



MANUAL TÉCNICO DEL SISTEMA

Equipo de desarrollo:

Andrés Felipe Ibarra Villamil
Eddie Santiago Escobar Vallejo
Fernando Villota Erazo
Kevin Alexander Pantoja Males



CONTENIDO

Pág.

PRESENTACIÓN.....	4
RESUMEN.....	5
OBJETIVO.....	6
FINALIDAD DEL MANUAL.....	6
INTRODUCCIÓN.....	7
1. ASPECTOS TÉCNICOS.....	8
1.1. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO.....	8
1.1.1. GitHub.....	8
2. DIAGRAMAS DE MODELAMIENTO.....	9
2.1. DIAGRAMA DE CASOS DE USO.....	9
3. ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA.....	10
3.1. ARQUITECTURA DEL SISTEMA.....	10
3.1.1. Frontend.....	10
3.1.2. Backend.....	10
3.1.3. Base de Datos.....	11
3.2. TECNOLOGIAS Y HERRAMIENTAS.....	11
3.2.1. Lenguajes y Frameworks.....	11
3.3. COMPONENTES DEL SISTEMA.....	12
3.3.1. Módulos del Frontend.....	12
3.3.2. Módulos del Backend.....	12
3.4. FLUJO DE DATOS.....	13
4. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.....	14
4.1. Requisitos Funcionales.....	14
4.2. Requisitos No Funcionales.....	14
BIBLIOGRAFÍA.....	15



LISTA DE FIGURAS

	Pág.
Figura 1 Diagrama de casos de uso	9



PRESENTACIÓN

El siguiente manual se ha desarrollado con la finalidad de dar a conocer la información necesaria de la plataforma web GessenApp, que está diseñada para brindar consejos y recetas personalizadas a usuarios que padecen enfermedades estomacales. GessenApp genera recomendaciones específicas para aliviar malestares gastrointestinales y promover una alimentación saludable y adecuada, basándose en la información proporcionada por cada usuario.



RESUMEN

El manual detalla los aspectos técnicos e informáticos de la plataforma web GessenApp con la finalidad de explicar la estructura del sitio web al usuario final para su correcto uso. La siguiente guía se encuentra dividida en las herramientas que se usaron para la creación de la plataforma web con una breve explicación paso a paso.



Manual técnico del sistema

OBJETIVO

Dar a conocer el uso adecuado de la plataforma web GessenApp en aspectos técnicos de manera descriptiva e ilustrada sobre los componentes y funcionalidades que conforman el buen funcionamiento del sistema de información.

FINALIDAD DEL MANUAL

La finalidad de este manual técnico es explicar detalladamente la estructura del desarrollo técnico de la plataforma web GessenApp.



INTRODUCCIÓN

El manual se realiza con el fin de detallar la plataforma web GessenApp en términos técnicos para que el usuario final pueda tener un buen uso de este mismo.

El documento se encuentra dividido en las siguientes secciones:

- **ASPECTOS TEÓRICOS:** Se darán a conocer conceptos, definiciones y explicaciones de los componentes de la plataforma web desde un punto de vista teórico para mayor entendimiento por parte del lector sobre el funcionamiento del sistema de información y herramientas.
- **DIAGRAMAS DE MODELAMIENTO:** Se compone por diagramas e ilustraciones alusivos al funcionamiento de la plataforma web.
- **ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA:** Corresponde a la instrucción al lector sobre los componentes de la plataforma web desde una perspectiva técnica en los aspectos de almacenamiento de datos, estructura del desarrollo y recomendaciones del uso debido de la plataforma.
- **REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES:** Detalla los requerimientos funcionales y no funcionales de la plataforma web.



1. ASPECTOS TÉCNICOS

La plataforma web GessenApp tiene la finalidad de recomendar recetas de platos saludables a personas que padecen de enfermedades digestivas.

1.1. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

Es ésta sección se procede a explicar las herramientas informáticas empleadas para el desarrollo del aplicativo:

1.1.1. GitHub

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo dentro de un proyecto. Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls. Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original. (Luciano Castillo, 2012)

Para GessenApp, se maneja un repositorio privado el cual se contribuye diferentes commits con ajustes o cambios que se realizan en la plataforma web, haciendo así un trabajo en equipo en el desarrollo del sitio web.

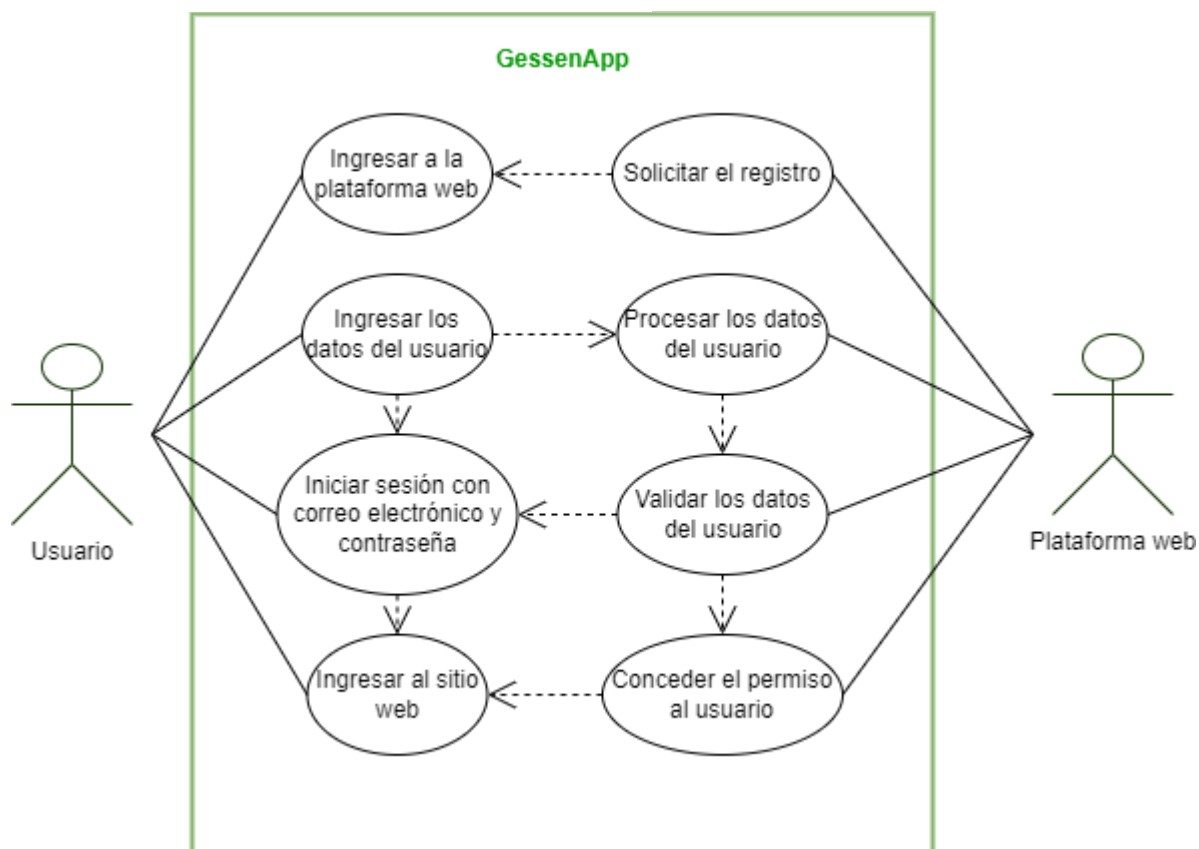


2. DIAGRAMAS DE MODELAMIENTO

2.1. DIAGRAMA DE CASOS DE USO

En el diagrama de casos de uso se detalla el papel a desempeñar en relación con la plataforma web por parte de las personas relacionadas, en este caso los actores son el usuario y la plataforma web.

Figura 1 Diagrama de casos de uso



Fuente: Por el equipo de desarrollo.



3. ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA

En la siguiente sección se procede a realizar una descripción detallada sobre los aspectos técnicos de la plataforma web.

3.1. ARQUITECTURA DEL SISTEMA

El sistema está basado en una arquitectura **cliente-servidor** que utiliza tecnologías web modernas. Está compuesto por tres capas principales: **Frontend**, **Backend** y **Base de Datos**.

3.1.1. Frontend

El frontend es la interfaz de usuario del sitio web, que interactúa directamente con los usuarios finales. Esta capa está construida utilizando tecnologías web estándar y un framework de JavaScript para facilitar el desarrollo de interfaces interactivas.

- **Lenguajes y Frameworks:**
 - **HTML:** Estructura básica de las páginas.
 - **CSS:** Diseño visual, animaciones y estilo de la aplicación.
 - **JavaScript:** Lógica de interacción en el navegador.
 - **Axios:** Para realizar solicitudes HTTP al backend.

3.1.2. Backend

El backend es responsable de procesar las solicitudes del usuario, gestionar la lógica de negocio y acceder a la base de datos. Esta capa también se encarga de generar las recomendaciones personalizadas basadas en los perfiles de salud de los usuarios.

- **Lenguajes y Frameworks:**
 - **Node.js (con Express.js) o Python (con Flask/Django):** Para crear un servidor eficiente que maneje las solicitudes HTTP y la lógica de negocio.
 - **JWT (JSON Web Tokens):** Para la autenticación de usuarios, protegiendo las rutas sensibles.
 - **RESTful API:** Para la comunicación entre el frontend y el backend, facilitando el intercambio de datos en formato JSON.
 - **Celery (si se utiliza Python):** Para la gestión de tareas asíncronas, como el procesamiento de recetas o la generación de listas de compras.



Manual técnico del sistema

3.1.3. Base de Datos

La base de datos almacena los datos de usuarios, recetas, menús, recomendaciones y registros de seguimiento de la dieta.

- **Tipos de base de datos:**

- **PostgreSQL/MySQL:** Bases de datos relacionales para almacenar la información estructurada.
- **MongoDB** (si se opta por un enfoque NoSQL): Base de datos flexible para almacenar perfiles de usuario con diferentes configuraciones de salud.

3.2. TECNOLOGIAS Y HERRAMIENTAS

3.2.1. Lenguajes y Frameworks

- **Frontend:**

- **HTML5, CSS:** Para la estructura y el diseño.
- **JavaScript:** Para las interacciones dinámicas en el cliente.

- **Backend:**

- **Node.js/Express.js** o **Python (Django/Flask):** Para la creación del servidor y las API.
- **JWT:** Para autenticación de usuarios y seguridad.
- **Swagger:** Para la documentación de la API y facilitar la comunicación entre frontend y backend.

- **Bases de Datos:**

- **PostgreSQL:** Base de datos relacional que almacena datos de usuarios, recetas y menús.
- **MongoDB:** Opción para almacenar datos no estructurados o más dinámicos (como preferencias de dieta).

- **Servicios de Nube:**

- **Amazon Web Services (AWS)** o **Google Cloud:** Para el alojamiento y escalabilidad.
- **Firebase:** Para autenticación de usuarios y notificaciones push.

- **Herramientas de Desarrollo**

- **VS Code:** Editor de código recomendado.
- **Postman:** Para pruebas de las API.
- **Git y GitHub:** Para control de versiones y colaboración en el desarrollo del sistema.



Manual técnico del sistema

3.3. COMPONENTES DEL SISTEMA

3.3.1. Módulos del Frontend

➤ **Página Principal:**

- Descripción general de la plataforma.
- Accesos a los módulos de "Recomendaciones", "Planificador de Menú", etc.

➤ **Perfil de Usuario:**

- Permite a los usuarios registrarse, editar su perfil, y gestionar sus condiciones de salud y preferencias alimenticias.

➤ **Sistema de Recomendación de Recetas:**

- Interfaz donde los usuarios pueden ver las recomendaciones personalizadas basadas en su perfil.
- Filtros avanzados (tipo de enfermedad, preferencias alimenticias).

➤ **Planificador de Menú:**

- Herramienta que permite a los usuarios planificar su dieta semanalmente.

➤ **Seguimiento de Dieta:**

- Registra las comidas consumidas y hace un seguimiento de la adherencia al plan alimenticio.

3.3.2. Módulos del Backend

➤ **API de Usuarios:**

- Registro, inicio de sesión, autenticación y gestión de perfiles.

➤ **API de Recetas:**

- Proporciona las recetas recomendadas basadas en el perfil del usuario.
- Permite la modificación y búsqueda de recetas en la base de datos.

➤ **API de Planificación de Menú:**

- Generación de menús semanales personalizados.
- Almacenamiento de las elecciones de recetas en la base de datos.

➤ **API de Notificaciones:**

- Notificaciones por correo electrónico y notificaciones push.



Manual técnico del sistema

3.4. FLUJO DE DATOS

- ✓ El usuario se registra en el frontend con su correo electrónico.
- ✓ El sistema valida la autenticación a través de JWT en el backend.
- ✓ El perfil de usuario se guarda en la base de datos, incluyendo información sobre enfermedades, alergias y preferencias alimenticias.
- ✓ Las recomendaciones de recetas se generan en el backend según los parámetros del perfil y se envían al frontend.
- ✓ El seguimiento de la dieta se almacena en la base de datos.



4. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

4.1. Requisitos Funcionales

- Registro de Usuarios: Permitir que los usuarios se registren en la plataforma web mediante correo electrónico.
- Perfil de Usuario: Permitir a los usuarios crear y gestionar su perfil, incluyendo información como preferencias dietéticas y alergias alimentarias.
- Menús Personalizados: Permitir a los usuarios ver diferentes menús, destacando las opciones saludables y proporcionando información nutricional detallada cuando esté disponible.
- Valoraciones y Reseñas: Permitir a los usuarios dejar valoraciones y reseñas sobre los platos, ayudando a otros usuarios a tomar decisiones informadas.
- Notificaciones en Tiempo Real: Enviar notificaciones en tiempo real a los usuarios recomendando nuevas recetas de su interés e informando acerca de nuevas novedades.
- Soporte al Cliente: Proporcionar un sistema de soporte al cliente integrado en la plataforma web, permitiendo a los usuarios realizar consultas, informar de problemas o solicitar ayuda en tiempo real.

4.2. Requisitos No Funcionales

- Seguridad de Datos: Garantizar la seguridad y privacidad de la información de los usuarios como sus datos personales, mediante la implementación de medidas de cifrado, autenticación y gestión de accesos.
- Escalabilidad: Diseñar la plataforma web de manera que pueda manejar un alto volumen de usuarios, escalando de manera eficiente según sea necesario sin afectar el rendimiento del sistema.
- Disponibilidad: Mantener una alta disponibilidad del servicio, asegurando que la plataforma web esté operativa la mayor parte del tiempo posible y minimizando el tiempo de inactividad mediante la redundancia de servidores, la recuperación ante desastres, etc.
- Rendimiento: Optimizar el rendimiento de la plataforma web para garantizar tiempos de carga rápidos, respuestas rápidas a las solicitudes de los usuarios y una experiencia fluida durante la navegación.
- Compatibilidad: Asegurar que la plataforma web sea compatible con una amplia gama de dispositivos, incluyendo teléfonos inteligentes, tabletas y ordenadores, así como diferentes sistemas operativos y navegadores web.
- Usabilidad: Diseñar la interfaz de usuario de manera intuitiva y fácil de usar, asegurando que los usuarios puedan navegar por la plataforma web, y gestionar su perfil sin dificultades, independientemente de su nivel de experiencia tecnológica.
- Mantenibilidad: Diseñar la plataforma web de manera que sea fácil de mantener y actualizar, utilizando buenas prácticas de desarrollo de software, documentación detallada y herramientas de gestión de versiones.



BIBLIOGRAFÍA

<https://www.who.int/es/news-room/fact-sheets/detail/healthy-diet>

<https://www.mayoclinic.org/es/diseases-conditions/eating-disorders/symptoms-causes/syc-20353603>

<https://www.niddk.nih.gov/health-information/informacion-de-la-salud/enfermedades-digestivas>