña
  setShowPassword(!showPassword);
};
```

Figura 3.11 – Creación de variables para registro de usuarios

Para lograr el registro de un usuario se cuenta con la funcionalidad correspondiente (ver **Figura 3.12**), el componente visual (ver **Figura 3.13**) teniendo como resultado la vista para el usuario final (ver **Figura 3.14**).

```

createUserWithEmailAndPassword(auth, email, password)
  .then(async (userCredential) => {
    console.log('Creando Cuenta -----');
    const user = userCredential.user;
    console.log(user);

    // Guardar información adicional en Firestore
    const userData = {
      name: nombre, // Reemplaza con el nombre real del usuario
      userType: accType, // Reemplaza con el tipo de usuario
      email: email,
      userId: user.uid,
    };

    // Añadir documento a la colección 'users' en Firestore
    const docRef = await addDoc(collection(db, 'users'), userData);
    console.log('Documento agregado con ID: ', docRef.id);

    navigation.navigate('Login');
  })
  .catch(error => {
    console.log("error!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
    Alert.alert(error.message);
  });
};

```

Figura 3.12 – Funcionalidad registro de usuario

```

<View style={{ flex: 1 }}>
  <KeyboardAvoidingView
    behavior={Platform.OS === 'ios' ? 'padding' : 'height'}
    style={{ flex: 1 }}
  >
    <Image source={require('../assets/fondo.jpg')} style={styles.container} />
    <View style={styles.body}>
      <View style={styles.formContainer}>
        <Text style={styles.text}>Nombre:</Text>
        <TextInput
          style={styles.input}
          placeholder="Juan"
          onChangeText={(text) => setNombre(text)}
        />
        <Text style={styles.errorText}>{nombreError}</Text>
        <Text style={styles.text}>Email:</Text>
        <TextInput
          style={styles.input}
          placeholder="alguien@gmail.com"
          onChangeText={(text) => setEmail(text)}
        />
        <Text style={styles.errorText}>{emailError}</Text>
        <Text style={styles.text}>Contraseña:</Text>
        <View style={styles.inputContainer}>
          <TextInput
            style={styles.input}
            placeholder="Contraseña"
            onChangeText={(text) => setPassword(text)}
            secureTextEntry={!showPassword}
          />
          <TouchableOpacity
            onPress={togglePasswordVisibility}
            style={styles.iconContainer}
          >
            <Icons name={showPassword ? 'eye-off' : 'eye'} size={20} color="#333" />
          </TouchableOpacity>
        </View>
        <Text style={styles.errorText}>{passwordError}</Text>
        <Text style={styles.text}>Confirmar Contraseña:</Text>
        <View style={styles.inputContainer}>
          <TextInput
            style={styles.input}
            placeholder="Confirmar Contraseña"
            onChangeText={(text) => setConfirmarContraseña(text)}
          />
          <Text style={styles.errorText}>{confirmarContraseñaError}</Text>
        </View>
      </View>
    </View>
  </View>

```

Figura 3.13 – Componente visual registro de usuario

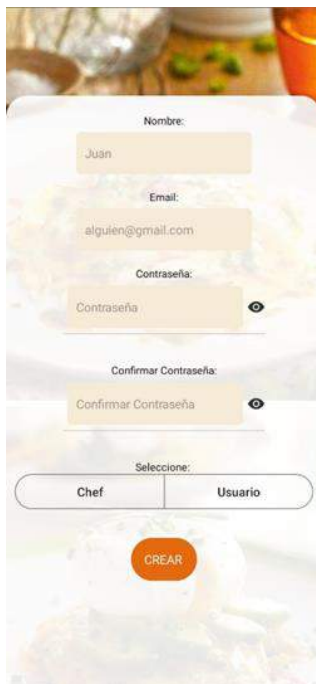


Figura 3.14 – Vista registro de usuario

Para que un usuario pueda loguearse, se ha hecho uso de diferentes tipos de dependencias las cuales permiten el proceso de *login* como tal en Firebase (ver **Figura 3.15**), además se ha desarrollado el respectivo componente visual (ver **Figura 3.16**) para tener la vista del usuario final (ver **Figura 3.17**).

```

export const LoginScreen = () => {
  //variables para el login
  const [email, setEmail] = React.useState("");
  const [password, setPassword] = React.useState("");
  const [showPassword, setShowPassword] = React.useState(false); // Nuevo estado
  const [loading, setLoading] = React.useState(false);

  const togglePasswordVisibility = () => {
    // Cambia el estado para mostrar u ocultar la contraseña
    setShowPassword(!showPassword);
  };

  const { login } = useAuth();

  const handleSignIn = async () => {
    const app = initializeApp(firebaseConfig);
    const auth = getAuth(app);

    try {
      setLoading(true);
      const userCredential = await signInWithEmailAndPassword(auth, email, password);
      const user = userCredential.user;

      // Obtener el userType desde Firestore
      const userType = await getUserType(user.uid);

      // Almacenar en el contexto
      login(user, userType);

      navigation.navigate("Home");
      console.log("El usuario es tipo: " + userType);
    } catch (error) {
      console.log("Error al iniciar sesión:", error.message);
      Alert.alert("Error", "Correo o contraseña son incorrectas");
    } finally {
      setLoading(false);
    }
  }
}

```

Figura 3.15 – Funcionalidad *login*

```

const navigation = useNavigation();

return (
  <View style={({ flex: 1 })}>
    <Image source={require("../assets/fondo.jpg")} style={styles.container} />
    <View style={styles.body}>
      <View style={ { flex: 8.5, justify-content: 'center' } }>
        <Text style={ { fontSize: 24, } }>INICIO DE SESION</Text>
      </View>

      <View style={ { alignItems: 'center', flex: 2 } }>
        <Text style={styles.text}>Email</Text>
        <TextInput style={styles.input} placeholder="alguien@gmail.com" onChangeText={(text) => setEmail(text)} />
      </View>

        <Text style={styles.text}>Contraseña</Text>
        <View style={styles.inputContainer}>
          <TextInput style={styles.input} placeholder="Contraseña" secureTextEntry={showPassword} onChangeText={(text) => setPassword(text)} />
        </View>
        <TouchableOpacity onPress={togglePasswordVisibility} style={styles.iconContainer}>
          <Icon name={showPassword ? 'eye-off' : 'eye'} size={20} color="#333" />
        </TouchableOpacity>
      </View>

      <View style={ { marginBottom: 30, flexDirection: 'row', marginTop: 20 } }>
        <TouchableOpacity style={styles.Button} onPress={handleSignIn}>
          <Text style={ { color: 'white' } }>Iniciar Sesión</Text>
        </TouchableOpacity>

        <TouchableOpacity style={styles.Button} onPress={() => { navigation.navigate('Register') } }>
          <Text style={ { color: 'white' } }>Crear Cuenta</Text>
        </TouchableOpacity>
      </View>
    </View>
    <Loading visible={loading} text="Iniciando sesión..." />
  </View>
);

```

Figura 3.16 - Componente visual *login*



Figura 3.17 – Vista *login*

Cierre de sesión y recuperar contraseña

El usuario puede realizar lo que es el cierre de sesión, ya que en la respectiva pantalla de Home (Chef o Usuario) (ver **Figura 3.18**), se encuentra un botón Cerrar sesión el cual le permite realizar la acción de cierre de sesión.



Figura 3.18 – Home, botón cerrar sesión

El usuario tiene la posibilidad de realizar la recuperación o cambio de contraseña, en la pantalla de inicio de sesión se encuentra una selección que le permite realizar esta acción, al momento de empezar con el cambio de contraseña se muestra un mensaje que explica que el cambio de contraseña se desarrolla mediante un correo enviado a la dirección de correo proporcionada (ver **Figura 3.19**).

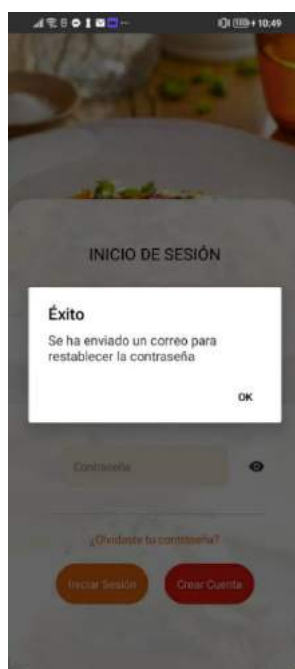


Figura 3.19 - Mensaje Recuperación contraseña.

Al momento de abrir el enlace que se proporciona por correo electrónico, se redirige a una página en la cual se permite realizar el cambio de la contraseña para el correo proporcionado (ver **Figura 3.20**).



Figura 3.20 – Pagina de cambio de contraseña

3.3.Sprint 2. Creación, Edición y Eliminación de receta

Para este *sprint* se tiene los siguientes resultados:

- Crear recetas
- Editar recetas
- Eliminar recetas

Crear recetas

Para crear una receta se cuenta con una función “HandleAddrecipe” para poder agregar la receta en la colección “recipes” (ver **Figura 3.21**) esta depende de un formulario donde se ingresan los datos de las recetas: título, descripción, ingredientes, pasos, tipo de receta, tiempo e imagen, cada campo cuenta con las respectivas validaciones (ver **Figura 3.22**).

```

const handleAddRecipe = async () => {
  try {
    setloading(true);

    if (!title || title.length > 50 || !description || description.length > 250 ||
      steps.some(step => !step || step.length > 150) || ingredients.some(ingredient => !ingredient || ingredient.length > 50)) {
      setloading(false);
      return Alert.alert('Error', 'Por favor, completa todos los campos antes de guardar tu receta.');
```

Figura 3.21 - Funcionalidad crear recetas

The screenshot shows a mobile application interface for creating a recipe. The form is styled with an orange header and contains the following elements:

- Título (máx. 50 caracteres):** A text input field.
- Descripción (máx. 250 caracteres):** A text input field.
- Ingrediente 1:** A text input field with a trash icon and a plus icon to its right.
- Paso 1 (máx. 150 caracteres):** A text input field with a trash icon and a plus icon to its right.
- Tipo de receta:** A dropdown menu with the selected option "Plato Fuerte".
- 15 min:** A dropdown menu showing a time value.
- ¿Es vegetariana?:** A dropdown menu with the selected option "No".
- Image:** A placeholder image of a bowl of soup with chicken and vegetables.

Figura 3.22 – Vista crear recetas

Editar recetas

Para realizar la edición de una receta se hace uso de una función que realiza un “update” dependiendo del id (ver **Figura 3.23**), esto permite traer los datos y mostrarlos en un formulario donde se puede realizar los cambios deseados por el usuario (ver **Figura 3.24**) y almacenarlos en la base de datos.

```

let newImageUrl = existingImageUrl; // Por defecto, mantener la imagen existente

if (imagen) {
  const randomCode = getRandomCode();
  const imageName = `${randomCode}.png`;

  const storage = getStorage();
  const imageRef = ref(storage, `images/${imageName}`);
  const blob = await (await fetch(imagen)).blob();

  const uploadTask = uploadBytesResumable(imageRef, blob, {
    contentType: 'image/png',
  });

  await uploadTask;

  // Obtener la URL de la nueva imagen
  newImageUrl = await getDownloadURL(uploadTask.snapshot.ref);
}

// Antes de actualizar el documento, eliminar la imagen existente si es necesario
if (existingImageUrl && imagen && existingImageUrl !== imagen) {
  const existingImageRef = ref(storage, `images/${existingImageUrl}`);
  await deleteObject(existingImageRef);
}

const recipeRef = doc(db, 'recipes', recipeId);
const updatedRecipeData = {
  title,
  description,
  steps,
  ingredients,
  time,
  isVegetarian,
  imageUrl: newImageUrl, // Usar la nueva URL o la existente
  recipeType,
};

await updateDoc(recipeRef, updatedRecipeData);

navigation.navigate('Home');
} catch (error) {
  console.error('Error al actualizar la receta:', error);
  // Puedes agregar un manejo específico del error aquí si es necesario
} finally {
  setLoading(false); // Desactivar el estado de carga incluso en caso de error
}

```

Figura 3.23 – Funcionalidad editar receta

Figura 3.24 – Vista editar receta

Eliminar recetas

Para realizar lo que es el proceso de la eliminación de una de las recetas, solo puede hacerlo el chef, y únicamente sus propias recetas (**Figura 3.25**). Al igual que al editar, se debe obtener el id de la receta seleccionada, para así poder eliminarla de manera correcta en la base de datos.

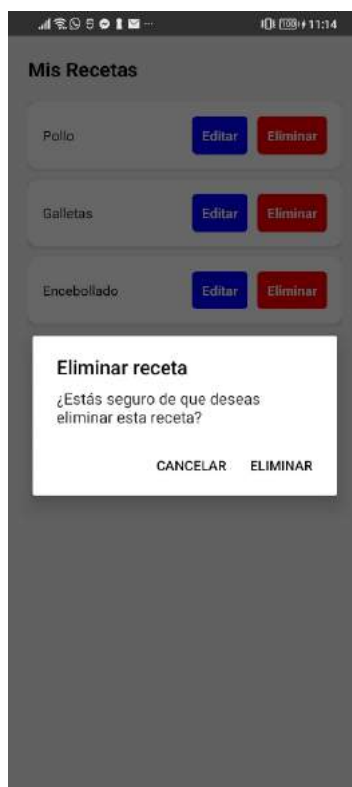


Figura 3.25 – Vista Eliminar receta

3.4. *Sprint 3*. Visualización de las recetas (Usuario)

Para este *sprint* se tiene los siguientes resultados:

- Visualizar lista de recetas
- Visualizar detalles de la receta

Visualizar lista de recetas

Se ha creado un componente encargado de traer la lista de recetas junto con la imagen correspondiente (ver **Figura 3.26**) dando como resultado la vista de la lista de recetas (ver **Figura 3.27**).

```

const renderRecipeItem = ({ item }) => {
  <View style={styles.recipeCard}>
    <Text style={styles.recipeTitle}>{item.title}</Text>
    <Image source={{ url: item.imageUrl }} style={styles.recipeImage} />
    <Text style={styles.recipeDescription}>{item.description}</Text>
    <Text style={styles.recipeSubtitle}>Ingredientes:</Text>
    <FlatList
      data={item.ingredients}
      keyExtractor={(ingredient, index) => index.toString()}
      renderItem={({ item: ingredient, index }) => {
        <Text key={index} style={styles.recipeStep}>{"${index + 1}. ${ingredient}"</Text>
      }}
    />
    <Text style={styles.recipeSubtitle}>Pasos:</Text>
    <FlatList
      data={item.steps}
      keyExtractor={(step, index) => index.toString()}
      renderItem={({ item: step, index }) => {
        <Text key={index} style={styles.recipeStep}>{"${index + 1}. ${step}"</Text>
      }}
    />
    <Text style={styles.recipeSubtitle}>Tiempo: {item.time} min</Text>
    <Text style={styles.recipeSubtitle}>¿Es vegetariana?: {item.isVegetarian ? 'Sí' : 'No'}</Text>

    <Button
      title="Ver Detalles"
      onPress={() => navigation.navigate('RecipeDetailScreen', { recipe: item })}
      color="#E5690E"
    />
  </View>
};

```

Figura 3.26 – Componente para traer la lista de recetas



Figura 3.27 – Vista lista de recetas

Visualizar detalles de la receta

Para visualizar el detalle de la receta se ha creado una pantalla que presenta toda la información de la receta, esto se logra al traer los datos de la receta seleccionada usando el id de la misma (ver **Figura 3.28**).



Figura 3.28 – Vista detalles de receta

3.5. *Sprint 4*. Sistema de búsqueda y selección de recetas

Para este *sprint* se tiene los siguientes resultados:

- Búsqueda de receta por nombre
- Selección de receta (tipo, tiempo, vegetariana)

Búsqueda de receta por nombre

Para lograr una búsqueda por nombre, lo que se realiza es una búsqueda mediante cada carácter el título de la receta (ver **Figura 3.29**).



Figura 3.29 - Búsqueda de receta por nombre

Selección de receta (tipo, tiempo, vegetariana)

Para la separación de las recetas de acuerdo con el tipo, el tiempo y si es vegetariana o no, se cuenta con la distinción correspondiente en la base de datos lo que permite que el momento de traer los datos de la receta solo se muestre la información necesaria dependiendo de la selección del usuario (ver **Figura 3.30** y **Figura 3.31**).



Figura 3.30 – Selección de receta por tipo

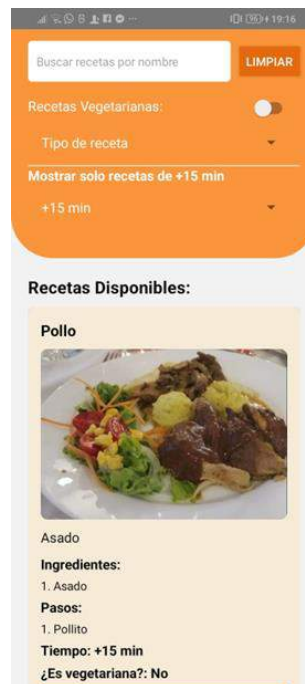


Figura 3.31 – Selección de receta por tiempo

3.6. *Sprint 5*. Votos receta favorita y ventana de home

Para este *sprint* se tiene los siguientes resultados:

- Creación del apartado de votos para las recetas
- Mostrar recetas más votadas

Creación del apartado de votos para las recetas

En las recetas se ha agregado la funcionalidad de “votar” asignar como favorita dicha receta, para ello se cuenta con un botón en la parte inferior de los detalles, esto se logra usando el contexto para guardar el estado del botón al momento de dar un *like* en la receta (ver **Figura 3.32** y **Figura 3.33**).



Figura 3.32 – Receta SIN marcación como Favorita



Figura 3.33 - Receta CON marcación como Favorita

Mostrar recetas más votadas

Para mostrar las tres recetas más votadas se cuenta con una función que permite traer las mismas (ver **Figura 3.34**) y mostrarlas a los usuarios en las diferentes pantallas de home, tanto para usuario como para chef, en la sección de la izquierda de la pantalla Home (ver **Figura 3.35** y **Figura 3.36**).

```
const fetchPopularRecipes = async () => {
  try {
    const recipesQuery = query(collection(db, 'recipes'), orderBy('like', 'desc'), limit(3));
    const recipesSnapshot = await getDocs(recipesQuery);
    const popularRecipeData = recipesSnapshot.docs.map((doc) => ({ id: doc.id, ...doc.data() }));
    setPopularRecipes(popularRecipeData);
  } catch (error) {
    console.error('Error al obtener las recetas populares:', error);
  }
};

// Utiliza useEffect para recargar las recetas populares cada vez que se visite la ventana de Home
useEffect(() => {
  fetchPopularRecipes();
}, []);
```

Figura 3.34 – Función para traer las 3 recetas más votadas



Figura 3.35 – Recetas más votadas -
Chef



Figura 3.36 - Recetas más votadas -
Usuario

3.7. *Sprint 6. Pruebas y Despliegue*

Para este *sprint* se tiene los siguientes resultados:

- Pruebas de funcionalidad
- Pruebas de compatibilidad
- Pruebas de aceptación
- Despliegue

Pruebas de Funcionalidad

Las pruebas de funcionalidad son un conjunto de pruebas que se llevan a cabo en el software con el objetivo de comprobar si cumple con los requisitos funcionales establecidos. Estas pruebas se enfocan en evaluar las funciones y características del sistema, asegurándose de que todas las acciones que el software debe realizar funcionen correctamente. Su propósito es verificar que el software se comporte de acuerdo con lo esperado y que cumpla con las expectativas del usuario. Es importante destacar que estas pruebas son fundamentales para garantizar el correcto funcionamiento del sistema y evitar posibles errores o fallos en su desempeño [9].

Las pruebas de funcionalidad son pruebas que se han realizado de forma manual en base a los requerimientos, a continuación, se presenta un ejemplo de las pruebas de funcionalidad realizadas, las pruebas restantes se encuentran en el **ANEXO II**

Requisito: Registrarse

Descripción: La aplicación móvil debe permitir que un usuario, para este caso con rol Chef, se registre en la plataforma.

Usuarios Involucrados: Chef

Escenarios de Uso:

- El Chef accede a la página de registro.
- Ingresa la información requerida, como nombre, correo electrónico, contraseña y confirmación de contraseña.
- Selecciona el tipo de usuario Chef.
- Acepta términos y condiciones.
- Envía el formulario de registro.

Pruebas Asociadas:

- Ingreso a la pantalla de registro (ver **Figura 3.37**)

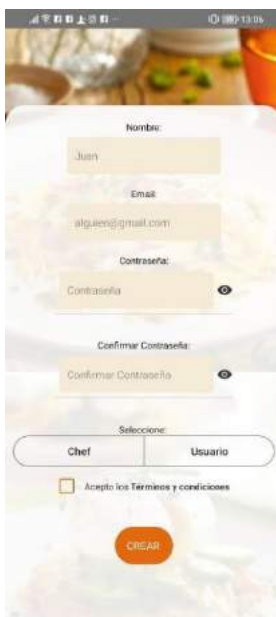


Figura 3.37 – Pantalla de registro

Caso de Prueba 1: Registro Fallido

- En este caso se dejan los inputs vacíos para así soltar una validación, las cuales no permiten la creación del usuario si los campos fallan (ver **Figura 3.38**).

A screenshot of a mobile application's registration screen. The form contains several input fields: 'Nombre:' with the text 'Jorge', 'Email:' with 'alguien@gmail.com', 'Contraseña:' (password), and 'Confirmar Contraseña:' (confirm password). Below these is a 'Selección:' section with two radio buttons, 'Chef' and 'Usuario'. There is a checkbox labeled 'Acepto los Términos y condiciones' which is checked. At the bottom is an orange button labeled 'CREAR'. The background shows a blurred image of a plate of food.

Figura 3.38 – Validación de campos pantalla registro

Caso de Prueba 2: Registro Fallido

- En este caso se ingresa un correo electrónico ya existente, lo cual muestra un mensaje de error (ver **Figura 3.39**).



Figura 3.39 – Mensaje de error por correo electrónico existente

Caso de Prueba 3: Registro Exitoso

- Se siguen los pasos planteados (ver **Figura 3.40**)

Resultado Esperado:

- La aplicación móvil registra con éxito al Chef, teniendo el registro en la base de datos (ver **Figura 3.41**).
- Se redirige al Chef a la página de inicio de sesión.

Criterios de Aceptación:

- El Chef puede iniciar sesión con las credenciales registradas.

Nombre: Jose

Email: Jose@gmail.com

Contraseña: [oculto]

Confirmar Contraseña: [oculto]

Selecciones: Chef Usuario

Acepto los términos y condiciones

CREAR

Figura 3.40 – Registro de Chef

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
jose@gmail.com	[icono de correo]	1 feb 2024	1 feb 2024	1HHoGPct8rfgsv01wgZUIMTZ...
ejemplo@gmail.com	[icono de correo]	31 ene 2024	31 ene 2024	8UMx1z4nxLQnlvFvleJ5xbiOUr...
dustin01ma@gmail.com	[icono de correo]	31 ene 2024	31 ene 2024	iv5wQB14QZzJlgo0gmV9dQx...
samuel01cu@gmail.com	[icono de correo]	31 ene 2024	31 ene 2024	mg3Y3v8vE3d1Mrbxtv904fS...
algo@example.com	[icono de correo]	26 ene 2024	26 ene 2024	mcfV5dzz1K00X4S8fhzL6wfd...

Figura 3.41 – Registro de Chef en Base de Datos

Pruebas de Compatibilidad

Las pruebas de compatibilidad de software son utilizadas para evaluar el rendimiento de la aplicación móvil en diferentes situaciones y en diversos tipos de dispositivos. Esto se debe a que cada usuario utiliza la aplicación en escenarios que varían según la cobertura de red, el sistema operativo y el navegador que emplea [7].

Se han realizado pruebas exhaustivas en varias versiones del sistema operativo Android para asegurar que la aplicación móvil sea accesible y funcione correctamente en diversos dispositivos.

El objetivo principal de las pruebas de compatibilidad es garantizar que la aplicación móvil de Gestión de Recetas funcione de manera consistente y sin problemas en diversas versiones del sistema operativo Android. Estas pruebas buscan confirmar la adaptabilidad y el rendimiento óptimo de la aplicación en una amplia gama de dispositivos Android, desde versiones más antiguas hasta las más recientes.

A continuación, se presenta un ejemplo de las pruebas de Compatibilidad realizadas, las pruebas restantes se encuentran en el

Pruebas de Compatibilidad.

Alcance: Android 10 (ver características del dispositivo en la **Figura 3.42**)



Figura 3.42 - Características de dispositivo Android 10

Criterios de Aceptación:

- Apertura de la aplicación de manera correcta.
- Conexión a la base de datos de manera exitosa.
- Configuración de la interfaz.

Resultados Esperados:

- La apertura de la aplicación se desarrolle de forma exitosa en cualquier dispositivo.
- La conexión hacia la base de datos se encuentre de manera correcta.

Problemas Encontrados:

Se han encontrado algunos problemas con la interfaz en esta versión de la aplicación, teniendo como resultados que con versiones antiguas de Android las vistas sufren un cambio en las medidas o estilos.

Registro de Pruebas:

- **Android 10:** se han presentado algunos problemas en tema de la interfaz, ya que los elementos se organizan de manera diferente a otras versiones de Android (ver **Figura 3.43**).
- No presenta errores en el rendimiento, referido al sistema del celular.
- No presenta errores al momento de realizar lo que es la apertura de la aplicación.
- No presenta error al momento de realizar actividades que lleven a la conexión con la base de datos (Login, Crear receta, etc.).



Figura 3.43 – Error en estilos en Android 10

Una vez realizadas las pruebas de compatibilidad se puede concluir que el error se vuelve más evidente al ejecutar la aplicación móvil en versiones antiguas de Android, lo que ocasiona problemas exclusivamente en la interfaz, como fallos de respuesta en los estilos. En contraste, en versiones más recientes de Android, estos problemas son inexistentes en puntos específicos de la aplicación móvil.

Pruebas de Aceptación:

Las pruebas de aceptación son un tipo de prueba que depende completamente de la retroalimentación proporcionada por los usuarios finales en relación con su experiencia con el producto de software. Los desarrolladores brindan las aplicaciones, la página web o el sistema desarrollado para que los usuarios interactúen con él y emitan un informe sobre su experiencia [8].

Pruebas de Aceptación beta:

Las pruebas beta de aceptación se realizan en ubicaciones específicas, generalmente desde los dispositivos en los que se instalará el software, para que los usuarios interactúen con él sin la presencia o intervención de los desarrolladores [11]. A continuación, se muestra un ejemplo de la prueba de aceptación beta realizada a la aplicación móvil del presente proyecto de titulación, las restantes pruebas se encuentran en el **ANEXO II**.

Información del Proyecto:

Nombre del Proyecto: Aplicación móvil de gestión de recetas

Responsable: Kevin Moreno

Versión del Producto: V 1.0.1

Objetivo de las Pruebas: Realizar la comprobación sobre los requerimientos solicitados

Verificar la realización de:

- Registro, Inicio y cierre de sesión
- Realizar la Creación, edición, eliminar la receta por el usuario tipo "Chef".
- Realizar la vista, búsqueda y sistema de votación realizado por el usuario tipo "Usuario"

Criterios de Aceptación:

- Realizar de manera correcta el registro de la cuenta mediante el correo electrónico, al igual que el inicio de sesión y además verificar el cierre de sesión.
- Lograr el registro de la receta, la edición, y si el chef lo desea llegar a eliminarla.
- Realizar la visualización de las diferentes recetas, al igual que la búsqueda y poder hacer una votación de receta favorita.

Herramientas Utilizadas: Dispositivo Android

Casos de Prueba:

- Caso Prueba #1: Registro, Inicio y cierre de sesión
- Caso Prueba #2: CRUD Recetas (chef)
- Caso Prueba #3: Sistema de recetas

Caso Prueba #1: Registro, Inicio y cierre de sesión

Descripción:

- Realizar de manera correcta el registro de la cuenta mediante el correo electrónico, al igual que el inicio de sesión y cierre de esta.

Pasos Por Seguir:

1. Llenar los campos necesarios para el registro
2. Seleccionar Tipo de cuenta
3. Aceptar términos y condiciones
4. Crear cuenta
5. Colocar el correo y contraseña para poder realizar el inicio de sesión
6. Ingresar a la ventana de Home
7. Realizar el cierre de sesión

Resultado Esperado:

Realizar el inicio de sesión para el usuario y entrar en el home, en el cual dependiendo del tipo del usuario permite diferentes acciones.

Proceso (ver Figura 3.44):

1. Llenar los campos necesarios para el registro.
2. Seleccionar Tipo de cuenta.
3. Aceptar términos y condiciones.

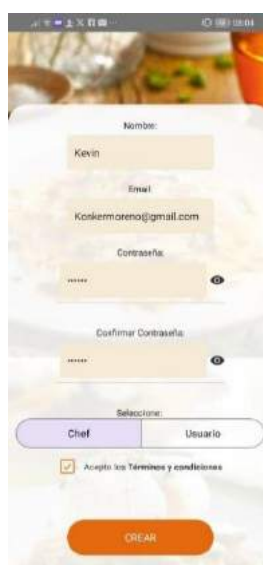


Figura 3.44 - Prueba ventada de registro

4. Crear cuenta. Verificación de cuenta creada en la base de datos (ver **Figura 3.45**).



kevin@gmail.com	✉	14 dic 2023	14 dic 2023	KCZgE3ZpWIMCvseR2YH1z2M...
yuutosakao6@gmail...	✉	7 dic 2023	5 feb 2024	XmIFShCwSR3eeM0Tk6yD7S...
konkormoreno@gmail.c...	✉	1 dic 2023	8 feb 2024	56GTIEToDARDHewKActaYHN...

Figura 3.45 - Registro de Cuenta en la base de datos

5. Colocar el correo y contraseña para poder realizar el inicio de sesión (ver **Figura 3.46**).



INICIO DE SESIÓN

Email

Konkormoreno@gmail.com

Contraseña

.....

¿Olvidaste tu contraseña?

Iniciar Sesión Crear Cuenta

Figura 3.46 - Ventana de inicio de sesión

6. Ingresar a la ventana de Home (ver **Figura 3.47**).



Figura 3.47 – Ventana inicio de home (chef)

Al momento de pulsar el botón de cierre de sesión, se redirecciona de manera automática a la pantalla de iniciar sesión (ver **Figura 3.48**).



Figura 3.48 – Cierre de sesión

Resultado:

El procedimiento de crear e iniciar sesión se desarrolla de manera correcta, logrando la validación de los datos al igual que el poder realizar el paso a la ventana de Home la cual cambia dependiendo del tipo de usuario.

Despliegue:

Uptodown es un sitio web de descarga de software que ha estado operando durante 13 años. Ofrece un amplio catálogo de 2 millones de archivos APK y cuenta con 40,000 reseñas traducidas a 14 idiomas [9].

Uptodown es el sitio elegido para desplegar la aplicación móvil del presente proyecto de titulación. Una vez que se han proporcionado todos los datos necesarios que solicita la página, el despliegue está listo.

El primer paso es proporcionar el icono, nombre de la aplicación y datos del Autor (ver **Figura 3.49**).

App

SELECT ICON FILE

Name *

Operating System *

Food and Drink

Package name

Official website

PEGI (ARC Generic)

My app contains ads

MANAGE APP DISTRIBUTION

Author

Nationality *

Developer's name *

Figura 3.49 - General specs


Se selecciona las diferentes capturas de pantalla que deben ser aprobadas para el despliegue de la aplicación (ver **Figura 3.50**).

Language


Spanish

Screenshots


+ ADD




Pending approval




Pending approval



Pending approval



Pending approval



Pending approval

Figura 3.50 - Screenshots

Luego se envía lo que es una descripción de la aplicación móvil, el cual debe ser validado por los administradores; la aplicación móvil puede estar dispuesta a cambios (ver **Figura 3.51**).

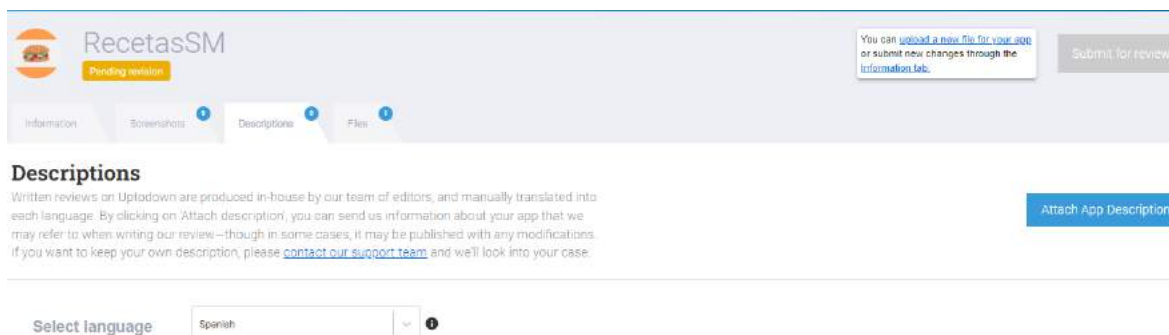


Figura 3.51 - Descriptions

Para finalizar se permite la selección de diferentes opciones para poder así realizar lo que es la revisión de la aplicación, la versión, y el lenguaje (ver **Figura 3.52**).

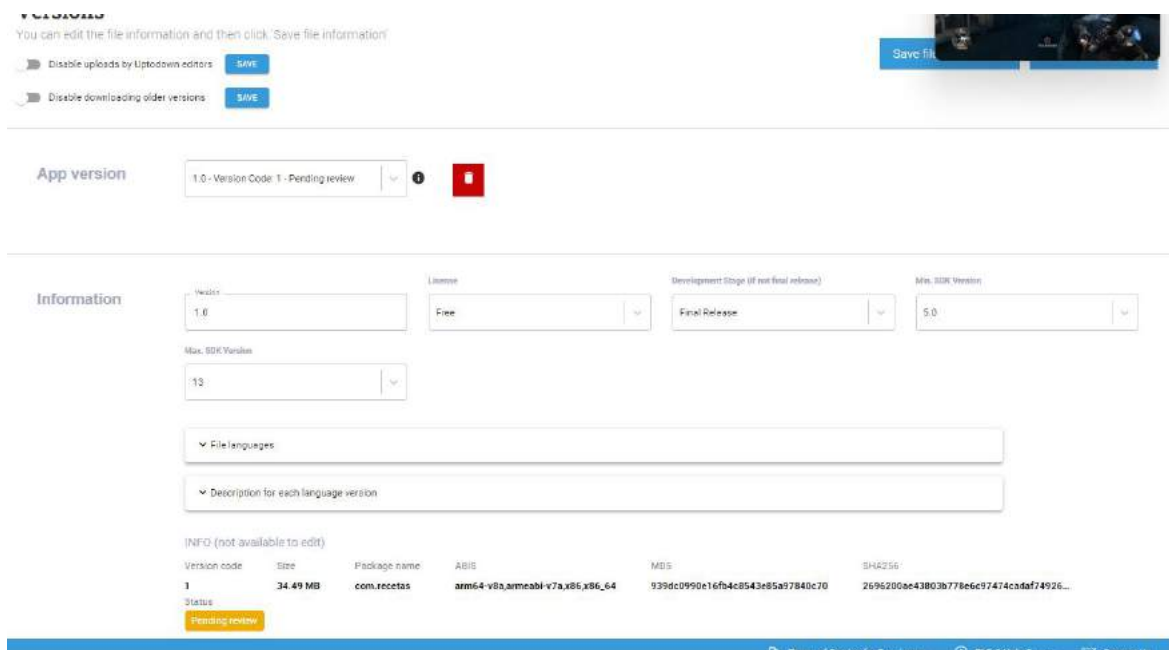


Figura 3.52 - Files

4 CONCLUSIONES

- La selección de React Native como framework de desarrollo ha demostrado su eficiencia al permitir la creación de una aplicación nativa para iOS y Android con un único código base. Esto ha significativamente reducido el tiempo y los esfuerzos necesarios para mantener dos versiones separadas de la aplicación.
- La integración exitosa de Firebase como *backend* ha proporcionado una solución escalable y confiable para el almacenamiento de datos, la autenticación de usuarios y la gestión de archivos multimedia. Esto ha permitido enfocarse en el desarrollo de características en lugar de preocuparse por la infraestructura.
- La aplicación presenta una interfaz de usuario atractiva y funcional, diseñada con React Native y aprovechando componentes nativos. Esto garantiza una experiencia de usuario consistente en ambas plataformas, manteniendo la fluidez y la estética visual.
- La implementación efectiva de la autenticación de usuarios a través de Firebase Authentication ha asegurado un acceso seguro y controlado a la aplicación. Las funciones de registro, inicio de sesión y restablecimiento de contraseña han sido implementadas de manera efectiva.
- La funcionalidad de gestión de recetas permite a los usuarios, especialmente a los chefs, agregar, editar y visualizar recetas de manera intuitiva. La clasificación por tipo, tiempo de preparación y otros criterios ofrece una experiencia completa.
- La inclusión de características sociales, como comentarios, likes y la visualización de recetas más votadas, ha mejorado la participación del usuario y fomentado una comunidad en torno a la aplicación.

5 RECOMENDACIONES

- Antes de iniciar el desarrollo, es crucial realizar una planificación detallada que incluya los requisitos del usuario, la arquitectura de la aplicación, las características clave y los flujos de trabajo. Esto proporcionará una guía clara durante todo el proceso de desarrollo.
- Es fundamental evaluar y seleccionar cuidadosamente las tecnologías a utilizar. En nuestro caso, React Native y Firebase han demostrado ser opciones sólidas. Asegúrate de que las tecnologías seleccionadas se integren bien y cumplan con los requisitos del proyecto.
- Diseña la arquitectura de la aplicación de manera escalable y modutable. Esto facilitará futuras expansiones y mejoras. Utiliza patrones de diseño adecuados y separa las preocupaciones para un mantenimiento más sencillo a largo plazo.
- La seguridad debe ser una prioridad en todas las etapas del desarrollo. Asegúrate de implementar las mejores prácticas de seguridad en la autenticación de usuarios, la gestión de datos y cualquier interacción sensible.
- Realiza pruebas de rendimiento y optimiza el código para garantizar una experiencia fluida para los usuarios. Minimiza la carga de red, optimiza imágenes y utiliza técnicas de carga diferida según sea necesario.
- Aprovecha las características de Firebase para una gestión efectiva de datos, como el uso de Firestore para bases de datos y Firebase Authentication para la autenticación de usuarios. Configura reglas de seguridad adecuadas en Firestore.
- Utiliza sistemas de control de versiones como Git para gestionar el código fuente, e implementa prácticas de despliegue continuo para facilitar actualizaciones y correcciones de errores de manera eficiente.
- Documenta claramente la arquitectura, el código y cualquier configuración importante, y mantén la documentación actualizada a lo largo del desarrollo y las actualizaciones de la aplicación.
- Implementa herramientas de monitoreo y analítica para realizar un seguimiento del rendimiento de la aplicación y comprender el

comportamiento del usuario, utilizando estos datos para tomar decisiones informadas sobre futuras actualizaciones.

- Obtén feedback regular de los usuarios y utilízalo para realizar mejoras continuas en la aplicación. Mantente receptivo a la retroalimentación y ajusta la aplicación según sea necesario.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] G. M. a. W. App, «Firebase,» [En línea]. Available: <https://firebase.google.com/?hl=es>. [Último acceso: 01 02 2024].
- [2] ayudaley, «Base de datos no relacional. ¿Qué es? Características y ejemplos,» Ayudaley, [En línea]. Available: <https://ayudaleyprotecciondatos.es/bases-de-datos/no-relacional/>. [Último acceso: 01 01 2024].
- [3] Atlassian, «Atlassian Desarrollo de software - Scrum,» [En línea]. Available: <https://www.atlassian.com/es/agile/scrum>. [Último acceso: 01 01 2024].
- [4] M. Díez, «React Native - Subir imágenes a Firebase.,» React antive, [En línea]. Available: <https://www.youtube.com/watch?v=tl7l0ypppxw>. [Último acceso: 23 01 2024].
- [5] P. Huet, «Arquitectura de software: Qué es y qué tipos existen,» OpenWebinars.net, [En línea]. Available: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>. [Último acceso: 07 02 2024].
- [6] L. Vespa., «“Medium”.,» [En línea]. Available: <https://medium.com/@vespasoft/patrones-arquitect%C3%B3nicos-en-android-ded39f7a2c10>. [Último acceso: 05 02 2024].
- [7] T. IT, «“¿Deberías aplicar pruebas de compatibilidad a tus apps móviles?” Empresa de pruebas de software,» [En línea]. Available: <https://www.testingit.com.mx/blog/pruebas-compatibilidad-software#:~:text=Las%20pruebas%20de%20compatibilidad%20de%20software%20son%20una%20herramienta%20para,operativo%20y%20el%20navegador%20utilizado..> [Último acceso: 8 01 2024].
- [8] T. IT, «Pruebas de aceptación de software, ¿Cuándo y por qué son necesarias?,» [En línea]. Available: <https://www.testingit.com.mx/blog/pruebas-aceptacion-software>. [Último acceso: 12 01 2024].
- [9] I. Ramírez., «Uptodown, la empresa española que le planta cara a Google Play,» Xataka Android , [En línea]. Available: <https://www.xatakandroid.com/aplicaciones-android/uptodown-la-empresa-espanola-que-le-planta-cara-a-google-play>. [Último acceso: 10 01 2024].
- [10] L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.
- [11] IBM, «¿Qué es una arquitectura de datos?,» IBM, 06 Agosto 2023. [En línea]. Available: <https://www.ibm.com/es-es/topics/data-architecture>. [Último acceso: 08 Agosto 2023].
- [12] Expo, «Ecpo Documentation,» [En línea]. Available: <https://docs.expo.dev/build/setup/>. [Último acceso: 01 febrero 2024].

- [13] npm, «Npm Docs,» [En línea]. Available: <https://docs.npmjs.com>. [Último acceso: 01 01 2023].
- [14] R. Native, «React native,» [En línea]. Available: <https://reactnative.dev>. [Último acceso: 01 02 2024].
- [15] E. E. P. Gomez, « “Cómo construir una aplicación React Native e integrarla con Firebase”,» [En línea]. Available: <https://www.freecodecamp.org/espanol/news/como-construir-una-aplicacion-react-native-e-integrarla-con-firebase/>. [Último acceso: 02 02 2024].
- [16] Miro, «Sprint backlog: qué es, cómo organizarlo, ejemplos y plantillas,» miro, [En línea]. Available: <https://miro.com/es/agile/que-es-sprint-backlog/>. [Último acceso: 11 01 2024].

7 ANEXOS

El respaldo del desarrollo del presente proyecto de titulación se encuentra en los siguientes anexos.

- **ANEXO I.** Certificado de Originalidad
- **ANEXO II.** Manual Técnico
- **ANEXO III.** Manual de Usuario
- **ANEXO IV.** Manual de Instalación

ANEXO I



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 15 de febrero de 2024

De mi consideración:

Yo, IVONNE FERNANDA MALDONADO SOLIZ, en calidad de Director del Trabajo de Integración Curricular titulado DESARROLLO DE UNA APLICACIÓN MÓVIL asociado al DESARROLLO DE UNA APLICACIÓN DE GESTIÓN DE RECETAS DE COCINA elaborado por el estudiante KEVIN JOEL MORENO GARCIA de la carrera en TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones (sin anexos), producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,



Ivonne Maldonado
Docente Ocasional a Tiempo Completo
ESFOT

ANEXO II

Levantamiento de requerimientos

A continuación, la **Tabla 1** muestra los requerimientos para el desarrollo de la aplicación móvil.

Tabla 1 - Levantamiento de Requerimientos

Recopilación de requerimientos		
Tipo de sistema	ID-RR	Enunciado del Ítem
Aplicación móvil	RR001	Como usuario con rol Chef, Usuario necesita: - Registrarse.
	RR002	Como usuario con rol Chef, Usuario necesita: - Iniciar y cerrar sesión.
	RR003	Como usuario con rol Chef necesita - Gestionar Recetas (agregar, editar, mostrar).
	RR004	Como usuario con rol Chef necesita: - Visualizar las recetas. (Recetas, Instrucciones, más votadas).
	RR005	Como usuario con rol Usuario necesita: - Visualizar las recetas. (Recetas, Instrucciones).
	RR006	Como usuario con rol Usuario necesita: - Votar recetas (Likes).
	RR007	Como usuario Chef, Usuario necesita: - La visualización de las recetas más votada en la pestaña de Home
	RR008	Como todos los roles necesita: - Visualizar Recetas por sus diferentes tipos, (Plato Fuerte, Entrada, Postre)
	RR009	Como todos los roles necesita: - Visualizar Recetas de tipo vegetariana
	RR010	Como todos los roles necesita: - Separación de las recetas mediante su tiempo de preparación
	RR011	Como todos los roles necesita: - Búsqueda de Recetas por nombre
	RR012	Como usuario con cualquier rol necesita: - Separar las recetas por tiempo de preparación.

Historias de Usuario

A continuación, de la **Tabla 2** a la **Tabla 11** se muestra las diferentes historias de usuario las cuales representan los diferentes requerimientos y como fueron establecidos para así ser realizados.

Tabla 2 - Iniciar sesión

HISTORIA DE USUARIO	
Identificador (ID): RR002	Usuario: Chef, Usuario
Nombre Historia: Iniciar de sesión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	
Descripción: El usuario requiere el inicio de sesión para poder acceder a los diferentes tipos de características que puede presentar dependiendo del tipo de usuario que selecciono al momento del registro del mismo	
Observación: El usuario tendrá una pantalla la cual permite el ingreso del nombre y su contraseña respectiva, además de tener un <i>feedback</i> si existe algún problema o error al inicio de sesión	

Tabla 3 - Gestionar recetas (Crear, Editar y Eliminar)

HISTORIA DE USUARIO	
Identificador (ID): RR003	Usuario: Chef
Nombre Historia: Gestionar recetas (Crear, Editar y eliminar)	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	
Descripción: El Chef requiere de un sistema de gestión que le permita crear, publicar y eliminar las recetas que puede generar.	
Observación: El Chef solo podrá eliminar y editar las recetas de su propia creación.	

Tabla 4 - Visualizar recetas

HISTORIA DE USUARIO	
Identificador (ID): RR004	Usuario: Chef/Usuario
Nombre Historia: Visualizar recetas	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	

<p>Descripción: El Chef requiere visualizar las recetas, como sus pasos y diferentes características, además de observar las recetas más votadas en su home. El Usuario requiere que pueda visualizar las diferentes recetas que fueron proporcionadas por los diferentes chefs para así poder encontrar alguna que sea de su agrado.</p>
<p>Observación: El Chef se le permite observar las 3 recetas más votadas en la pantalla de home.</p>

Tabla 5 - Calificar las diferentes recetas

HISTORIA DE USUARIO	
Identificador (ID): RR005	Usuario: Usuario
Nombre Historia: Calificar las diferentes recetas	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	
Descripción: Al Usuario se le permitirá realizar un voto por su receta favorita para así poder apoyarla y poder mostrarla a los diferentes usuarios.	
Observación: Las recetas que sean más votadas se mostraran en la pantalla de los usuarios para que así puedan acceder a ellas	

Tabla 6 - Visualizar las recetas más votadas

HISTORIA DE USUARIO	
Identificador (ID): RR006	Usuario: Chef, Usuario
Nombre Historia: Visualizar las recetas más votadas	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	
Descripción: Al Chef y Usuario se le permitirá la visualización de las recetas más votadas, en la ventana de home para así poder acceder a estos de manera mucho más rápida y flexible.	
Observación: Las recetas que sean más votadas se mostraran en la pantalla de los usuarios para que así puedan acceder a ellas	

Tabla 7 - Visualizar las recetas por su diferente tipo

HISTORIA DE USUARIO	
Identificador (ID): RR007	Usuario: Usuario
Nombre Historia: Visualizar las recetas por su diferente tipo.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	

Responsable (es): Kevin Moreno
Descripción: El usuario tiene la capacidad de realizar una búsqueda dependiendo del tipo de plato que desee, (Plato fuerte, Entrada, Postre)
Observación: Las recetas que sean más votadas se mostraran en la pantalla de los usuarios para que así puedan acceder a ellas

Tabla 8 - Visualizar las recetas vegetarianas

HISTORIA DE USUARIO	
Identificador (ID): RR008	Usuario: Usuario
Nombre Historia: Visualizar las recetas vegetarianas	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	
Descripción: Al Usuario se le permitirá realizar una división de las recetas, en las cuales solo se mostrarán las que son vegetarianas	
Observación: Las recetas que son tipo vegetarianas se podrán mostrar si se activa un interruptor para así poder visualizarlos y dividirlos de manera más sencilla	

Tabla 9 - Generar recetas mediante su tiempo de preparación

HISTORIA DE USUARIO	
Identificador (ID): RR009	Usuario: Chef
Nombre Historia: Generar recetas mediante su tiempo de preparación	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	
Descripción: El chef podrá seleccionar si su receta tiene un tiempo de preparación, (+15 min, 15 min, - 15 min) para que así los usuarios tengan una mayor facilidad de búsqueda de una receta que se adapte a su tiempo.	
Observación: Ninguna	

Tabla 10 - Buscar por nombre y separación por tipo vegetariana

HISTORIA DE USUARIO	
Identificador (ID): RR010	Usuario: Usuario
Nombre Historia: Buscar por nombre y separación por tipo vegetariana	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	

Descripción: Al Usuario se le permitirá la búsqueda de las recetas por su nombre además de la separación de su vista si es o no vegetariana.
Observación: Se le permitirá al usuario realizar una búsqueda y separación más sencilla sobre las recetas

Tabla 11 - Separación de la receta por tiempo de preparación

HISTORIA DE USUARIO	
Identificador (ID): RR011	Usuario: Usuario
Nombre Historia: Separación de la receta por tiempo de preparación.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Iteración Asignada: 1	
Responsable (es): Kevin Moreno	
Descripción: Al usuario se le permitirá realizar una búsqueda mucho más exhaustiva mediante el uso del tiempo que requiera la preparación de un tipo de receta.	
Observación: Las recetas requieren de un tiempo de preparación para así poder ser seleccionadas	

Product Backlog

Los requisitos se organizan según las necesidades del *Product Owner* y la complejidad de su implementación (Ver **Tabla 12**).

Tabla 12 – *Product backlog*

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID-RR	Enunciado del Ítem	Iteración	Estado	Prioridad
RR001	Como usuario con rol Chef, Usuario necesita: - Registrarse.	2	Finalizado	Alta
RR002	Como usuario con rol Chef, Usuario necesita: - Iniciar y cerrar sesión.	2	Finalizado	Alta
RR003	Como usuario con rol Chef necesita: - Gestionar Recetas (agregar, editar, mostrar).	2	Finalizado	Alta
RR004	Como usuario con rol Chef necesita: - Visualizar las recetas. (Recetas, Instrucciones, más votadas).	1	Finalizado	Baja
RR005	Como usuario con rol Usuario necesita: - Visualizar las recetas. (Recetas, Instrucciones).	1	Finalizado	Media

RR006	Como usuario con rol Usuario necesita: - Calificar recetas (Likes).	2	Finalizado	Media
RR007	Como usuario Chef, Usuario necesita: - La visualización de las recetas más votada en la pestaña de Home	1	Finalizado	Media
RR008	Como todos los roles necesita: - Visualizar Recetas por sus diferentes tipos, (Plato Fuerte, Entrada, Postre)	1	Finalizado	Media
RR009	Como todos los roles necesita: - Visualizar Recetas de tipo vegetariana	1	Finalizado	Baja
RR010	Como todos los roles necesita: - Separación de las recetas mediante su tiempo de preparación	1	Finalizado	Media
RR011	Como todos los roles necesita: - Búsqueda de Recetas por nombre	1	Finalizado	Media
RR012	Como usuario con cualquier rol necesita: - Separar las recetas por tiempo de preparación.	1	Finalizado	Media

Sprint Backlog

La **Tabla 13** muestra el *Sprint backlog* del presente proyecto de titulación, guía de cada una de las tareas que se han ejecutado.

Tabla 13 – Sprint Backlog

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>					
ID – SB	NOMBRE	MÓDULO	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	MÓDULO DE PREPARACIÓN Y CONFIGURACIÓN	Configuración del ambiente de desarrollo.	Inicio del proyecto (Creación) y ajuste de las herramientas necesarias para realizarlo.	<ul style="list-style-type: none"> • Recopilación y definición de requerimientos. • Preparación del entorno de desarrollo en <i>Visual Studio Code</i>. • Instalación y configuración de paquetes requeridos para la aplicación en React native 	20H

SB001	MÓDULOS PARA CHEF	Autenticación y registro (Chef y Usuario)	Iniciar y cerrar sesión y registro	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para inicio y cierre de sesión. • Registro de cuenta con el rol de chef • Pruebas respectivas 	90H
SB002		Creación, Edición y Eliminación de receta	Ingresar en la gestión de recetas	<ul style="list-style-type: none"> • Diseño e implementación de la ventana de Home para la aplicación de chef • Pruebas respectivas 	

			<p>Gestionar (Crear/Editar/Eliminar) Recetas.</p>	<ul style="list-style-type: none"> • Diseño la pantalla para la creación de las recetas mediante el uso de la base de datos Firebase y configurar la subida de imágenes e información e manera correcta • Poder obtener los datos de manera correcta para así poder realizar las diferentes gestiones (Editar y eliminar las recetas). • Pruebas respectivas 	
SB003	MÓDULOS PARA USUARIO	Visualización de las recetas (Usuario)	Visualizar las recetas existentes	<ul style="list-style-type: none"> • Diseño la pantalla para la visualización de las recetas existentes • Pruebas respectivas 	90H
			Módulo de visualización de recetas (detalle)	<ul style="list-style-type: none"> • Diseño y Creación de un módulo que represente todas las características de la receta seleccionada 	
SB004			Sistema de búsqueda de recetas y clasificación	<ul style="list-style-type: none"> • Diseño y creación de un sistema de búsquedas de recetas por su nombre • Diseño y creación de un sistema de clasificación de recetas por su tipo, tiempo y si es vegetariana 	

SB005		Votos receta favorita y ventana de home	Módulo de votación de la receta favorita.	<ul style="list-style-type: none"> • Diseño y creación de un sistema de votación de recetas para poder mostrar las más populares 	20H
			Visualización de recetas más votadas e ingreso a la ventana de lista recetas	<ul style="list-style-type: none"> • Diseño e implementación de la ventana de Home para la aplicación de usuario • Pruebas respectivas 	
SB006	Pruebas y Despliegue	<ul style="list-style-type: none"> • Prueba de Funcionalidad • Pruebas de compatibilidad. • Prueba de aceptación. • Despliegue de la aplicación móvil en Uptodown. 			10H
Documentación		<ul style="list-style-type: none"> • Trabajo de Integración Curricular. • Anexos. 			10H
TOTAL					240 H

Diseño de Interfaces

A continuación, se representan los diferentes diseños de interfaces utilizadas para el desarrollo de la aplicación móvil, con diferentes prototipos antes de llegar a la versión final.

Inicio de sesión: (ver **Figura 1** y **Figura 2**)



Figura 1 - Prototipo inicio de sesión



Figura 2 - Inicio de Sesión

Registro: (ver **Figura 3** y **Figura 4**)



Figura 3 - Prototipo Registro



Figura 4 - Registro

Home (Chef): (Ver Figura 5 y Figura 6)



Figura 5 - Prototipo Home (Chef)



Figura 6 - Home (Chef)

Home (Usuario): (Ver Figura 7 y Figura 8)




Figura 7 - Prototipo Home (Usuario)



Figura 8 - Home (Usuario)

Formulario creación de receta: (Ver Figura 9 y Figura 10)



Formulario de creación de receta (Figura 9) que muestra un prototipo con los siguientes elementos:

- Campo de texto con el valor "Pollo".
- Campo de texto con el valor "Asado".
- Campo de texto con el valor "Pollito" y un botón "ELIMINAR" a su derecha.
- Botón "AGREGAR PASO" en azul.
- Campo de texto con el valor "Asado" y un botón "ELIMINAR" a su derecha.
- Botón "AGREGAR INGREDIENTE" en azul.
- Campo de texto con el valor "+15 min" y una flecha hacia abajo.
- Campo de texto con el valor "¿Es vegetariana?" y un botón "No" con una flecha hacia abajo.
- Campo de texto con el valor "Tipo de receta:" y un botón "Entrada" con una flecha hacia abajo.
- Imagen de un plato de comida.

Figura 9 - Prototipo Creación de receta



Formulario de creación de receta (Figura 10) que muestra el formulario final con los siguientes elementos:

- Campo de texto con el valor "Título (máx. 50 caracteres)".
- Campo de texto con el valor "Descripción (máx. 250 caracteres)".
- Campo de texto con el valor "Ingrediente 1" y botones de eliminar y agregar.
- Campo de texto con el valor "Paso 1 (máx. 150 caracteres)" y botones de eliminar y agregar.
- Campo de texto con el valor "Tipo de receta:" y un botón "Plato Fuerte" con una flecha hacia abajo.
- Campo de texto con el valor "-15 min" y una flecha hacia abajo.
- Campo de texto con el valor "¿Es vegetariana?" y un botón "No" con una flecha hacia abajo.
- Botón "Seleccionar Imagen" con un ícono de cámara.
- Botón "Subir Receta" con un ícono de plus.

Figura 10 - Creación de receta

Selección de receta (Chef): (Ver Figura 11)



Lista de recetas (Figura 11) que muestra un prototipo con los siguientes elementos:

- Título "Mis Recetas".
- Campo de texto con el valor "Pollo" y botones "Editar" (azul) y "Eliminar" (rojo).
- Campo de texto con el valor "Galletas" y botones "Editar" (azul) y "Eliminar" (rojo).
- Campo de texto con el valor "Encabollado" y botones "Editar" (azul) y "Eliminar" (rojo).

Figura 11 - Lista recetas (Chef)

Edición de receta: (Ver Figura 12 y Figura 13)

Pollo

Asado

Pollo **ELIMINAR**

AGREGAR PASO

Asado **ELIMINAR**

AGREGAR INGREDIENTE

+15 min

¿Es vegetariana?

No

Tipo de receta:

Entrada

Figura 12 - Prototipo Formulario edición recetas

Pollo

Asado

Pollo **ELIMINAR**

AGREGAR PASO

Asado **ELIMINAR**

AGREGAR INGREDIENTE

+15 min

¿Es vegetariana?

No

Tipo de receta:

Entrada

Figura 13 - Formulario edición recetas

Visualización: (Ver Figura 14 y Figura 15)

Recetas Disponibles:

Pollo

Asado

Ingredientes:

1. Asado

Pasos:

1. Pollo

Tiempo: +15 min

¿Es vegetariana?: No

VER DETALLES

Galletas

Figura 14 - Prototipo Visualización de recetas

Buscar recetas por nombre **LIMPIAR**

Recetas Vegetarianas:

Postre

Mostrar solo Postre:

Tiempo

Recetas Disponibles:

Galletas

Galletitas

Ingredientes:

1. Harina

2. Azucar

3. Leche

Pasos:

1. Mezclar

Figura 15 - Visualización de recetas

Pruebas

A continuación, se presenta la evidencia de las pruebas realizadas a la aplicación móvil.

Pruebas de Funcionalidad

Requisito: Registrarse

Descripción: La aplicación móvil debe permitir que un usuario con rol Chef se registre en la plataforma. Esto implica la creación de funcionalidades específicas que permitan la entrada de información necesaria para la creación de una cuenta.

Usuarios Involucrados: Chef

Escenarios de Uso:

- El Chef accede a la página de registro.
- Ingresa la información requerida, como nombre, correo electrónico y contraseña.
- Envía el formulario de registro (ver **Figura 16**).
- Pruebas Asociadas:



The screenshot shows a mobile registration form with the following elements:

- Nombre:** Input field containing "Juan".
- Email:** Input field containing "alguien@gmail.com".
- Contraseña:** Input field containing "Contraseña" with a visibility toggle icon.
- Confirmar Contraseña:** Input field containing "Confirmar Contraseña" with a visibility toggle icon.
- Selección:** A dropdown menu with two options: "Chef" (selected) and "Usuario".
- Acepto los Términos y condiciones
- CREAR** button.

Figura 16 - Ingreso a la pantalla de registro

Caso de Prueba 1: Registro Fallido

En este caso se dejan los inputs vacíos para así soltar una validación la cual no permite la creación del usuario si esto falla (ver **Figura 17**).

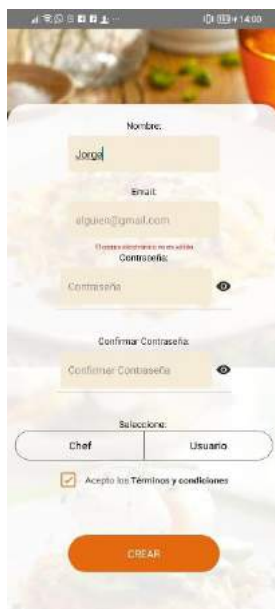


Figura 17 - Validación

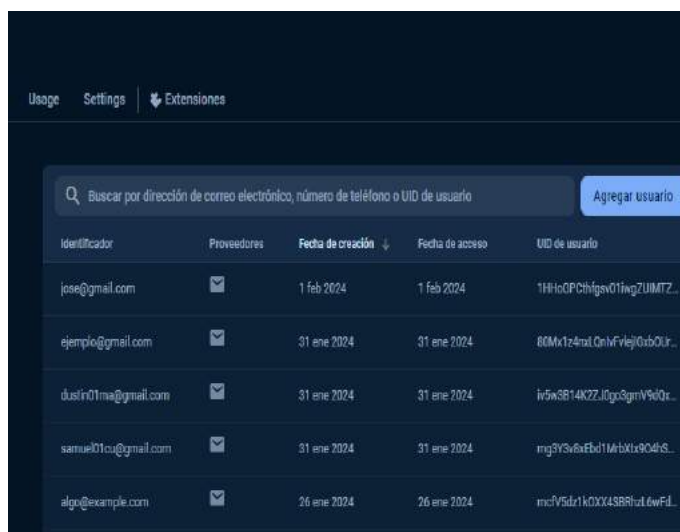
Caso de Prueba 2: Registro Fallido

Ahora se ingresa un correo ya existente por lo cual se muestra un mensaje de error (ver **Figura 18**).



Figura 18 - Validación

Caso de Prueba: Registro Exitoso (Ver Figura 19 y Figura 20).



The screenshot shows a web interface with a dark theme. At the top, there are navigation links for 'Usage', 'Settings', and 'Extensiones'. Below this is a search bar with the placeholder text 'Buscar por dirección de correo electrónico, número de teléfono o UID de usuario' and a blue button labeled 'Agregar usuario'. The main content is a table with the following columns: 'Identificador', 'Proveedores', 'Fecha de creación', 'Fecha de acceso', and 'UID de usuario'. The table contains five rows of user data.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
jose@gmail.com	✉	1 feb 2024	1 feb 2024	1HfIoCPchfgsv01ingZUIMT2...
ejemplo@gmail.com	✉	31 ene 2024	31 ene 2024	8DMx1z4ml.Qn1WVlejGxb0Ur...
dustin01ma@gmail.com	✉	31 ene 2024	31 ene 2024	h5w8B14K2ZJl0go3gmV9d0r...
samuel01ca@gmail.com	✉	31 ene 2024	31 ene 2024	mg3Y3v8xEbd1MxbXt904dS...
algo@example.com	✉	26 ene 2024	26 ene 2024	mcV5dr1k0XX4SBRtL6wFd...

Figura 19 – Registro en la base de datos



The screenshot shows a mobile registration form with the following fields and elements:

- Nombre:** Input field containing 'Jose'.
- Email:** Input field containing 'Jose@gmail.com'.
- Contraseña:** Input field with a toggle for visibility.
- Confirmar Contraseña:** Input field with a toggle for visibility.
- Selecciones:** Two radio buttons, 'Chef' (selected) and 'Usuario'.
- Checkboxes:** A checked checkbox labeled 'Acepto los Términos y condiciones'.
- Botón:** A large orange button labeled 'CREAR'.

Figura 20 - Registro

Pasos:

- Accede a la página de registro.
- Ingresa información válida en todos los campos.
- Envía el formulario de registro.

Resultado Esperado:

- El sistema registra con éxito al Chef.

- Se redirige al Chef a la página de inicio de sesión.
- Criterios de Aceptación:
- El Chef puede iniciar sesión con las credenciales registradas.

Requisito: Iniciar y Cerrar Sesión

Descripción: El sistema debe permitir que un usuario con rol Chef inicie y cierre sesión en la plataforma. Esto implica la creación de funcionalidades específicas para gestionar el proceso de autenticación.

Usuarios Involucrados: Chef

Escenarios de Uso:

- El Chef accede a la página de inicio de sesión.
- Ingresa sus credenciales (correo electrónico y contraseña).
- Envía el formulario de inicio de sesión.

Pruebas Asociadas:

Caso de Prueba: Inicio de Sesión Exitoso (Ver

Figura 21 y Figura 22).



Figura 21 - Pantalla de inicio de Sesión



Figura 22 - Pantalla Home

Pasos:

- Accede a la página de inicio de sesión.
- Ingresa credenciales válidas (correo electrónico y contraseña).
- Envía el formulario de inicio de sesión.

Resultado Esperado:

- El sistema autentica al Chef.
- Se redirige al Chef a la página principal.

Criterios de Aceptación:

- El Chef recibe una confirmación de inicio de sesión exitoso.
- El Chef tiene acceso a las funciones y características exclusivas del rol Chef.

Caso de Prueba: Cierre de Sesión**Pasos:**

- Accede a la opción de cerrar sesión.
- Confirma la acción.

Resultado Esperado:

- El sistema cierra la sesión del Chef.
- Se redirige al Chef a la página de inicio de sesión.

Criterios de Aceptación:

- El Chef recibe una confirmación de cierre de sesión.
- El Chef ya no tiene acceso a las funciones exclusivas del rol Chef.

Requisito: Gestionar Recetas (Agregar, Editar, Mostrar)

Descripción: El sistema debe permitir que un usuario con rol Chef pueda gestionar sus recetas. Esto incluye la capacidad de agregar nuevas recetas, editar las existentes y visualizar la información detallada de una receta.

Usuarios Involucrados:

- Chef

Escenarios de Uso:**Agregar Nueva Receta:**

- El Chef accede a la opción de agregar nueva receta.
- Completa el formulario con los detalles de la receta.
- Envía el formulario para guardar la nueva receta.

Editar Receta Existente:

- El Chef accede a la lista de sus recetas.
- Selecciona una receta para editar.
- Realiza las modificaciones necesarias en el formulario de edición.
- Guarda los cambios.

Visualizar Detalles de Receta:

- El Chef accede a la lista de sus recetas.
- Selecciona una receta para ver detalles.
- Visualiza información detallada de la receta, incluyendo ingredientes, pasos de preparación, etc.

Pruebas Asociadas:**Caso de Prueba:** Agregar Nueva Receta (ver **Figura 23**).

The screenshot shows a mobile application interface for adding a new recipe. The form is displayed on a smartphone screen with a status bar at the top showing signal strength, Wi-Fi, and battery icons, and the time 1:30. The form fields are as follows:

- Título (máx. 50 caracteres):** A text input field.
- Descripción (máx. 250 caracteres):** A text input field.
- Ingrediente 1:** A text input field with a red trash icon and a green plus icon.
- Paso 1 (máx. 150 caracteres):** A text input field with a red trash icon and a green plus icon.
- Tipo de receta:** A dropdown menu with "Plato Fuerte" selected.
- 15 min:** A dropdown menu.
- ¿Es vegetariana?:** A dropdown menu with "No" selected.

At the bottom of the form, there are two orange buttons: "Seleccionar Imagen" (with a camera icon) and "Subir Receta" (with a plus icon).

Figura 23 - Pantalla de creación de recetas

Pasos:

- Accede a la opción de agregar nueva receta.
- Completa el formulario con información válida.
- Envía el formulario para guardar la nueva receta.

Resultado Esperado:

- La nueva receta se guarda correctamente en el sistema.
- Se muestra una confirmación de que la receta fue agregada con éxito.

Criterios de Aceptación:

- Los detalles proporcionados en el formulario se guardan correctamente.
- La receta agregada se refleja en la lista de recetas del Chef.

Caso de Prueba: Editar Receta Existente**Pasos:**

- Accede a la lista de recetas del Chef.
- Selecciona una receta para editar.
- Realiza modificaciones en el formulario de edición.
- Guarda los cambios.

Resultado Esperado:

- Los cambios en la receta se guardan correctamente en el sistema.
- Se muestra una confirmación de que la receta fue editada con éxito.

Criterios de Aceptación:

Las modificaciones en la receta se reflejan correctamente en la lista de recetas del Chef.

Caso de Prueba:

Visualizar Detalles de Receta

Pasos:

- Accede a la lista de recetas del Chef.
- Selecciona una receta para ver detalles.

Resultado Esperado:

Se muestra información detallada de la receta, incluyendo ingredientes, pasos de preparación, etc.

Visualización de la receta creada para Chef: (Ver Figura 24 y Figura 25).



Figura 24 – Visualización de receta



Figura 25 - Visualización de receta

Criterios de Aceptación:

La información detallada de la receta se presenta de manera clara y legible.

Requisito: Visualizar las Recetas (Recetas, Instrucciones, Más Votadas)

Descripción: El sistema debe permitir a un usuario con rol Chef visualizar sus propias recetas, incluyendo detalles como los ingredientes y pasos de preparación. Además, se debe ofrecer la capacidad de visualizar las recetas más votadas por la comunidad.

Usuarios Involucrados: Chef

Visualizar Propias Recetas:

- El Chef accede a la lista de sus recetas.

- Explora las recetas propias para ver detalles como ingredientes y pasos de preparación.

Visualizar Recetas Más Votadas:

- El Chef accede a la sección de recetas más votadas.
- Explora la lista de recetas y visualiza aquellas que han recibido más votos de la comunidad.

Pruebas Asociadas:

Caso de Prueba: Visualizar Propias Recetas (Ver **Figura 26**).



Figura 26 – Visualización de recetas propias

Pasos:

- Accede a la lista de tus propias recetas.
- Selecciona una receta para ver detalles.

Resultado Esperado:

Se muestra información detallada de la receta, incluyendo ingredientes y pasos de preparación.

Criterios de Aceptación:

La información de las recetas propias se presenta de manera clara y legible.

Caso de Prueba: Visualizar Recetas Más Votadas

Pasos:

- Accede a la sección de recetas más votadas.
- Explora la lista de recetas más votadas.

Resultado Esperado:

- Se muestran las recetas más votadas por la comunidad.
- Cada receta incluye detalles como ingredientes y pasos de preparación.

Criterios de Aceptación:

- La lista de recetas más votadas se actualiza de acuerdo con la valoración de la comunidad.
- La información de cada receta es precisa y fácil de entender.

Requisito: Calificar Recetas (*likes*)**Descripción:**

Los usuarios, tanto Chefs como Usuarios, deben tener la capacidad de calificar las recetas expresando *likes*.

Usuarios Involucrados: Chef y Usuario

Escenarios de Uso:**Expresar *likes*:**

- Chefs y Usuarios pueden indicar que les gusta una receta.
- El sistema registra la cantidad total de *likes* recibidos por cada receta.

Pruebas Asociadas:

Caso de Prueba: Expresar *like* a una Receta (Ver **Figura 27**).



Figura 27 – Like para una receta

Pasos:

- Accede a una receta.
- Busca la opción de indicar que te gusta.
- Expresa tu *like*.

Resultado Esperado:

- La cantidad total de *likes* para la receta se actualiza.
- La receta muestra que has expresado tu *like*.

Criterios de Aceptación:

- El sistema registra con precisión la cantidad de *likes* por receta.
- La información sobre los *likes* es visible y actualizada en tiempo real.

Requisito: Visualización de Recetas más Votadas en la Pestaña de Home

Descripción:

Los usuarios con el rol de Chef o Usuario deben poder ver las recetas más votadas directamente desde la pestaña de inicio (Home) para descubrir rápidamente las preferencias populares.

Usuarios Involucrados: Chef y Usuario

Escenarios de Uso: Recetas más Votadas (ver **Figura 28 - Receta más votada**).



Figura 28 - Receta más votada

Acceso a Recetas Populares:

Al iniciar sesión, los usuarios pueden ver una sección en la pestaña de inicio que presenta las recetas más votadas.

Caso de Prueba: Visualizar Recetas más Votadas en la Pestaña de Home

Pasos:

- Inicia sesión en la aplicación.
- Accede a la pestaña de inicio (Home).
- Busca la sección de "Recetas Populares".

Resultado Esperado:

- Se muestra un listado de recetas ordenadas según su cantidad de votos.
- Las recetas más votadas se presentan de manera destacada.
- La información sobre votos es precisa y actualizada.

Criterios de Aceptación:

- La sección de "Recetas Populares" se encuentra fácilmente en la pestaña de inicio.
- Las recetas se ordenan correctamente según la cantidad de votos.
- La información de votos se actualiza en tiempo real.

Requisito: Visualizar Recetas por sus Diferentes Tipos (Plato Fuerte, Entrada, Postre)

Descripción:

Los usuarios con cualquier rol deben tener la capacidad de explorar las recetas según su categoría, permitiéndoles encontrar rápidamente platos que se ajusten a sus preferencias o necesidades culinarias específicas.

Usuarios Involucrados: Chef y Usuario

Exploración por Tipo de Receta:

Los usuarios pueden navegar por la aplicación para encontrar recetas agrupadas por categorías, como Plato Fuerte, Entrada y Postre.

Pruebas Asociadas:

Caso de Prueba: Visualizar Recetas por Categoría (ver **Figura 29**).



Figura 29 - Mostrar solo postre

Pasos:

- Inicia sesión en la aplicación.
- Navega a la sección de "Recetas" o "Explorar".
- Busca la opción para filtrar recetas por categoría o tipo.
- Selecciona la categoría "Plato Fuerte" y revisa las recetas disponibles.

Resultado Esperado:

- Se presenta un menú intuitivo para filtrar recetas por categorías.
- Al seleccionar la categoría "Plato Fuerte", se muestran únicamente recetas que pertenecen a esta categoría.

Criterios de Aceptación:

- Los tipos de recetas son claramente definidos y reconocibles.
- La navegación por categorías es sencilla y fácil de entender.
- Las recetas asociadas a cada categoría son precisas y relevantes.

Requisito: Visualizar Recetas de Tipo Vegetariana

Descripción:

Los usuarios, independientemente de su rol, deben tener la capacidad de identificar y explorar recetas que se ajusten a la categoría vegetariana. Esto facilita a aquellos que siguen una dieta vegetariana encontrar opciones culinarias apropiadas para sus preferencias.

Usuarios Involucrados: Chef y Usuario

Escenarios de Uso:

- Exploración de Recetas Vegetarianas:
- Los usuarios pueden acceder a una sección específica dedicada a recetas vegetarianas.

Pruebas Asociadas:

Caso de Prueba: Visualizar Recetas de Tipo Vegetariana

Pasos:

- Inicia sesión en la aplicación.
- Dirígete a la sección de "Recetas" o "Explorar".
- Busca la opción para filtrar recetas por tipo.
- Selecciona la categoría "Vegetariana" y revisa las recetas disponibles.

Resultado Esperado:

- La aplicación presenta una categoría clara y fácil de encontrar para recetas vegetarianas.

- Al seleccionar la categoría "Vegetariana", se muestran únicamente recetas que cumplen con este criterio.

Criterios de Aceptación:

- La categoría "Vegetariana" está disponible en una ubicación fácilmente accesible.
- Las recetas etiquetadas como vegetarianas son correctas y relevantes.
- Los usuarios pueden cambiar fácilmente entre categorías y explorar diferentes tipos de recetas.

Requisito: Búsqueda de Recetas por Nombre**Descripción:**

La capacidad de buscar recetas por nombre es esencial para que los usuarios encuentren rápidamente sus platillos favoritos o aquellas recetas específicas que desean preparar. Esta funcionalidad facilita la navegación y mejora la experiencia del usuario.

Usuarios Involucrados: Chef y Usuario**Búsqueda Directa por Nombre:**

- Los usuarios pueden ingresar el nombre de una receta en un campo de búsqueda designado.
- La aplicación realiza una búsqueda en tiempo real a medida que se ingresa el nombre.

Resultados de Búsqueda:

Después de ingresar el nombre de la receta, la aplicación presenta resultados coincidentes.

Pruebas Asociadas:

Caso de Prueba: Búsqueda de Recetas por Nombre (ver **Figura 30**).



Figura 30 - Búsqueda de recetas por nombre

Pasos:

- Dirígete a la sección de "Buscar" o "Explorar".
- Localiza el campo de búsqueda y escribe el nombre de una receta existente en la base de datos.
- Observa los resultados que aparecen a medida que escribes.
- Selecciona una receta específica de los resultados.

Resultado Esperado:

- La aplicación ofrece un campo de búsqueda fácilmente accesible.
- A medida que se escribe el nombre de la receta, los resultados se actualizan en tiempo real.
- Se presentan resultados coincidentes y relevantes para el nombre de la receta.

Criterios de Aceptación:

- La función de búsqueda es fácil de encontrar y utilizar.
- Los resultados de la búsqueda son precisos y relevantes.
- Los usuarios pueden hacer clic en una receta específica desde los resultados para obtener más detalles.

Requisito: Búsqueda de Recetas por Tiempo de Preparación

Descripción:

La funcionalidad de búsqueda de recetas por tiempo de preparación es fundamental para que los usuarios encuentren opciones de cocina que se ajusten a sus limitaciones de tiempo. Esto proporciona una manera eficiente de seleccionar recetas que se adapten a diferentes horarios y preferencias.

Usuarios Involucrados: Chef y Usuario

Filtrar por Tiempo de Preparación:

- Los usuarios pueden acceder a la sección de búsqueda o exploración.
- Existe una opción para filtrar las recetas según su tiempo de preparación.

Selección del Rango de Tiempo:

- Los usuarios pueden especificar un rango de tiempo deseado (por ejemplo, recetas que toman menos de 30 minutos).
- La aplicación muestra las recetas que cumplen con el rango seleccionado.

Pruebas Asociadas:

Caso de Prueba: Búsqueda de Recetas por Tiempo de Preparación (ver **Figura 31**).



Figura 31 - Búsqueda por tiempo

Pasos:

- Navega a la sección de "Buscar" o "Explorar".
- Busca la opción de filtrar o clasificar recetas.

- Selecciona la opción para filtrar por tiempo de preparación.
- Especifica un rango de tiempo (por ejemplo, 30 minutos a 1 hora).
- Examina las recetas presentadas después de aplicar el filtro.

Resultado Esperado:

- La opción de filtrar por tiempo de preparación es clara y fácil de encontrar.
- Los usuarios pueden especificar un rango de tiempo deseado.
- Las recetas mostradas cumplen con el rango de tiempo seleccionado.

Criterios de Aceptación:

- La funcionalidad de filtrar por tiempo de preparación está disponible en una ubicación intuitiva.
- Los usuarios pueden ingresar y ajustar fácilmente un rango de tiempo.
- Las recetas que se muestran después de aplicar el filtro cumplen con las especificaciones del tiempo de preparación.

Pruebas de Compatibilidad

A continuación, desde la **Figura 32 - Android 10** a la **Figura 39** se muestran las diferentes pruebas de compatibilidad realizadas a la aplicación móvil.



Figura 32 - Android 10

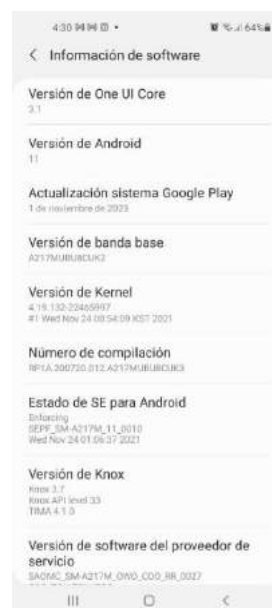


Figura 33 - Android 11

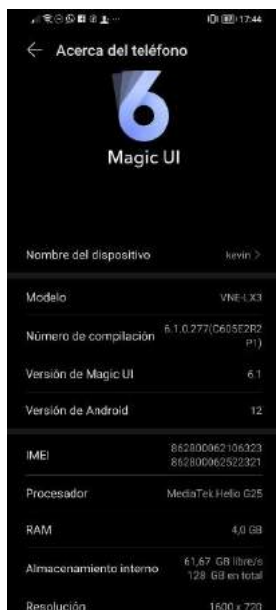


Figura 34 - Android 12

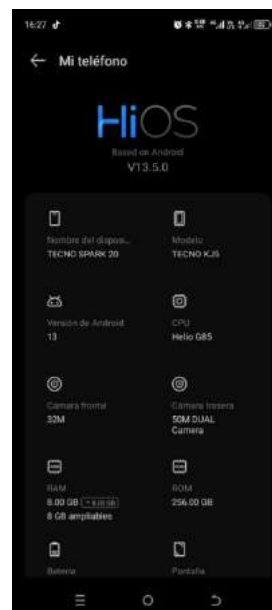


Figura 35 - Android 13

Criterios de Aceptación:

- Apertura de la aplicación de manera correcta
- Conexión a la base de datos de manera exitosa
- Configuración de la interfaz

Resultados Esperados:

- La apertura de la aplicación se desarrolle de forma exitosa en cualquier dispositivo
- La conexión hacia la base de datos se encuentre de manera correcta

Problemas Conocidos:

Se encontraron algunos problemas con la interfaz en esta versión de la aplicación

Registro de Pruebas:

Android 10: se presenta algunos problemas en tema de la interfaz, ya que los elementos se organizaban de manera diferentes a otras versiones de Android (ver Figura 36).

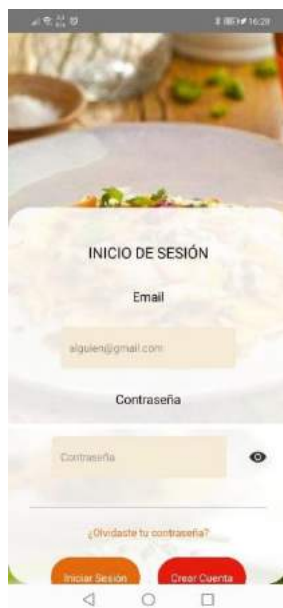


Figura 36 – Prueba Android 10

Android 11: No presenta ningún tipo de error relacionado a la interfaz, ni al momento de la apertura ni al momento de la conexión (ver **Figura 37**).



Figura 37 - Prueba Android 11

Android 12: No presenta ningún tipo de error relacionado a la interfaz, ni al momento de la apertura ni al momento de la conexión (ver **Figura 38**).



Figura 38 - Prueba Android 12

Android 13: No presenta ningún tipo de error relacionado a la interfaz, ni al momento de la apertura ni al momento de la conexión (ver **Figura 39**).



Figura 39 - Prueba Android 13

Pruebas de Aceptación:

A continuación, se presenta el resto de las pruebas de aceptación tipo beta que se han realizado a la aplicación móvil.

Nombre del Caso de Prueba: Caso prueba #2 CRUD Recetas (chef)**Descripción:**

- Lograr el registro de la receta al igual que su propia edición, y si el chef lo desea llegar a eliminarlo.

Pasos Por Seguir:

- Entrar en el formulario de creación de recetas
- Rellenar todos los datos solicitados, al igual que una imagen para la receta
- Crear la receta
- Poder editarla o eliminar la receta

Resultado Esperado:

- Poder realizar lo que es la creación, la edición y eliminación de la receta de manera correcta.

Proceso:

- Entrar en el formulario de creación de recetas (ver **Figura 40**).



The image shows a mobile application interface for creating a recipe. The form is titled 'Pan' and contains several input fields and instructions. The fields are: 'Pan de sal', 'Huevos', 'Harina', and 'Leche'. Each of these fields has a red trash icon to its right and a green plus icon to its left. Below these fields are three instruction boxes: 'Mezclar huevos harina', 'Amasar harina', and 'Hornear'. Each instruction box also has a red trash icon to its right and a green plus icon to its left. At the bottom of the form, there is a dropdown menu labeled 'Tipo de receta:' with 'Entrada' selected, and another dropdown menu showing '*15 min'.

Figura 40 – Formulario creación recetas

- Rellenar todos los datos solicitados, al igual que una imagen para la receta (ver **Figura 41**).



Figura 41 – Formulario creación receta

- Crear la receta (ver **Figura 42**).



Figura 42 – Vista recetas Chef

- Poder editarla o eliminar la receta (ver **Figura 43** y **Figura 44**).



Figura 43 – Formulario edición recetas



Figura 44 – Aceptar eliminar recetas

Resultado:

Se logra crear la receta al igual que su edición y la posibilidad de ser eliminada

Nombre del Caso de Prueba: Caso prueba #3 Sistema de recetas

Descripción:

- Realizar la visualización de las diferentes recetas, al igual que su búsqueda y poder el realizar la votación de su receta favorita.

Pasos Por Seguir:

- Visualizar la lista de recetas
- Buscar las recetas por nombre, tipo, tiempo, es vegetariana.
- Poder observar los detalles de la receta
- Votar por la receta

Resultado Esperado:

Poder realizar al visualización y uso del sistema de gestión de recetas de cocina, realizar la votación de la receta favorita

Proceso:

- Visualizar la lista de recetas (ver **Figura 45**).



Figura 45 – Vista de recetas

- Buscar las recetas por nombre, tipo, tiempo, es vegetariana (ver **Figura 46**).



Figura 46 – Búsqueda por nombre

- Poder observar los detalles de la receta (ver **Figura 47**).



Figura 47 – Vista características de recetas

- Votar por la receta (ver Figura 48).



Figura 48 – Botón de votación

Resultado:

Se ha logrado realizar la visualización completa de las recetas, al igual que usar el sistema de búsqueda para poder observar o encontrar lo que necesita el usuario, poder visualizar las características como el tiempo, pasos, etc. Al igual que poder realizar la votación para que en la parte derecha del home, se muestra las recetas más votadas.

ANEXO III

En el siguiente video se ofrece una explicación detallada sobre las características fundamentales y el funcionamiento de la aplicación, además de resaltar las distinciones entre los perfiles de usuarios.

<https://www.youtube.com/watch?v=kyHlaWrxVx4>

ANEXO IV

A continuación, se describen las credenciales de acceso para la aplicación móvil, así como la ubicación del repositorio en GitHub donde está almacenado el código fuente. En la sección README del repositorio se detallan los pasos para llevar a cabo la instalación local.

Para la aplicación móvil, para acceder a la versión de producción, se debe descargar la aplicación desde:

- Uptodown

Link: <https://recetasm.uptodown.com/android>

Credenciales con rol de Chef:

- Usuario: konkermoreno@gmail.com
- Contraseña: Konker5002

Credenciales con rol de Usuario:

- Usuario: yuutosaka606@gmail.com
- Contraseña: Konker5002

Repositorio del código fuente de la aplicación móvil

El código fuente de este proyecto de integración curricular se encuentra en un repositorio de GitHub, y puede ser visualizado o clonado a través de Git en la siguiente dirección:

https://github.com/KevinxDx3/RECETAS_SM.git