### **ÍNDICE**

[**1. Introducción**](#_izggiko2bqa9) **1**

[**2. Objetivos del Proyecto**](#_x3it1q5cwz6k) **1**

[**3. Funcionalidades Principales**](#_cli7odv5rvnm) **2**

[**3.1. Funcionalidades Generales**](#_sll3x2ejw5d7) **2**

[3.2. Funcionalidades Exclusivas para Usuarios Premium](#_7oudhoopxt6y) 2

[**3.3. Funcionalidades del Administrador**](#_z5rmgpazj9jq) **3**

[**4. Mapeo de Funcionalidades por Roles**](#_gzmt334saqd6) **3**

[**5. Flujo de Funcionalidades**](#_m9hoecb5jp9o) **5**

[**6. Requisitos No Funcionales**](#_i99nrxtef5hc) **7**

[**7. Metodología de Desarrollo**](#_wlqfujw6exdh) **7**

[**8. Arquitectura del Sistema**](#_ivryrln0mcm1) **8**

[**9. Modelo de Negocio**](#_rjlbzz4hwpa1) **9**

[**10. Costos de Implementación**](#_feu5q3jv03or) **10**

[**Resumen de Costos**](#_9h79eoh12cp8) **11**

### 

### **1. Introducción**

**Descripción General**"Chef en Casa" es una aplicación móvil diseñada para ayudar a los usuarios a preparar recetas saludables basadas en los ingredientes disponibles en sus hogares. La aplicación no solo promueve una alimentación consciente y saludable, sino que también busca reducir el desperdicio de alimentos. A través de funcionalidades intuitivas, los usuarios pueden gestionar sus ingredientes, recibir sugerencias personalizadas y acceder a recetas que se ajusten a sus preferencias y necesidades dietéticas.

**Contexto**El desperdicio de alimentos es un problema significativo, y muchas personas enfrentan dificultades al intentar preparar comidas saludables que se adapten a sus necesidades específicas y a los ingredientes disponibles en casa. "Chef en Casa" aborda esta problemática proporcionando herramientas para optimizar el uso de ingredientes y ayudar a los usuarios a adoptar hábitos más sostenibles y saludables.

### **2. Objetivos del Proyecto**

**Objetivo General**Reducir el desperdicio de alimentos en los hogares en un 30% mediante la optimización de la gestión de ingredientes y la recomendación de recetas personalizadas.

**Objetivos Específicos**

* Ofrecer recetas adaptadas a las condiciones de salud de los usuarios (diabetes, hipertensión, pérdida de peso, aumento de masa muscular).
* Fomentar hábitos de alimentación consciente y saludable.
* Optimizar el uso de ingredientes para minimizar el desperdicio alimentario.

**Estrategia del Proyecto**La aplicación recolecta información sobre los ingredientes disponibles y las preferencias dietéticas de cada usuario. Al ingresar sus ingredientes, "Chef en Casa" genera recetas adaptadas, ayudando a los usuarios a utilizar al máximo sus recursos alimenticios y promoviendo hábitos sostenibles.

### **3. Funcionalidades Principales**

### **3.1. Funcionalidades Generales**

* **Personalización de Recetas**: Recetas adaptadas a las necesidades de salud del usuario, como diabetes, hipertensión, pérdida de peso o aumento de masa muscular.
* **Gestión de Ingredientes**: Inventario de cocina donde los usuarios pueden agregar y gestionar sus ingredientes, optimizando el uso de alimentos disponibles en casa y reduciendo el desperdicio.
* **Búsqueda Avanzada de Recetas**: Filtros personalizados para buscar recetas por tipo de plato, tiempo de preparación y preferencias dietéticas.
* **Lista de Compras**: Generación automática de listas de compras basada en los ingredientes faltantes para las recetas seleccionadas. Los usuarios pueden añadir, eliminar y marcar ingredientes como comprados.
* **\*\*Revisar como opción premium por tope de desarrollo\*\*Visualización de Gráficos**: Los usuarios pueden ver gráficos de su consumo de proteínas, carbohidratos y calorías, así como el porcentaje de reducción de desperdicios en el hogar, promoviendo la gestión consciente de los alimentos.
* **Recetas Recomendadas**: La app sugiere recetas en base a los ingredientes que faltan para completar cada receta, de modo que el usuario puede decidir si comprar los ingredientes faltantes para realizar la receta.
* **Recetas en Tendencia**: Visualización de las recetas populares y tendencias en la comunidad de usuarios, para que el usuario pueda inspirarse en lo que otros están cocinando.
* **Panel de Noticias**: Sección dedicada a la visualización de noticias relevantes sobre alimentación y salud, utilizando la integración con la API de NewsAPI.
* **Guardar Recetas**: Los usuarios pueden marcar sus recetas favoritas para guardarlas y acceder a ellas rápidamente en el futuro.
* **Consultas a Nutricionista o Administrador**: Los usuarios pueden realizar consultas en la app para recibir asesoría nutricional o resolver dudas sobre el funcionamiento de la plataforma.
* **Valoración de Recetas**: Los usuarios pueden valorar las recetas, permitiendo que otros usuarios vean la calificación y fomentando la interacción y retroalimentación en la comunidad.

#### **3.2. Funcionalidades Exclusivas para Usuarios Premium**

* **Filtro de Ingredientes por Alergia**: Exclusivo para usuarios premium, permite excluir ingredientes alérgenos.
* **IMC y Dieta Personalizada**: Calculadora de IMC que sugiere dietas adecuadas a cada usuario.
* **Canje de Puntos**: Puntos acumulables por reducción de desperdicio, canjeables por cupones de descuento.

### **3.3. Funcionalidades del Administrador**

* **Gestión de Recetas**: Añadir, monitorear y editar recetas disponibles en la aplicación para que los usuarios tengan acceso a opciones variadas y personalizadas.
* **Gestión de Ingredientes**: Añadir y editar ingredientes en la plataforma, asegurando que la lista de ingredientes se mantenga actualizada y completa para sugerencias de recetas.
* **Monitoreo de Usuarios**: Seguimiento de la actividad de los usuarios en la app, permitiendo observar el uso de las funcionalidades y realizar ajustes según las necesidades de los usuarios.
* **Resolución de Consultas**: Responder las dudas y consultas de los usuarios, ya sea por parte de un nutricionista o del administrador, para brindar asistencia y mejorar la experiencia del usuario en la plataforma.
* **Convenios**: Establecer y gestionar asociaciones con proveedores locales de ingredientes, promoviendo alianzas que enriquezcan la oferta de productos para los usuarios y fomenten el consumo de ingredientes naturales y locales.
* **Gráficos Operativos**: Acceso a gráficos que muestran datos operativos de la app, proporcionando una visión general del funcionamiento y permitiendo al administrador tomar decisiones basadas en datos.
* **Meta de Reducción de Desperdicios**: Monitoreo del avance en la reducción de desperdicios por parte de los usuarios de la app, tanto a nivel individual como general, para evaluar si se están cumpliendo los objetivos del proyecto.
* **Notificaciones**: Creación y envío de notificaciones para informar a los usuarios sobre actualizaciones, eventos y otras novedades relevantes en la aplicación.
* **Gestión de Cupones**: Creación, edición y administración de cupones de descuento, los cuales pueden ser canjeados por los usuarios mediante la acumulación de puntos al reducir el desperdicio de alimentos en sus hogares.

### **4. Mapeo de Funcionalidades por Roles**

**Usuario Básico**

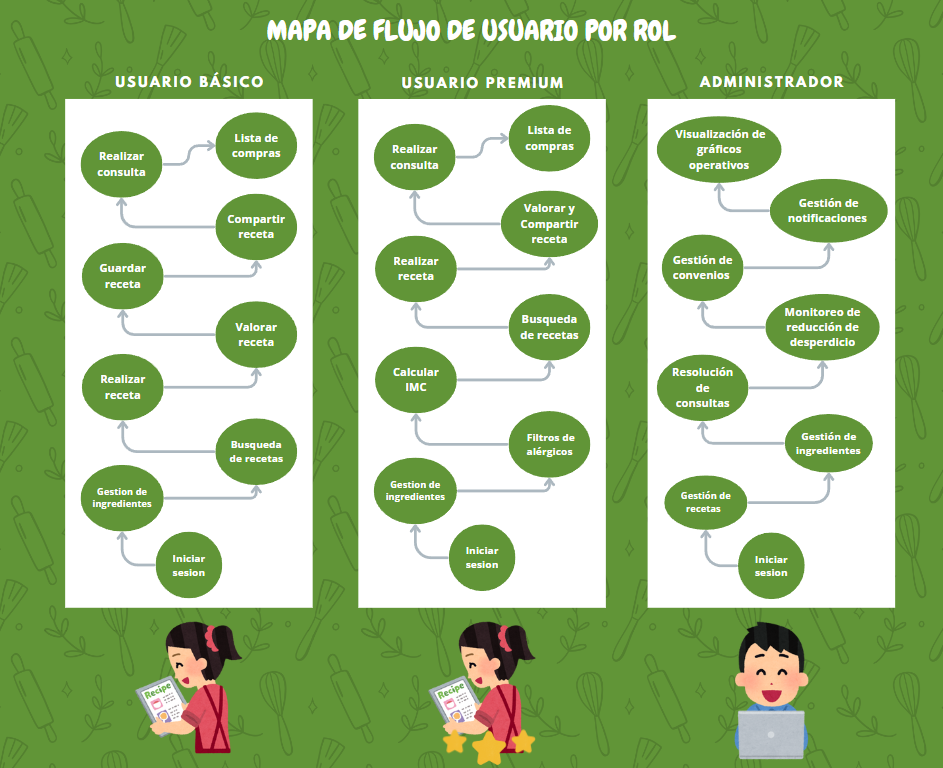
* Gestionar ingredientes, buscar recetas con filtros personalizados, crear listas de compras basadas en ingredientes faltantes, visualizar gráficos sobre consumo de proteínas, carbohidratos, calorías y reducción de desperdicios en el hogar, acceder a recomendaciones de recetas que faltan pocos ingredientes, visualizar recetas populares y en tendencia, acceder a un panel de noticias sobre alimentación y salud, guardar recetas favoritas para acceso rápido, y realizar consultas a nutricionista o administrador para resolver dudas.

**Usuario Premium**

* Además de las funcionalidades básicas, puede filtrar por alérgenos, calcular su IMC y recibir recetas sugeridas en base a su tipo de dieta.

**Administrador**

* Gestor de contenido que incluye la administración de recetas e ingredientes, además de la capacidad para resolver consultas de los usuarios, monitorear la reducción de desperdicio de alimentos tanto a nivel individual como general, acceder a gráficos operativos de la app, establecer y gestionar convenios con proveedores locales, crear y gestionar cupones de descuento, y enviar notificaciones a los usuarios.

****

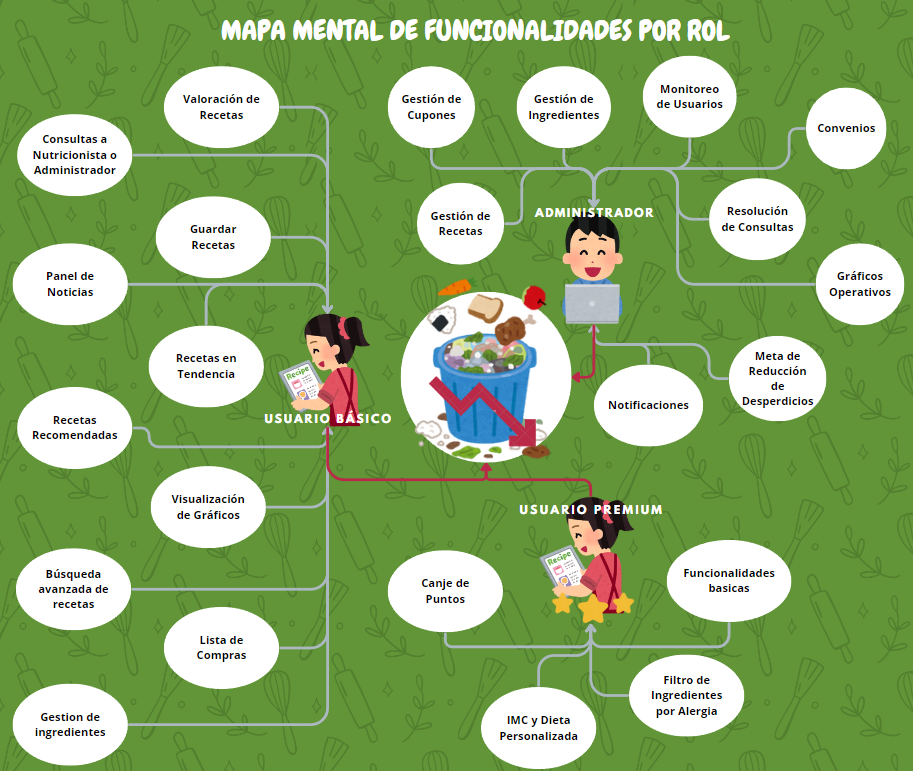
**Diagrama sugerido**: Un **diagrama de flujo** de usuario para cada rol:

* **Usuario Básico**:
  + Inicio de sesión → Gestión de ingredientes → Búsqueda de recetas → Lista de compras → Visualización de gráficos → Consulta al administrador/nutricionista → Recomendaciones y tendencias → Guardado y valoración de recetas.
* **Usuario Premium**:
  + Inicio de sesión → Gestión de ingredientes y filtros de alérgenos → Búsqueda de recetas → Lista de compras → Visualización de gráficos → Consulta al administrador/nutricionista → Recomendaciones y tendencias → Guardado, valoración de recetas y calculadora de IMC.
* **Administrador**:
  + Inicio de sesión → Gestión de recetas e ingredientes → Resolución de consultas → Monitoreo de reducción de desperdicio → Gestión de convenios y notificaciones → Visualización de gráficos operativos.

### **5. Flujo de Funcionalidades**

**Mapa de Flujo**

1. **Inicio de Sesión**: El usuario accede a la aplicación
2. **Gestión de Ingredientes**: El usuario añade y gestiona ingredientes en su inventario de cocina, optimizando el uso de los ingredientes disponibles.
3. **Recomendación de Recetas**: La aplicación sugiere recetas basadas en los ingredientes disponibles, preferencias dietéticas, y necesidades de salud del usuario, ayudando a maximizar el uso de recursos y reducir el desperdicio.
4. **Búsqueda Avanzada de Recetas**: El usuario puede buscar recetas utilizando filtros personalizados (tipo de plato, tiempo de preparación, preferencias dietéticas) para encontrar opciones que se ajusten a sus necesidades.
5. **Lista de Compras**: Generación automática de listas de ingredientes faltantes necesarios para las recetas seleccionadas. Los usuarios pueden añadir, eliminar y marcar ingredientes como comprados.
6. **Visualización de Gráficos**: El usuario accede a gráficos que muestran su consumo de proteínas, carbohidratos y calorías, así como el porcentaje de reducción de desperdicio, para fomentar una gestión alimentaria consciente.
7. **Acumulación de Puntos**: Los usuarios acumulan puntos por reducir el desperdicio de alimentos. Estos puntos son canjeables por cupones de descuento, incentivando el uso eficiente de los ingredientes.
8. **Consulta al Administrador/Nutricionista**: Los usuarios pueden realizar consultas para recibir asesoramiento nutricional o resolver dudas sobre el uso de la app.
9. **Recetas en Tendencia y Recomendadas**: Los usuarios pueden ver recetas populares y en tendencia, así como recomendaciones de recetas donde solo faltan pocos ingredientes.
10. **Valoración de Recetas**: Los usuarios pueden valorar las recetas que prueban, permitiendo que otros usuarios vean sus opiniones y promoviendo la interacción en la comunidad.
11. **Gestión del Administrador:** El administrador tiene acceso a funcionalidades para añadir, editar y monitorear recetas, ingredientes y convenios, gestionar cupones de descuento, enviar notificaciones, resolver dudas y consultas de los usuarios, monitorear la reducción de desperdicios y acceder a gráficos operativos de la app para facilitar una mejor toma de decisiones.



**Explicación sugerida**:

* **Usuario Básico**:
  + **Gestión de ingredientes**: Ayuda a visualizar y controlar el inventario, optimizando el uso de alimentos.
  + **Búsqueda avanzada de recetas y lista de compras**: Facilita la planificación de comidas y evita la compra innecesaria de ingredientes.
  + **Visualización de gráficos**: Permite a los usuarios monitorizar el impacto de su consumo en términos de nutrientes y desperdicio.
* **Usuario Premium**:
  + **Filtro de alérgenos e IMC**: Facilita recetas seguras y adaptadas a necesidades personales, fomentando hábitos específicos y sostenibles.
  + **Canje de puntos**: Incentiva la reducción de desperdicio mediante recompensas.
* **Administrador**:
  + **Monitoreo de usuarios y gráficos operativos**: Permite identificar tendencias en el uso de la app y medir si se están alcanzando los objetivos de reducción de desperdicio.
  + **Resolución de consultas**: Mejora la experiencia del usuario y asegura que reciban soporte adecuado.

### **6. Requisitos No Funcionales**

* **Escalabilidad**: La arquitectura del sistema debe ser capaz de soportar un crecimiento en el número de usuarios sin comprometer el rendimiento, permitiendo agregar funcionalidades y manejar un mayor volumen de datos conforme la base de usuarios crezca.
* **Disponibilidad**: La aplicación debe estar disponible para los usuarios las 24 horas del día, los 7 días de la semana, minimizando los tiempos de inactividad para asegurar una experiencia continua y confiable.
* **Seguridad**: La autenticación de los usuarios debe implementarse de forma segura, utilizando JSON Web Tokens (JWT) para proteger los datos de los usuarios y las interacciones en la aplicación.
* **Interfaz Amigable**: La interfaz debe ser intuitiva y fácil de usar, con un diseño visual y funcional que facilite la navegación y permita a los usuarios encontrar rápidamente las funcionalidades y la información que necesitan.
* **Compatibilidad**: La aplicación debe estar optimizada para dispositivos móviles y ser compatible con diferentes versiones de sistemas operativos, garantizando un rendimiento adecuado en una variedad de dispositivos y plataformas.

### **7. Metodología de Desarrollo**

**Metodología Agile/Scrum**Para el desarrollo de nuestra aplicación y su interfaz web, hemos estado utilizando la metodología ágil Scrum, lo cual nos ha permitido gestionar el proyecto de manera efectiva y alcanzar un alto nivel de avance en poco tiempo. Al aplicar Scrum, hemos trabajado en sprints definidos que nos han permitido realizar entregas incrementales del producto, ajustar continuamente y responder a cambios en las necesidades del proyecto. Este enfoque ha sido fundamental para mantener un flujo de trabajo eficiente y enfocado.

En cada sprint, seguimos un conjunto de reuniones que estructuran nuestro trabajo:

1. **Sprint Planning**: Al inicio de cada sprint, nos reunimos para planificar las tareas a completar. Durante estas reuniones, hemos logrado definir prioridades claras, desglosar tareas del backlog y estimar el esfuerzo necesario, lo que nos ha ayudado a mantener un ritmo de trabajo consistente.
2. **Daily Stand-up**: Estas reuniones diarias han sido clave para la comunicación y sincronización del equipo. Cada miembro comparte sus avances, sus próximos pasos y cualquier obstáculo que haya encontrado. Esta práctica ha sido esencial para resolver problemas rápidamente, minimizar retrasos y asegurar que todos estén alineados.
3. **Sprint Review**: Al finalizar cada sprint, llevamos a cabo una revisión en la que presentamos las funcionalidades desarrolladas, tanto en la app como en la web. Este espacio de revisión ha sido crucial para recibir retroalimentación de los stakeholders, validar los resultados y realizar ajustes, asegurando que el producto esté alineado con las expectativas.
4. **Sprint Retrospective**: Después de la revisión, el equipo realiza una retrospectiva para analizar el desarrollo del sprint. Evaluamos lo que funcionó bien, identificamos áreas de mejora y establecemos ajustes para el próximo sprint. Estas retrospectivas nos han permitido evolucionar continuamente nuestro flujo de trabajo y optimizar la productividad.

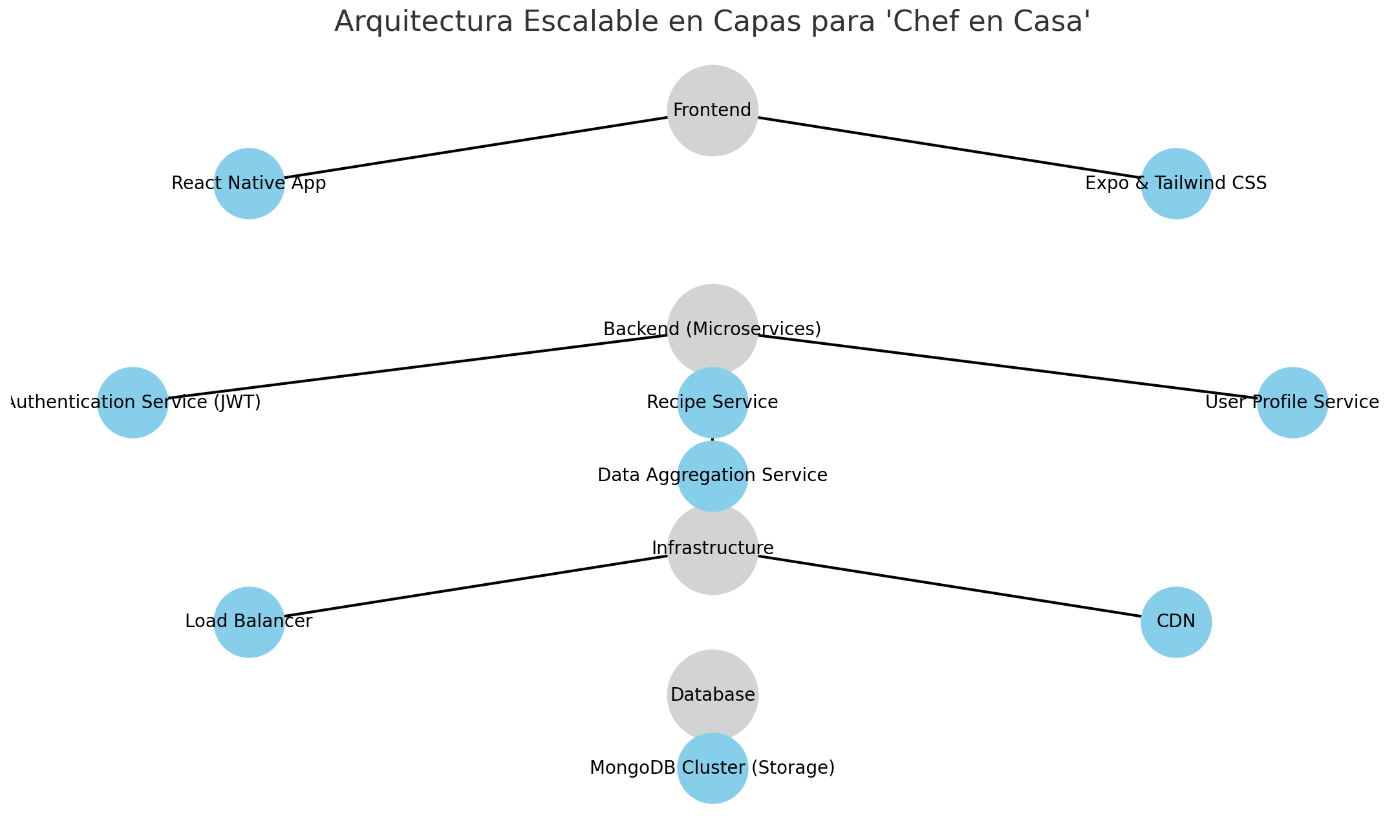
Gracias a esta metodología, hemos logrado mantener un enfoque ágil y adaptativo durante todo el proceso, lo que ha sido esencial para llevar el proyecto a un estado casi finalizado con una alta calidad en el producto. Scrum ha facilitado la organización del equipo y la implementación progresiva de las funcionalidades de la app y el sitio web, asegurando que ambos estén casi listos para el despliegue final.

### **8. Arquitectura del Sistema**

**Arquitectura Actual** El sistema está dividido en tres partes principales: frontend, backend y base de datos. La comunicación se realiza a través de una RESTful API.

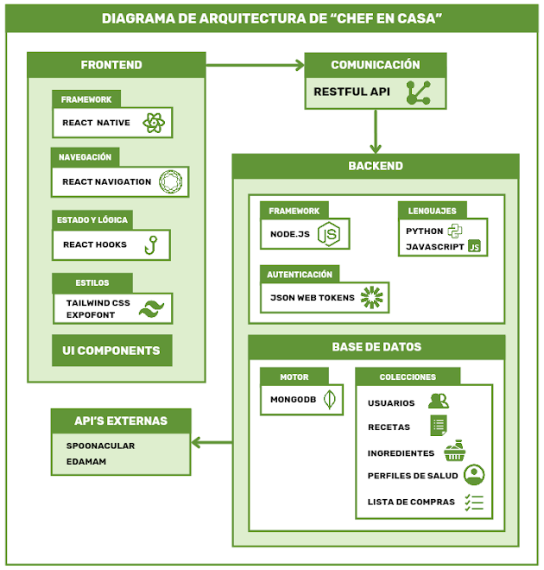
* **Frontend**: Utiliza React Native para la aplicación móvil, junto con Tailwind CSS y ExpoFont para los estilos.
* **Backend**: Construido en Node.js con autenticación por JSON Web Tokens.
* **Base de Datos**: MongoDB como base de datos NoSQL, que permite almacenar y gestionar colecciones de usuarios, recetas, ingredientes y perfiles de salud.

**Propuesta de Arquitectura Escalable**Para soportar el crecimiento en usuarios y la carga de datos, se recomienda una arquitectura basada en microservicios, con balanceo de carga y capacidad de expansión horizontal. MongoDB podría escalarse con una configuración de clúster, y se podría utilizar un CDN para optimizar la entrega de contenido estático.



**Ampliación sugerida**:

* **Estructura de Microservicios**:
  + Permite a cada funcionalidad de la aplicación operar como un servicio independiente, facilitando actualizaciones y escalabilidad.
* **Balanceo de carga y CDN**:
  + Mejora la disponibilidad y reduce el tiempo de respuesta para los usuarios en diferentes ubicaciones.
* **MongoDB en clúster**:
  + Aumenta la capacidad de la base de datos para manejar un volumen creciente de usuarios y datos, asegurando un almacenamiento y recuperación eficientes de grandes volúmenes de información, como ingredientes y recetas.



### **9. Modelo de Negocio**

**Versión Básica y Premium**

* **Versión Básica**: Acceso a la mayoría de las funcionalidades de la aplicación.
* **Versión Premium**: Acceso a funciones exclusivas, como el filtrado por alérgenos y la calculadora de IMC.

**Fuentes de Ingresos**

* **Suscripción Mensual/Anual**: Para la versión Premium.
* **Publicidad**: Opción de incluir publicidad en la versión gratuita.
* **Convenios**: Ingresos a través de alianzas con proveedores de ingredientes, las cuales se promocionan en notificaciones.

### **10. Costos de Implementación**

### **Costos de Desarrollo**

**Desarrollo de Frontend (React Native, Tailwind CSS, ExpoFont):**

* **Desarrollo y diseño de la interfaz:** Creación de pantallas, navegación y ajustes de estilo en React Native con Tailwind CSS y ExpoFont.  
  **Costo ajustado:** **$4,000,000 CLP***(Equipo: 1 desarrollador frontend, 1 diseñador UX/UI durante 1.5 meses)*
* **Integración con APIs externas (NewsAPI, Spoonacular):** Implementación de servicios externos en el frontend.  
  **Costo ajustado:** **$800,000 CLP***(Equipo: 1 desarrollador frontend durante 1 semana)*
* **Pruebas y ajustes finales:** Pruebas funcionales y de usabilidad, corrección de errores.  
  **Costo ajustado:** **$600,000 CLP***(Equipo: 1 desarrollador frontend y 1 tester durante 1 semana)*

**Subtotal ajustado:** **$5,400,000 CLP**

**Desarrollo de Backend (Node.js):**

* **Creación y configuración de servicios backend:** Desarrollo de la lógica de backend utilizando Node.js y configuración de infraestructura.  
  **Costo ajustado:** **$3,500,000 CLP***(Equipo: 2 desarrolladores backend durante 2.5 meses)*
* **Implementación de autenticación segura (JWT):** Configuración de seguridad y gestión de sesiones.  
  **Costo ajustado:** **$600,000 CLP***(Equipo: 1 desarrollador backend durante 2 semanas)*
* **Desarrollo de endpoints:** Creación de endpoints para gestionar recetas, usuarios, ingredientes, etc.  
  **Costo ajustado:** **$1,800,000 CLP***(Equipo: 3 desarrolladores backend durante 1 mes)*

**Subtotal ajustado:** **$5,900,000 CLP**

**Integración de APIs de terceros:**

* **Costos de implementación y prueba de APIs:** Integración de APIs externas necesarias como NewsAPI y Spoonacular en el backend.  
  **Costo ajustado:** **$1,000,000 CLP**

**Subtotal ajustado:** **$1,000,000 CLP**

**Total Costos de Desarrollo Ajustado:** **$12,300,000 CLP**

### **Costos de Infraestructura**

* **Almacenamiento en la Nube:  
  Costo ajustado:** **$1,200,000 CLP al año**
* **Base de Datos MongoDB:  
  Costo ajustado:** **$2,000,000 CLP al año**
* **Servicios de Monitoreo:  
  Costo ajustado:** **$600,000 CLP al año**
* **Balanceo de Carga y CDN:  
  Costo ajustado:** **$800,000 CLP al año**

**Total Costos de Infraestructura Ajustado:** **$4,600,000 CLP al año**

### **Costos de Mantenimiento (Anuales)**

**Mantenimiento del Frontend:**

* **Actualización y soporte de la interfaz:** Ajustes menores en la UI/UX, corrección de errores y mejoras periódicas.  
  **Costo ajustado:** **$1,000,000 CLP***(Equipo: 1 desarrollador frontend durante 1 mes al año)*

**Mantenimiento del Backend:**

* **Actualización y soporte de la lógica de backend:** Mantenimiento de servicios, corrección de errores y mejoras en la infraestructura.  
  **Costo ajustado:** **$1,200,000 CLP***(Equipo: 1 desarrollador backend durante 1.5 meses al año)*
* **Monitoreo y optimización de la base de datos y APIs:** Supervisión continua y ajustes en la base de datos y APIs externas.  
  **Costo ajustado:** **$600,000 CLP***(Equipo: 1 desarrollador backend durante 1 mes al año)*

**Mantenimiento de APIs de terceros:**

* **Costos de implementación de actualizaciones y pruebas:** Realización de actualizaciones y pruebas para garantizar la compatibilidad continua de las APIs externas.  
  **Costo ajustado:** **$500,000 CLP***(Equipo: 1 desarrollador frontend/backend durante 1 mes al año)*

**Total Mantenimiento Anual Ajustado:** **$3,300,000 CLP**

### **Resumen Ajustado de Costos**

* **Costos de Desarrollo:** **$12,300,000 CLP**
* **Costos de Infraestructura:** **$4,600,000 CLP al año**
* **Costos de Mantenimiento:** **$3,300,000 CLP al año**