**Documentación técnica para el Proyecto**

**“Mediconecta ”**

***Fecha:xx/xx/xxxx***

Indice

[Datos del documento 3](#__RefHeading___Toc677_2802160172)

[1. Introducción 4](#__RefHeading___Toc679_2802160172)

[2. Arquitectura del Sistema 5](#__RefHeading___Toc681_2802160172)

[3. Modelos de Datos y Estructura de la Base de Datos 7](#__RefHeading___Toc683_2802160172)

[4. API y Servicios Backend 9](#__RefHeading___Toc685_2802160172)

[5. Interfaces de Usuario 12](#__RefHeading___Toc687_2802160172)

[6. Pruebas 14](#__RefHeading___Toc689_2802160172)

[7. Despliegue e Infraestructura 16](#__RefHeading___Toc691_2802160172)

[8. Seguridad 17](#__RefHeading___Toc693_2802160172)

[9. Manual del Usuario 18](#__RefHeading___Toc695_2802160172)

[10. Mantenimiento y Actualizaciones 19](#__RefHeading___Toc697_2802160172)

[11. Conclusión 20](#__RefHeading___Toc699_2802160172)

# Datos del documento

Histórico de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Descripción/cambio | autor |
| 0.1 |  | Inicio Proyecto | Product Owner |
|  |  |  |  |
|  |  |  |  |

Información del Proyecto

|  |  |
| --- | --- |
| Organización | Duoc UC. Escuela de Informática y Telecomunicaciones |
| Sección |  |
| Proyecto (Nombre) | Mediconecta |
| Fecha de Inicio |  |
| Fecha de Término |  |
| Patrocinador principal |  |
| Docente | FABIAN ALEJANDRO ALCANTARA GUAJARDO |

Integrantes

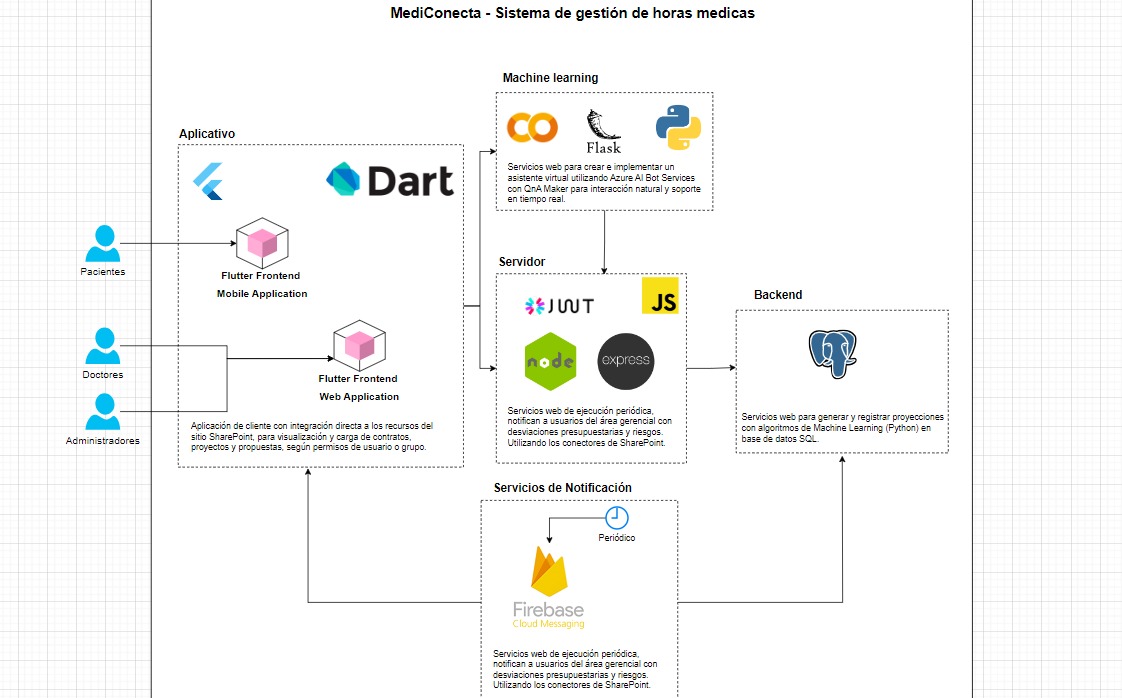
| Rut | Nombre | Correo |
| --- | --- | --- |
| **19916403-1** | **Roberto Valenzuela** |  |
| **20044378-0** | **Efraín Pasarín** |  |
| **20604587-6** | **Sebastián Mora** | **s.mora@duocuc.cl** |

# 1. Introducción

Descripción del Proyecto:  
MediConecta es una plataforma de salud digital diseñada para facilitar el acceso a servicios médicos y el monitoreo de la salud de los adultos mayores. El proyecto combina una aplicación móvil y una plataforma web para permitir a los pacientes gestionar sus citas médicas, recibir notificaciones importantes y monitorear indicadores de salud desde la comodidad de sus hogares. MediConecta está especialmente enfocada en la accesibilidad y facilidad de uso, ofreciendo funciones como recordatorios, registros de salud y asistencia en la gestión de citas, con el objetivo de mejorar la calidad de vida y la autonomía de sus usuarios.  
  
Objetivo de la Documentación:  
Esta documentación técnica está diseñada para proporcionar una visión completa de la estructura y funcionamiento de MediConecta. Su objetivo es detallar los componentes técnicos clave del sistema, explicar los procesos de desarrollo, instalación y mantenimiento, y establecer directrices para asegurar la calidad y el rendimiento del sistema en el tiempo. Este documento servirá como referencia para desarrolladores, administradores de sistemas y otros stakeholders involucrados, facilitando el desarrollo, actualización y escalabilidad de MediConecta.

# 2. Arquitectura del Sistema

Descripción General:  
La arquitectura de MediConecta se basa en una estructura modular compuesta por tres capas principales: Frontend, Backend y Base de Datos. Esta arquitectura sigue un modelo monolítico con integración a través de API REST, que permite una comunicación eficiente y segura entre los diferentes componentes. La estructura está diseñada para soportar la interacción entre pacientes, doctores y administradores, facilitando el manejo de datos y proporcionando una interfaz de usuario accesible y optimizada para dispositivos móviles y web. Las tecnologías utilizadas en cada componente están orientadas a maximizar la escalabilidad, seguridad y rendimiento del sistema.  
  
Diagrama de Arquitectura:

  
  
Componentes Clave:  
  
Frontend:  
  
Aplicación Móvil: Desarrollada con Flutter, esta aplicación está optimizada para dispositivos móviles y ofrece una interfaz accesible para adultos mayores, permitiendo la gestión de citas, recepción de notificaciones y registro de indicadores de salud.  
Plataforma Web: Construida con React, la plataforma web está destinada principalmente para doctores y administradores, quienes pueden visualizar y gestionar la información de pacientes a través de dashboards y funcionalidades específicas.  
Backend:  
  
Desarrollado con Node.js y Express, el backend de MediConecta actúa como el núcleo de la lógica de negocio. Gestiona las interacciones entre el Frontend y la Base de Datos, proporciona autenticación y autorización para diferentes roles (Administrador, Doctor, Paciente), y permite el manejo seguro de la información de los usuarios.  
Base de Datos:  
  
MediConecta utiliza PostgreSQL como sistema de gestión de bases de datos relacional, lo que permite el almacenamiento seguro y escalable de datos críticos, tales como información de usuarios, citas, y registros de salud. La estructura de la base de datos incluye relaciones clave entre entidades, como pacientes, doctores y citas, asegurando que el sistema pueda manejar datos complejos de manera eficiente.  
Integración de Componentes:  
La interacción entre los componentes de MediConecta se realiza principalmente a través de una API REST, que permite la comunicación entre el Frontend y el Backend. La API está diseñada para realizar operaciones de autenticación, autorización y manejo de datos, facilitando un flujo de información seguro y controlado. La autenticación se realiza mediante JWT (JSON Web Tokens), garantizando un control de acceso adecuado según el rol de cada usuario, mientras que la base de datos se conecta directamente con el Backend para manejar consultas y almacenamiento de datos de manera eficaz.

# 3. Modelos de Datos y Estructura de la Base de Datos

Diagrama ER (Entidad-Relación):  
El diagrama ER muestra las relaciones entre las entidades principales del sistema, lo cual incluye a Usuarios, Doctores, Pacientes, Citas e Indicadores de Salud. Este diagrama proporciona una visión clara de cómo están conectados los elementos clave y facilita la comprensión de las interacciones y dependencias entre los datos.  
  
(Incluir un diagrama visual que muestre las entidades principales y sus relaciones, como “uno a muchos” entre doctores y pacientes, y “muchos a muchos” entre pacientes y citas).  
  
Descripción de las Tablas:  
A continuación, se detallan las tablas principales de la base de datos junto con sus campos y tipos de datos:  
  
Usuarios  
  
id\_usuario: INT (PK) – Identificador único de usuario.  
nombre: VARCHAR(100) – Nombre del usuario.  
email: VARCHAR(150) – Correo electrónico del usuario.  
contraseña: VARCHAR(255) – Contraseña cifrada del usuario.  
rol: ENUM(‘Administrador’, ‘Doctor’, ‘Paciente’) – Define el rol del usuario en el sistema.  
Doctores  
  
id\_doctor: INT (PK) – Identificador único del doctor.  
id\_usuario: INT (FK) – Referencia al ID en la tabla Usuarios.  
especialidad: VARCHAR(100) – Especialidad médica del doctor.  
horario\_disponible: JSON – JSON que especifica horarios de disponibilidad.  
Pacientes  
  
id\_paciente: INT (PK) – Identificador único del paciente.  
id\_usuario: INT (FK) – Referencia al ID en la tabla Usuarios.  
fecha\_nacimiento: DATE – Fecha de nacimiento del paciente.  
informacion\_contacto: JSON – JSON con detalles de contacto adicionales.  
Citas  
  
id\_cita: INT (PK) – Identificador único de la cita.  
id\_paciente: INT (FK) – Referencia al paciente que agenda la cita.  
id\_doctor: INT (FK) – Referencia al doctor con quien se agenda la cita.  
fecha\_hora: TIMESTAMP – Fecha y hora de la cita.  
estado: ENUM(‘Pendiente’, ‘Confirmada’, ‘Cancelada’) – Estado actual de la cita.  
Indicadores\_Salud  
  
id\_indicador: INT (PK) – Identificador único del registro de salud.  
id\_paciente: INT (FK) – Referencia al paciente que realiza el registro.  
tipo\_indicador: VARCHAR(50) – Tipo de indicador (ej. presión arterial, glucosa).  
valor: FLOAT – Valor del indicador de salud.  
fecha\_registro: TIMESTAMP – Fecha y hora del registro.  
Relaciones entre Entidades:  
Las relaciones entre las entidades son fundamentales para el correcto funcionamiento del sistema y se definen de la siguiente manera:  
  
Usuarios - Doctores / Pacientes: Relación de "uno a uno". Cada usuario tiene un rol único (administrador, doctor o paciente) que se especifica en la tabla de Usuarios.  
Doctores - Pacientes: Relación de "muchos a muchos" a través de la tabla de Citas, ya que cada doctor puede atender a varios pacientes y cada paciente puede tener citas con diferentes doctores.  
Pacientes - Indicadores de Salud: Relación de "uno a muchos", en la cual cada paciente puede tener múltiples registros de indicadores de salud a lo largo del tiempo.  
Esta estructura asegura la integridad y coherencia de los datos, permitiendo que MediConecta maneje eficientemente la información relevante para la gestión de citas y el monitoreo de la salud de los pacientes.

# 4. API y Servicios Backend

Descripción de la API:  
La API de MediConecta está diseñada como una API RESTful, lo que permite una comunicación estructurada y estandarizada entre el frontend y el backend. La API facilita la gestión de recursos, como usuarios, citas y registros de salud, y es accesible a través de una serie de métodos HTTP (GET, POST, PUT, DELETE). Esta estructura RESTful asegura que las operaciones en el sistema sean seguras, escalables y fáciles de mantener, con un enfoque en la reutilización de componentes y la implementación de buenas prácticas de desarrollo.  
  
Endpoints de la API:  
A continuación se presenta una lista de los endpoints principales de la API, detallando las rutas, métodos, funcionalidad, parámetros y respuestas esperadas:  
  
Usuarios  
  
Ruta: /api/usuarios  
  
Método: POST  
Descripción: Registro de un nuevo usuario en el sistema.  
Parámetros:  
nombre (String, requerido): Nombre del usuario.  
email (String, requerido): Correo electrónico.  
contraseña (String, requerido): Contraseña del usuario.  
rol (String, requerido): Rol del usuario (Administrador, Doctor, Paciente).  
Respuestas:  
Éxito (201): { "mensaje": "Usuario registrado con éxito", "id\_usuario": "123" }  
Error (400): { "error": "El correo electrónico ya está en uso" }  
Ruta: /api/usuarios/login  
  
Método: POST  
Descripción: Autenticación del usuario en el sistema.  
Parámetros:  
email (String, requerido): Correo electrónico.  
contraseña (String, requerido): Contraseña.  
Respuestas:  
Éxito (200): { "token": "jwt\_token", "rol": "Doctor" }  
Error (401): { "error": "Credenciales inválidas" }  
Citas  
  
Ruta: /api/citas  
  
Método: POST  
Descripción: Creación de una nueva cita entre un paciente y un doctor.  
Parámetros:  
id\_paciente (INT, requerido): ID del paciente.  
id\_doctor (INT, requerido): ID del doctor.  
fecha\_hora (TIMESTAMP, requerido): Fecha y hora de la cita.  
Respuestas:  
Éxito (201): { "mensaje": "Cita creada exitosamente", "id\_cita": "456" }  
Error (400): { "error": "El doctor no está disponible en la fecha y hora seleccionada" }  
Ruta: /api/citas/{id\_cita}  
  
Método: GET  
Descripción: Consulta de detalles de una cita específica.  
Parámetros:  
id\_cita (INT, requerido): ID de la cita.  
Respuestas:  
Éxito (200): { "id\_cita": 456, "fecha\_hora": "2024-08-12T10:00:00Z", "estado": "Confirmada" }  
Error (404): { "error": "Cita no encontrada" }  
Ruta: /api/citas/{id\_cita}  
  
Método: PUT  
Descripción: Actualización del estado de una cita.  
Parámetros:  
estado (String, requerido): Nuevo estado de la cita (Pendiente, Confirmada, Cancelada).  
Respuestas:  
Éxito (200): { "mensaje": "Estado de la cita actualizado" }  
Error (400): { "error": "Estado no válido" }  
Indicadores de Salud  
  
Ruta: /api/indicadores  
  
Método: POST  
Descripción: Registro de un nuevo indicador de salud para un paciente.  
Parámetros:  
id\_paciente (INT, requerido): ID del paciente.  
tipo\_indicador (String, requerido): Tipo de indicador (ej. presión arterial).  
valor (Float, requerido): Valor del indicador.  
Respuestas:  
Éxito (201): { "mensaje": "Indicador registrado con éxito", "id\_indicador": "789" }  
Error (400): { "error": "Datos del indicador incompletos" }  
Ruta: /api/indicadores/{id\_paciente}  
  
Método: GET  
Descripción: Consulta de los indicadores de salud de un paciente específico.  
Parámetros: Ninguno.  
Respuestas:  
Éxito (200): [ { "id\_indicador": 789, "tipo\_indicador": "presión arterial", "valor": 120, "fecha\_registro": "2024-08-12T10:00:00Z" }, ... ]  
Error (404): { "error": "Indicadores no encontrados" }  
Autenticación y Seguridad:  
La autenticación en MediConecta se implementa mediante JWT (JSON Web Tokens), donde los usuarios reciben un token después de autenticarse en el sistema. Este token se incluye en los encabezados de las solicitudes para validar el acceso del usuario. La autorización se gestiona según el rol del usuario:  
  
Administrador: Acceso completo a la administración de usuarios, citas y configuraciones del sistema.  
Doctor: Acceso a las citas asignadas y la información de los pacientes.  
Paciente: Acceso a su perfil, citas y registros de salud.  
Estas medidas de autenticación y autorización aseguran que solo los usuarios autorizados puedan acceder y modificar la información relevante, protegiendo así la privacidad y seguridad de los datos en MediConecta.

# 5. Interfaces de Usuario

Descripción de la UI/UX:  
La interfaz de usuario de MediConecta está diseñada bajo principios de accesibilidad y usabilidad, con un enfoque en la experiencia de adultos mayores. Los elementos de la UI (User Interface) están estructurados para facilitar la navegación, utilizando botones grandes, fuentes legibles y un esquema de color que mejora la visibilidad. Además, se implementan opciones de interacción simplificadas y un flujo lógico para reducir las barreras tecnológicas. La experiencia de usuario (UX) se optimiza mediante un diseño intuitivo, minimizando la cantidad de pasos para realizar las tareas principales, como la reserva de citas o el monitoreo de salud.  
  
Mockups y Wireframes:  
Los siguientes mockups y wireframes muestran los prototipos de las pantallas principales de MediConecta:  
  
Inicio de Sesión: Pantalla de autenticación donde los usuarios ingresan su correo electrónico y contraseña. Esta interfaz incluye una opción de recuperación de contraseña para usuarios que lo necesiten.  
  
Panel Principal del Paciente: Pantalla donde el paciente puede ver su próxima cita, acceder a su historial médico, y visualizar sus indicadores de salud.  
  
Gestión de Citas: Interfaz para que el paciente pueda reservar, ver o cancelar sus citas. Incluye un calendario que muestra las fechas disponibles para cada doctor y la opción de confirmar la cita seleccionada.  
  
Dashboard de Doctores: Pantalla dedicada a los doctores, donde pueden revisar las citas programadas, ver información de pacientes y acceder a gráficos que muestran el historial de indicadores de salud de cada paciente.  
  
Perfil del Usuario: Sección donde los usuarios (pacientes y doctores) pueden actualizar su información de contacto y ajustes personales.  
  
(Incluir imágenes o enlaces a los prototipos de estas pantallas si están disponibles).  
  
Flujo de Usuario:  
A continuación se describe el recorrido típico de un usuario en MediConecta:  
  
Creación de Cuenta: El usuario accede a la plataforma y, en caso de ser un nuevo usuario, se registra proporcionando su información personal básica. El administrador confirma y asigna roles a los usuarios que así lo requieran.  
  
Inicio de Sesión: El usuario inicia sesión ingresando sus credenciales. Dependiendo del rol (Administrador, Doctor, Paciente), la plataforma redirige a la interfaz correspondiente.  
  
Reserva de Cita (Paciente):  
  
Desde el panel principal, el paciente accede a la sección de Gestión de Citas.  
Elige un doctor y selecciona una fecha y hora disponibles.  
Confirma la cita, que se guarda en el sistema, y el paciente recibe una notificación de confirmación.  
Consulta de Resultados e Indicadores de Salud (Paciente):  
  
El paciente puede ver su historial de citas y sus indicadores de salud en el panel principal.  
Los indicadores se muestran en un formato gráfico para facilitar la comprensión del estado de salud.  
Revisión de Citas y Pacientes (Doctor):  
  
El doctor accede al Dashboard y consulta las citas programadas.  
Puede revisar el historial de indicadores de salud de cada paciente, lo que le permite realizar evaluaciones informadas durante las consultas.  
Actualización de Información:  
  
Los usuarios pueden acceder a la sección de Perfil para actualizar su información de contacto o realizar ajustes en su configuración personal.  
Este flujo de usuario asegura que cada tipo de usuario (Paciente, Doctor, Administrador) tenga acceso a una interfaz y funcionalidades específicas, permitiendo una experiencia fluida y personalizada en MediConecta.

# 6. Pruebas

Pruebas Unitarias:  
Las pruebas unitarias se implementan para verificar la funcionalidad de cada módulo de manera independiente, asegurando que cada componente básico del sistema funcione como se espera. Estas pruebas se realizan en el Backend y en el Frontend:  
  
Backend: Las pruebas unitarias cubren los módulos de autenticación, gestión de usuarios, manejo de citas y almacenamiento de indicadores de salud. Cada endpoint de la API se prueba de manera aislada para verificar la entrada y salida de datos, asegurando que el backend responda correctamente a diferentes escenarios y maneje adecuadamente los errores.  
  
Frontend: Las pruebas unitarias en el Frontend se enfocan en componentes clave de la interfaz, como formularios de ingreso, botones de navegación y componentes visuales. Estas pruebas aseguran que los elementos de la UI se muestren y funcionen correctamente, y que las validaciones en los campos de entrada respondan a los errores de manera adecuada.  
  
Pruebas de Integración:  
Las pruebas de integración se realizan para validar la interacción entre diferentes módulos y asegurar que funcionen en conjunto sin problemas. Estas pruebas se centran en verificar:  
  
Comunicación entre Frontend y Backend: Se prueba el correcto flujo de datos entre la interfaz de usuario y la API del backend, asegurando que las operaciones como creación de usuarios, reserva de citas y actualización de indicadores se ejecuten correctamente.  
  
Integración con la Base de Datos: Las pruebas de integración verifican que el backend pueda realizar operaciones de creación, lectura, actualización y eliminación (CRUD) en la base de datos sin errores. Esto garantiza que las operaciones de almacenamiento y recuperación de datos se realicen de forma eficiente y segura.  
  
Pruebas de Regresión y Rendimiento:  
Las pruebas de regresión se implementan para asegurar que nuevas funcionalidades o cambios en el sistema no afecten negativamente a las funcionalidades existentes. Estas pruebas se ejecutan después de cada iteración de desarrollo y antes de cada despliegue:  
  
Pruebas de Regresión: Consisten en repetir las pruebas unitarias e integradas en cada actualización para garantizar la estabilidad del sistema. Cualquier fallo detectado es registrado y corregido antes de proceder con el despliegue.  
  
Pruebas de Rendimiento: Estas pruebas miden la capacidad de respuesta y velocidad del sistema bajo diversas condiciones de carga, asegurando que MediConecta pueda manejar un volumen alto de usuarios simultáneamente sin pérdida de rendimiento. Se monitorean métricas como tiempo de carga de la página, tiempo de respuesta del servidor y uso de recursos del sistema.  
  
Pruebas de Aceptación del Usuario (UAT):  
Las pruebas de aceptación del usuario (User Acceptance Testing - UAT) se realizan con usuarios finales para validar que la plataforma MediConecta cumpla con las expectativas y requisitos de los usuarios en un entorno real:  
  
Proceso de Pruebas UAT: Los usuarios finales, que incluyen pacientes, doctores y administradores, utilizan la plataforma en condiciones simuladas para probar funcionalidades clave como el inicio de sesión, gestión de citas y consulta de indicadores de salud. Los usuarios proporcionan feedback sobre la usabilidad y eficiencia del sistema.  
  
Criterios de Éxito: Los criterios de aceptación incluyen la facilidad de uso, la correcta funcionalidad de las características clave, la velocidad de respuesta y la claridad de la interfaz. Cualquier problema reportado por los usuarios se documenta y se soluciona antes del lanzamiento oficial de la plataforma.  
  
Estas pruebas aseguran que MediConecta sea un sistema confiable, de alto rendimiento y fácil de usar, brindando una experiencia satisfactoria a todos sus usuarios.

# 7. Despliegue e Infraestructura

Entorno de Desarrollo y Producción:  
MediConecta se despliega en dos entornos distintos para facilitar el desarrollo, pruebas y operación del sistema:  
  
Entorno de Desarrollo: Configurado localmente o en un servidor de pruebas, este entorno permite que los desarrolladores realicen cambios y pruebas sin afectar la producción. Utiliza una versión de la base de datos menos restrictiva y permite la depuración activa.  
  
Entorno de Producción: Este es el entorno en el que los usuarios acceden a la plataforma. Configurado en un servidor seguro, incluye optimizaciones de rendimiento y seguridad, así como acceso a la base de datos de producción con cifrado de datos en reposo y en tránsito.  
  
Proceso de Despliegue:  
Para desplegar MediConecta en producción, se siguen los siguientes pasos:  
  
Configuración del Servidor: Instalar el sistema operativo, el entorno de ejecución y las herramientas de administración necesarias (por ejemplo, NGINX o Apache).  
Base de Datos: Configuración y migración de la base de datos PostgreSQL, incluyendo los datos iniciales si es necesario.  
Despliegue de Backend y Frontend: Subir el código al servidor de producción y configurar el acceso a través de la API.  
Pruebas Finales y Validación: Ejecutar pruebas de rendimiento y validación en producción antes del lanzamiento oficial.  
Integración Continua y Entrega Continua (CI/CD):  
MediConecta utiliza herramientas como GitLab CI/CD para automatizar la compilación, pruebas y despliegue. Estas herramientas aseguran que cada cambio en el código pase por un pipeline de pruebas antes de integrarse en producción, reduciendo errores y garantizando la estabilidad del sistema.  
  
Monitorización y Logs:  
Herramientas como New Relic o Grafana se utilizan para monitorear el rendimiento del sistema en tiempo real, rastrear métricas de uso y detectar problemas. El sistema de registro de logs permite identificar y solucionar errores rápidamente, mejorando la confiabilidad de MediConecta.

# 8. Seguridad

Autenticación y Autorización:  
MediConecta implementa JWT (JSON Web Tokens) para autenticar usuarios y gestionar sesiones. Los roles (Administrador, Doctor, Paciente) definen los permisos de cada usuario, asegurando que solo tengan acceso a las funciones y datos que les corresponden.  
  
Cifrado de Datos:  
Se utiliza HTTPS para el cifrado en tránsito, asegurando que la información transferida esté protegida. Los datos en reposo en la base de datos PostgreSQL están cifrados para evitar el acceso no autorizado.  
  
Protocolos de Seguridad:  
MediConecta cuenta con firewalls para proteger contra accesos no autorizados y realiza auditorías de seguridad periódicas. Estas medidas incluyen configuraciones para detectar y prevenir amenazas, cumpliendo con las mejores prácticas de seguridad.

# 9. Manual del Usuario

Instrucciones de Inicio de Sesión:  
Para registrarse, los usuarios completan un formulario con su información básica y el sistema les asigna un rol. En caso de perder su contraseña, pueden recuperarla a través de la opción "¿Olvidaste tu contraseña?".  
  
Gestión de Citas:  
Los pacientes pueden seleccionar un doctor, una fecha y una hora disponible para reservar una cita. Además, tienen la opción de ver y cancelar citas previamente agendadas.  
  
Monitoreo de Salud:  
Los pacientes pueden registrar sus indicadores de salud y ver el historial de sus mediciones en forma de gráficos. Esto les permite monitorear su estado de salud con facilidad.  
  
Preguntas Frecuentes (FAQ):  
Sección que responde a preguntas comunes, como "¿Cómo me registro?", "¿Cómo reservo una cita?" y "¿Cómo puedo actualizar mis datos personales?".

# 10. Mantenimiento y Actualizaciones

Guía de Mantenimiento:  
Instrucciones para realizar mantenimiento en la base de datos y en el API, como optimización de consultas y limpieza de datos obsoletos. El frontend también recibe actualizaciones periódicas para mejorar la experiencia de usuario.  
  
Actualización de Software:  
Cada actualización pasa por pruebas en el entorno de desarrollo antes de ser lanzada en producción. Las pruebas incluyen validación de nuevas funcionalidades y regresión para asegurar la estabilidad.  
  
Resolución de Problemas Comunes:  
Instrucciones paso a paso para resolver problemas técnicos, como errores de inicio de sesión o fallos en la carga de datos. También incluye guías para restaurar servicios en caso de caídas menores.

# 11. Conclusión

El Product Roadmap Técnico y esta documentación aseguran una planificación y ejecución estructurada para MediConecta, con un enfoque en la accesibilidad, seguridad y eficiencia. Desde el desarrollo y pruebas hasta el despliegue y mantenimiento, cada etapa del proyecto contribuye a crear una plataforma confiable para la gestión de la salud de adultos mayores. Esta documentación proporciona una referencia completa para el equipo de desarrollo y los usuarios finales, asegurando que MediConecta se mantenga como una solución efectiva y de alto impacto para la mejora de la calidad de vida de sus usuarios.