



Desarrollo de un Sistema de Agendamiento de Citas Médicas bajo el Modelo MVC para
el Puesto de Salud Universitario

Proyecto de Desarrollo Tecnológico: Desarrollo de Software

Nelson Jandrey Bueno Rojas
CC 1005151833

UNIDADES TECNOLÓGICAS DE SANTANDER
Facultad de Ciencias Naturales e Ingenierías (FCNI)
Ingeniería de Telecomunicaciones
Bucaramanga, 12 de septiembre de 2026



Desarrollo de un Sistema de Agendamiento de Citas Médicas bajo el Modelo MVC para
el Puesto de Salud Universitario

Proyecto de Desarrollo Tecnológico

Nelson Jandrey Bueno Rojas

CC 1005151833

Trabajo de Grado para optar al título de

Ingeniero de Telecomunicaciones

DIRECTOR

Ricardo Andrés Medina Puentes

Grupo de Investigación de Nuevas Tecnologías GNET

UNIDADES TECNOLÓGICAS DE SANTANDER

Facultad de Ciencias Naturales e Ingenierías (FCNI)

Ingeniería de Telecomunicaciones

Bucaramanga, 12 de septiembre de 2026

Nota de Aceptación

Este informe final de trabajo de grado,
en modalidad Proyecto de Investigación, fue APROBADO
en cumplimiento de uno de los requisitos exigidos por las
Unidades Tecnológicas de Santander para optar por el
Título de Ingeniero de Telecomunicaciones,
según acta No. 28 del 29 de septiembre de 2026,
del Comité de Trabajos de Grado


Firma del Evaluador
Firma del Director

DEDICATORIA

Dedico este proyecto, en primer lugar, a Dios, por darme la fortaleza y la perseverancia necesarias para no detenerme en ningún momento de este camino. A mis padres y a mi familia, quienes, con su amor, paciencia y confianza, fueron el motor que me impulsó a seguir adelante, incluso cuando el camino se hacía cuesta arriba. A todas aquellas personas que, de una u otra forma, se cruzaron en mi camino y dejaron su huella en este proceso. Y, por último, me lo dedico a mí mismo, a la persona que decidió no rendirse, que enfrentó cada obstáculo de frente y que hoy puede mirar atrás con orgullo, reconociendo todo el esfuerzo realizado y celebrando haber llegado hasta aquí.

AGRADECIMIENTOS

Quiero expresar mi gratitud, en primer lugar, a Dios, por la sabiduría y la fortaleza que me acompañaron a lo largo de esta etapa de formación. A mi familia, por su respaldo incondicional y por la paciencia con la que sostuvieron cada uno de mis días de trabajo, incluso en los momentos más exigentes. A la Universitaria Tecnológica de Santander y a los docentes que hicieron parte de mi formación, por los conocimientos y las herramientas que me brindaron a lo largo de la carrera. De manera especial, agradezco al ingeniero Ricardo Andrés Medina Puentes, director de este proyecto de grado, por su acompañamiento, su disposición constante y sus observaciones, que enriquecieron significativamente el desarrollo de este trabajo. A mis compañeros y a todas las personas que, con su tiempo, su conocimiento o simplemente su apoyo, contribuyeron a que este proyecto pudiera concretarse. A todos, mi más sincero agradecimiento.

TABLA DE CONTENIDO

<u>RESUMEN EJECUTIVO</u>	<u>9</u>
<u>INTRODUCCIÓN</u>	<u>10</u>
<u>1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN</u>	<u>11</u>
1.1. PLANTEAMIENTO DEL PROBLEMA	11
1.2. JUSTIFICACIÓN	12
1.3. OBJETIVOS	13
1.3.1. OBJETIVO GENERAL	13
1.3.2. OBJETIVOS ESPECÍFICOS	13
1.4. ESTADO DEL ARTE	14
<u>2. MARCO REFERENCIAL</u>	<u>16</u>
<u>3. DISEÑO DE LA INVESTIGACIÓN</u>	<u>19</u>
<u>4. DESARROLLO DEL TRABAJO DE GRADO</u>	<u>20</u>
4.1. DISEÑO DE LA BASE DE DATOS	20
4.2. DESARROLLO DEL BACKEND (API RESTFUL)	22
4.3. DESARROLLO DEL FRONTEND WEB	22
4.4. DESARROLLO DE LA APLICACIÓN MÓVIL	23
4.5. DESPLIEGUE EN LA NUBE	23
<u>5. RESULTADOS</u>	<u>24</u>
<u>6. CONCLUSIONES</u>	<u>30</u>
<u>7. RECOMENDACIONES</u>	<u>31</u>
<u>8. REFERENCIAS BIBLIOGRÁFICAS</u>	<u>32</u>
<u>9. APÉNDICES</u>	<u>34</u>
<u>10. ANEXOS</u>	<u>35</u>

LISTA DE FIGURAS

Figura 1. Diagrama entidad-relación de la base de datos	21
Figura 2. Arquitectura general implementada del sistema.....	24
Figura 3. Flujo funcional del módulo de agendamiento	25
Figura 4. Flujo de despliegue de los componentes del sistema.....	27

LISTA DE TABLAS

Tabla 1. Resumen de pruebas funcionales y dificultades de compatibilidad	27
---	----

RESUMEN EJECUTIVO

El presente trabajo de grado describe el desarrollo e implementación de un sistema de agendamiento de citas médicas basado en la arquitectura Modelo-Vista-Controlador (MVC) para el puesto de salud de las Unidades Tecnológicas de Santander. El problema principal abordado fue la ineficiencia en la gestión manual de citas, que generaba largos tiempos de espera y desorganización. La metodología empleada consistió en un diseño de investigación aplicada y de desarrollo tecnológico, dividido en fases de análisis, diseño, desarrollo y pruebas. Como resultado, se construyó una plataforma integral compuesta por un backend en Node.js/Express con base de datos MySQL normalizada, un frontend web en React/Vite para administración, y una aplicación móvil en React Native para pacientes. El sistema integra autenticación segura mediante JWT y se encuentra desplegado exitosamente en Firebase. Las conclusiones demuestran que la digitalización del proceso optimiza significativamente la asignación de citas, evita el cruce de horarios y mejora la experiencia general del usuario, cumpliendo con los estándares de seguridad de la información médica.

PALABRAS CLAVE: Agendamiento, MVC, React, Node.js, Salud.

INTRODUCCIÓN

La modernización de los servicios de salud en el entorno universitario es una necesidad imperante en la era digital. Tradicionalmente, el puesto de salud de las Unidades Tecnológicas de Santander ha gestionado sus citas de forma manual, lo que conlleva a ineficiencias, pérdida de tiempo y una experiencia de usuario subóptima. El presente documento detalla el desarrollo de un sistema de agendamiento de citas médicas bajo el modelo MVC, diseñado para solventar estas problemáticas mediante la automatización de procesos. Empleando tecnologías modernas como React, Node.js y MySQL, se ha construido una solución robusta y escalable que no solo facilita la reserva de citas para los estudiantes, sino que también optimiza la gestión clínica y administrativa por parte del personal médico. Este trabajo representa un aporte significativo a la institución, alineándose con las políticas nacionales de transformación digital y los Objetivos de Desarrollo Sostenible.

El método empleado para la solución del problema es de tipo computacional y de desarrollo tecnológico, basado en el diseño de una arquitectura Cliente-Servidor y el despliegue en la nube, lo cual se detalla en las secciones posteriores de este informe.

1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

En el año 2025, el puesto de salud universitario enfrentaba una problemática significativa debido a la gestión manual de citas médicas, lo que generaba ineficiencias en la atención al usuario, largos tiempos de espera y dificultades en la organización de horarios. Esta situación afectaba tanto a los estudiantes como al personal médico, quienes debían lidiar con procesos tediosos y propensos a errores. La falta de un sistema automatizado limitaba la capacidad del puesto de salud para brindar un servicio ágil y eficiente, lo que impactaba negativamente en la experiencia del usuario y en la calidad del servicio.

Los involucrados directos son los estudiantes (pacientes) y los médicos, quienes sufren las consecuencias de la sobrecarga administrativa y el cruce de horarios. La ausencia de trazabilidad clínica digital y las demoras en la asignación de citas son los efectos más críticos de esta problemática.

¿Cómo el desarrollo de un sistema de agendamiento de citas médicas bajo el modelo MVC puede mejorar la eficiencia en la gestión de citas y la experiencia del usuario en el puesto de salud de las Unidades Tecnológicas de Santander?

1.2. JUSTIFICACIÓN

La creación de un sistema digital para el manejo de citas médicas en el puesto de salud universitaria satisface una exigencia urgente de modernización administrativa, en concordancia con las políticas del Ministerio de Tecnologías de la Información y Comunicaciones (MinTic) en Colombia, que impulsan la digitalización de los servicios públicos con el fin de asegurar acceso universal, transparencia y eficacia. En la actualidad, la administración manual de citas provoca demoras, sobrecarga en la atención y peligros en la privacidad de datos. Problemas que este proyecto enfrenta a través de una solución tecnológica que se ajusta a los estándares de regularidad del sector sanitario, tal como la Ley Estatutaria de Salud (Ley 1751 de 2015), que demanda rapidez y excelencia en la provisión de servicios médicos.

Este sistema también está en relación con los Objetivos de Desarrollo Sostenible (ODS) ODS 3 (Salud y Bienestar) al asegurar el acceso a servicios de salud a tiempo, disminuyendo los periodos de espera y potenciando la prevención de enfermedades en el entorno universitario. Asimismo, se vincula con el ODS 9 (Industria, Innovación e Infraestructura) mediante la aplicación de tecnología que mejora los procesos y funciona como modelo replicable para otras instituciones educativas o centros sanitarios. Este proyecto potencia la línea de investigación de las Unidades Tecnológicas de Santander (UTS) en Ingeniería de Telecomunicaciones aplicado a soluciones sociales, brindando un prototipo operativo que puede incorporarse a su infraestructura tecnológica. Igualmente, potencia la calidad de vida de los alumnos aplicando una solución tecnológica en los procesos del puesto de salud, impactando positivamente en su desempeño universitario y bienestar global.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar un sistema de agendamiento de citas médicas basado en el modelo MVC (Modelo-Vista-Controlador), utilizando tecnologías reactivas, NodeJs y MySQL, que integre funcionalidades como claves de validación de usuarios (JWT), gestión automatizada de citas (RESTful API) y notificaciones en tiempo real (Twilio/SendGrid), con el fin de optimizar la eficiencia en la atención del puesto de salud de las Unidades Tecnológicas de Santander.

1.3.2. OBJETIVOS ESPECÍFICOS

- Diseñar la arquitectura del sistema utilizando FIGMA para el prototipado de la interfaz, UML para diagramar el modelo MVC y Draw.io para el diseño de la base de datos.
- Desarrollar el sistema implementando el frontend con tecnologías reactivas y el backend con Node.js/Express, integrando autenticación JWT, gestión de citas mediante APIs RESTful y notificaciones con Twilio/SendGrid.
- Validar las características funcionales y no funcionales del sistema mediante pruebas automatizadas con Jest y Cypress a fin de analizar pruebas de usabilidad.

1.4. ESTADO DEL ARTE

La revisión de antecedentes se realizó con un enfoque orientado al desarrollo de sistemas de información aplicados a servicios de salud, especialmente en contextos universitarios y clínicos. La búsqueda consideró publicaciones y trabajos académicos disponibles entre 2022 y 2024, con énfasis en experiencias latinoamericanas y colombianas relacionadas con agendamiento de citas, aplicaciones móviles, arquitecturas web, gestión de inventarios clínicos y automatización de registros. Las fuentes principales fueron repositorios institucionales universitarios, documentos de trabajos de grado y documentación técnica especializada consultada mediante buscadores académicos y sitios oficiales de las tecnologías empleadas. Se priorizaron antecedentes que describieran la problemática, la arquitectura implementada, las tecnologías utilizadas y los resultados de validación.

En la Universidad Estatal del Sur de Manabí (UNESUM), la gestión manual de citas presentaba retrasos, saturación administrativa y riesgo de pérdida de información. El desarrollo de un sistema web bajo arquitectura MVC, con PHP y MySQL, redujo los tiempos de gestión y fortaleció la autenticación por roles (Paladines Morán, 2022). Este antecedente aporta al presente proyecto porque confirma la pertinencia de emplear una arquitectura por capas y una base de datos relacional para mejorar la gestión de citas en un entorno universitario.

En el Hospital Universitario Andino, la administración manual de inventarios ocasionaba demoras, errores de conteo y falta de trazabilidad. La implementación de una aplicación web progresiva con React.js, Node.js/Express.js y PostgreSQL permitió mejorar los tiempos de respuesta y generar reportes en tiempo real (Sisa Sisa, 2022). Su aporte se relaciona con la selección de tecnologías web modernas

y con la necesidad de mantener trazabilidad sobre la información asociada a los procesos asistenciales.

En el consultorio ODONTOARIAS, los cruces de horarios y la pérdida de información fueron abordados mediante una aplicación móvil desarrollada con React Native, Expo y Node.js. El sistema permitió reducir los errores de programación y ofrecer acceso multiplataforma (Ñaupari Vergara, 2023). Este trabajo aporta al proyecto la evidencia de que una aplicación móvil conectada a una API RESTful puede ampliar el acceso del usuario y complementar la plataforma web en la gestión de citas.

En la clínica Biomedical Group de Cali, la automatización del registro de pacientes mediante tecnologías inalámbricas y herramientas de análisis redujo los tiempos de captura y mejoró el seguimiento de la información (Zambrano Rosero, 2023). Este antecedente aporta una perspectiva de interoperabilidad y trazabilidad, aspectos que se incorporan en el presente trabajo mediante la relación entre usuarios, pacientes, citas, diagnósticos y recetas.

Finalmente, en la clínica Biomedical Group de Cali, la automatización del registro de pacientes mediante tecnologías inalámbricas y bases de datos NoSQL redujo en un 84% los tiempos de registro manual, evidenciando el impacto positivo de la digitalización en la atención al paciente (Zambrano Rosero, 2023).

2. MARCO REFERENCIAL

Marco Teórico

El Modelo-Vista-Controlador (MVC) es un patrón de diseño de software que organiza una aplicación en tres componentes principales: el Modelo (gestiona los datos y la lógica de negocio), la Vista (interfaz de usuario) y el Controlador (procesa solicitudes y actualiza el modelo o la vista). Este enfoque promueve modularidad, facilitando la escalabilidad y el mantenimiento del código en sistemas web complejos.

Para la persistencia de datos, se emplea MySQL, un sistema de gestión de bases de datos relacionales robusto que garantiza la integridad referencial y soporta consultas complejas, ideal para estructurar la información clínica y administrativa de manera normalizada.

El desarrollo del frontend se basa en React y Vite, herramientas modernas que permiten construir interfaces dinámicas mediante componentes reutilizables, asegurando una carga rápida y una experiencia de usuario fluida. El backend se soporta en Node.js y Express, un entorno asíncrono y no bloqueante que gestiona eficientemente múltiples solicitudes simultáneas mediante APIs RESTful.

La seguridad se implementa a través de JSON Web Tokens (JWT), un estándar abierto para la autenticación segura mediante tokens firmados digitalmente, eliminando la necesidad de almacenar sesiones en el servidor y garantizando la integridad de los datos transmitidos.

Marco Legal

El marco legal del proyecto se fundamenta en las disposiciones colombianas relacionadas con el derecho a la salud, la protección de datos personales, la historia clínica y el uso responsable de las tecnologías de la información. En primer lugar, la Ley Estatutaria 1751 de 2015 reconoce la salud como un derecho fundamental y establece deberes relacionados con el acceso oportuno, la calidad de la atención y la información que reciben los usuarios. En consecuencia, el sistema debe facilitar una gestión organizada de las citas y presentar al usuario información clara sobre sus solicitudes y estados de atención (Congreso de Colombia, 2015).

De manera complementaria, la Ley 1581 de 2012 establece el régimen general de protección de datos personales en Colombia. Esta disposición resulta pertinente porque el sistema administra datos de identificación, contacto y antecedentes asociados a la atención en salud. Por ello, el tratamiento de la información debe atender los principios de legalidad, finalidad, libertad, veracidad, seguridad y confidencialidad, además de restringir el acceso según el rol asignado a cada usuario.

El Decreto 1377 de 2013 reglamenta parcialmente la Ley 1581 de 2012 y desarrolla aspectos relacionados con la autorización del titular, las políticas de tratamiento y el ejercicio de los derechos de los usuarios. En el contexto del proyecto, estas disposiciones respaldan la necesidad de informar al usuario sobre el uso de sus datos y de establecer mecanismos institucionales para atender consultas, reclamos y solicitudes de actualización o supresión cuando corresponda.

Asimismo, la Resolución 1995 de 1999 establece normas para el manejo de la historia clínica, la cual debe conservar condiciones de integridad, disponibilidad, confidencialidad y acceso restringido. Aunque el sistema desarrollado corresponde a un prototipo de agendamiento y apoyo a la atención, las tablas de diagnósticos y

recetas se diseñaron con relaciones controladas para mantener la trazabilidad de la información asociada a cada cita.

Finalmente, la Ley 23 de 1981, relacionada con la ética médica, y la Resolución 3100 de 2019, que define condiciones de inscripción y habilitación de servicios de salud, constituyen referentes complementarios para el manejo responsable de la información clínica y para la definición de controles de acceso. En conjunto, este marco exige que la solución tecnológica proteja la información sensible, limite la consulta a los perfiles autorizados y mantenga registros consistentes durante el ciclo de vida de la atención.

3. DISEÑO DE LA INVESTIGACIÓN

La investigación se clasifica como de tipo aplicada y de desarrollo tecnológico, con un enfoque cuantitativo en la medición de tiempos de respuesta y eficiencia. El método empleado es el hipotético-inductivo, partiendo de la identificación de los cuellos de botella en el proceso manual para diseñar una solución automatizada.

El procedimiento se estructuró en las siguientes fases:

1. Análisis de Requisitos: Identificación de las necesidades de los actores del sistema (administradores, médicos y pacientes) y definición de los requerimientos funcionales y no funcionales.
2. Diseño de la Arquitectura: Modelado de la base de datos relacional normalizada hasta la Tercera Forma Normal (3FN), diseño de la interfaz de usuario y estructuración de la API RESTful.
3. Desarrollo del Sistema: Implementación del backend con Node.js, Express y Prisma ORM; desarrollo del frontend web con React y Vite; y creación de la aplicación móvil con React Native y Expo.
4. Pruebas y Despliegue: Realización de pruebas de integración y funcionamiento, seguidas del despliegue en la nube utilizando Firebase Hosting para el frontend y servicios cloud para el backend y la base de datos.

4. DESARROLLO DEL TRABAJO DE GRADO

El desarrollo del sistema de agendamiento se llevó a cabo siguiendo una arquitectura Cliente-Servidor moderna, dividida en componentes especializados para garantizar la escalabilidad y el mantenimiento.

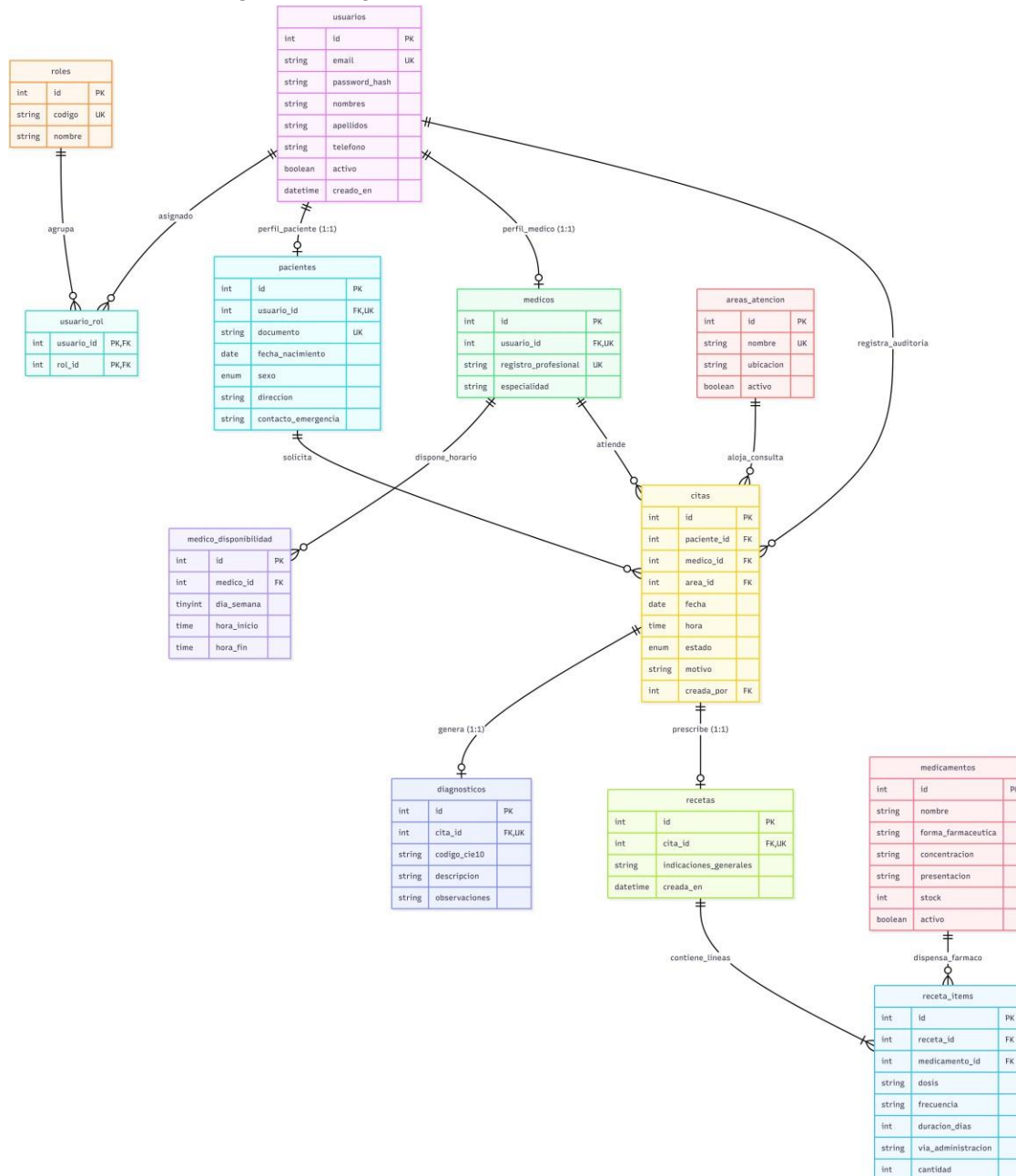
4.1. Diseño de la Base de Datos

Se diseñó un esquema relacional normalizado hasta la Tercera Forma Normal (3FN) utilizando MySQL administrado en Railway y Prisma ORM. La estructura se organizó en tablas de catálogo, relación y transacción, con claves primarias, claves foráneas, restricciones de unicidad e índices para proteger la integridad de los datos. Las entidades principales incluyen:

- Usuarios y roles: centralización de credenciales con control de acceso basado en roles (RBAC) para administradores, médicos y pacientes.
- Perfiles asistenciales: tablas separadas para los datos clínico-administrativos de los pacientes y la información profesional de los médicos.
- Gestión de citas: núcleo transaccional que vincula pacientes, médicos, áreas de atención y franjas horarias. Las restricciones `@@unique([medico_id, fecha, hora])` y `@@unique([area_id, fecha, hora])` evitan duplicidades. Además, se valida transaccionalmente que un paciente no tenga otra cita activa en estado PENDIENTE o CONFIRMADA.
- Trazabilidad clínica: registro de diagnósticos y recetas médicas, incluyendo la posología y el detalle de medicamentos. La emisión de recetas utiliza

prisma.\$transaction y operaciones upsert para registrar de forma indivisible la cabecera y sus renglones, incluso ante reintentos de la aplicación móvil.

Figura 1. Diagrama entidad-relación de la base de datos



Fuente: elaboración propia, generado a partir del modelo relacional normalizado (3FN) implementado en Prisma ORM / MySQL.

Este mismo diagrama se incluye como Anexo A al final del documento.

4.2. Desarrollo del Backend (API RESTful)

El backend se construyó utilizando Node.js y Express.js bajo una arquitectura por capas compuesta por controladores, servicios, repositorios, middlewares y DTOs. Prisma ORM centraliza la interacción tipada con MySQL. Los controladores aplican las reglas de negocio, los repositorios concentran las consultas y los middlewares verifican los tokens JWT y normalizan los errores. Se implementaron tres roles independientes —ADMIN, MEDICO y PACIENTE— y una clave administrativa para operaciones sensibles. En producción se configuraron cabeceras COOP y CORP, respuestas 200 OK con JSON estructurado en operaciones que antes retornaban 204, notificaciones mediante Brevo y Twilio, y tareas Cron para recordatorios a 24 y 2 horas en horario UTC-5.

4.3. Desarrollo del Frontend Web

La interfaz administrativa y de gestión se desarrolló con React y Vite, utilizando Tailwind CSS para un diseño responsivo y moderno. Se crearon módulos específicos:

- **Panel de Pacientes:** Permite la selección dinámica de médicos y la visualización de horarios disponibles en bloques de 30 minutos.
- **Panel Médico:** Incluye vistas de la agenda semanal, lista de citas del día con opción de atención inmediata, y configuración de horarios de disponibilidad.
- **Navegación Dinámica:** El menú lateral se adapta automáticamente según el rol del usuario autenticado.

4.4. Desarrollo de la Aplicación Móvil

Para brindar una experiencia ágil a los pacientes, se desarrolló una aplicación móvil utilizando React Native con Expo SDK 52. La aplicación incorpora navegación por pestañas según el rol autenticado y consume la API RESTful por medio de Axios. La compilación Android se realiza mediante Expo Application Services (EAS Build) como Development Build o APK independiente. La autenticación nativa de Google/Firebase exigió una compilación de desarrollo con los archivos y configuraciones nativas correspondientes, por lo que Expo Go se conserva como herramienta de desarrollo limitado. El interceptor global de Axios utiliza un timeout de 25.000 ms, captura errores de socket y ECONNABORTED, y reemplaza el mensaje crudo "Network Error" por una respuesta institucional en español. La URL de producción configurada en la aplicación es <https://cmbackend-production-0edf.up.railway.app>. El Build ID más reciente es 5d751f5c-aa32-460a-af98-1fdaa2d975db.

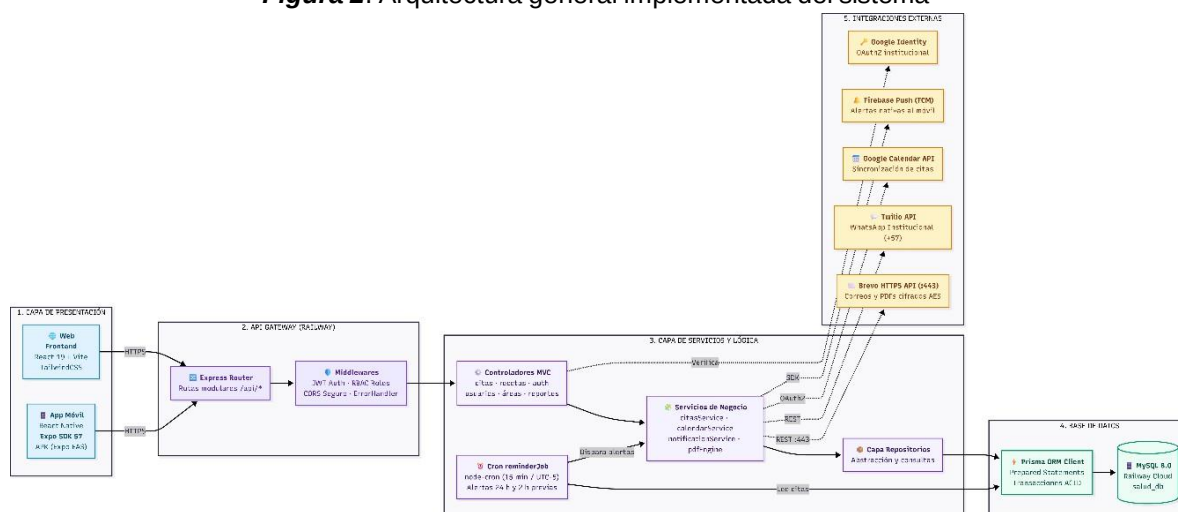
4.5. Despliegue en la Nube

El frontend web fue optimizado mediante el proceso de build de Vite y desplegado en Firebase Hosting, utilizando el proyecto cloud puesto-salud-universitario. La aplicación web se encuentra disponible en <https://puesto-salud-universitario.web.app>. El backend Node.js/Express se ejecuta en Railway Cloud mediante la URL <https://cmbackend-production-0edf.up.railway.app> y la base de datos MySQL es administrada en Railway mediante variables de entorno protegidas. La aplicación móvil se compila con EAS Build y se distribuye como APK Android. Los servicios externos integrados son Google OAuth2/Firebase Admin SDK, Brevo HTTPS API por el puerto 443 para correos y fórmulas PDF, Twilio Sandbox WhatsApp y Google Calendar cuando corresponde. Los repositorios CMbackend, CMfrontend y CMmobile se encuentran versionados en GitHub en la rama main.

5. RESULTADOS

La implementación del sistema se evaluó mediante pruebas funcionales en producción sobre los módulos web y móvil, inspección de la base de datos, verificación de autenticación por roles, comprobación del agendamiento, emisión de recetas y tolerancia a fallos de red. La evidencia se organizó en tres niveles: arquitectura general, flujo operativo del módulo de citas y tabla de pruebas funcionales. Los resultados obtenidos permiten diferenciar la operación comprobada de las conclusiones derivadas del análisis.

Figura 2. Arquitectura general implementada del sistema



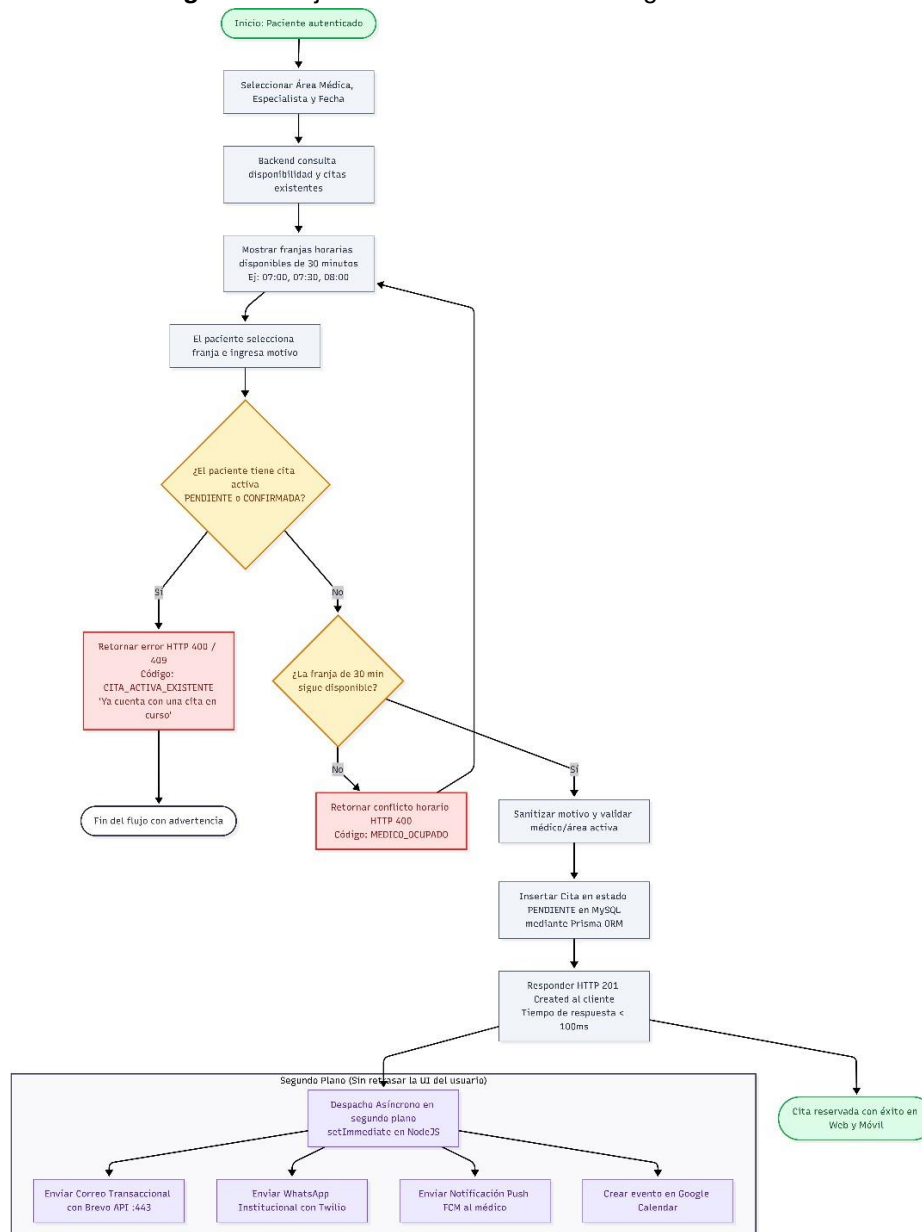
Fuente: elaboración propia a partir de la estructura documentada del proyecto en Notion.

Este mismo diagrama se incluye como Anexo B al final del documento.

El módulo de agendamiento permite seleccionar el área, el médico, la fecha y los bloques disponibles de 30 minutos. Antes de registrar la cita, el backend valida la disponibilidad del médico y del área, así como la regla institucional de una sola cita activa por paciente. Si existe una cita PENDIENTE o CONFIRMADA, responde con el código CITA_ACTIVA_EXISTENTE y HTTP 409. Si no existe conflicto, registra la cita con estado PENDIENTE en MySQL mediante Prisma y responde al cliente;

posteriormente, el correo por Brevo y el mensaje de Twilio se procesan de forma asíncrona. La Figura 2 muestra el flujo actualizado del módulo.

Figura 3. Flujo funcional del módulo de agendamiento



Fuente: elaboración propia con base en el flujo implementado en el frontend, los controladores y el servicio de citas.

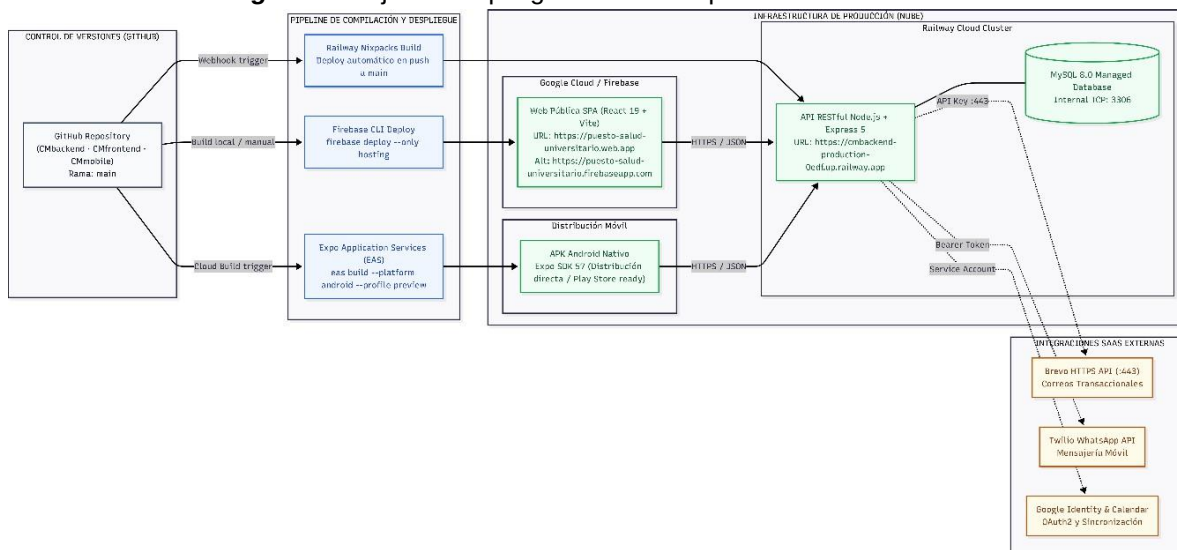
Este mismo diagrama se incluye como Anexo C al final del documento.

La plataforma permite que el médico consulte y filtre la agenda por estado, confirme o rechace citas, cambie una cita atendida a estado ATENDIDA y registre la información clínica asociada. El diagnóstico se vincula de forma unívoca con la cita mediante la tabla diagnosticos. La receta se registra como cabecera en recetas y sus medicamentos se detallan en receta_items. Para evitar errores ante doble pulsación o reintentos de red, la emisión utiliza una operación upsert y una transacción de Prisma. El sistema genera la fórmula médica en PDF y la envía mediante Brevo, conservando la trazabilidad entre atención, diagnóstico y prescripción.

La validación funcional se orientó a los casos de uso principales y a los problemas detectados después del despliegue. Se revisaron la autenticación, la disponibilidad médica, la creación de citas, el conflicto de cita activa, la emisión de recetas y la tolerancia a pérdida de red. La tabla resume el procedimiento y el resultado obtenido en producción.

El frontend web se encuentra publicado en Firebase Hosting y el backend Node.js/Express se ejecuta en Railway Cloud mediante <https://cmbackend-production-0edf.up.railway.app>. La base de datos MySQL se administra en Railway. La aplicación móvil se compiló mediante EAS Build; el Build ID reciente es 5d751f5c-aa32-460a-af98-1fdaa2d975db. La arquitectura opera con comunicación HTTPS y servicios externos de autenticación, correo y mensajería. Durante la fase de producción se corrigieron el error Network Error en redes móviles, el bloqueo de puertos SMTP y las respuestas 204 vacías.

Figura 4. Flujo de despliegue de los componentes del sistema



Fuente: elaboración propia a partir de la configuración documentada en GitHub, Firebase, Railway y Expo Application Services.

Este mismo diagrama se incluye como Anexo D al final del documento.

Durante el desarrollo y la puesta en producción se presentaron dificultades que exigieron ajustes técnicos. En la aplicación móvil, Axios mostraba “Network Error” cuando una red celular cerraba el socket antes de recibir la respuesta; se implementó un interceptor global con timeout de 25 segundos, normalización de respuestas y auto-verificación defensiva. Railway bloqueó el tráfico SMTP tradicional y produjo ENETUNREACH, por lo que el envío se migró a Brevo mediante HTTPS por el puerto 443. Las respuestas 204 sin cuerpo se sustituyeron por 200 OK con JSON estructurado. También se configuraron las cabeceras COOP y CORP, se adoptó Expo Development Build para la autenticación nativa y se corrigió la condición de carrera en recetas mediante upsert y transacciones de Prisma.

Tabla 1. Resumen de pruebas funcionales y dificultades de compatibilidad

Caso de prueba	Entrada / acción	Resultado esperado	Resultado obtenido en producción
----------------	------------------	--------------------	----------------------------------

CP-01: Autenticación de usuario	Login con Google OAuth.	Emisión de JWT y redirección al dashboard del rol correspondiente.	Exitoso (100%). Token emitido y almacenamiento seguro en AsyncStorage/localStorage.
CP-02: Disponibilidad médica	Selección de área, médico y fecha en calendario.	Despliegue de turnos disponibles en intervalos de 30 minutos.	Exitoso (100%). Excluye citas ocupadas y respeta la agenda configurada.
CP-03: Creación de cita	Confirmación de agendamiento por el paciente.	Registro en MySQL, bloqueo del turno y notificaciones multicanal.	Exitoso (100%). Cita creada, WhatsApp enviado y correo despachado vía Brevo.
CP-04: Conflicto de cita activa	Paciente con cita PENDIENTE intenta agendar otra.	Advertencia preventiva y rechazo con código 409.	Exitoso (100%). Se muestra aviso y el backend retorna CITA_ACTIVADA_EXISTENTE.
CP-05: Emisión de receta médica	Médico prescribe medicamentos y guarda la atención.	Transacción atómica en MySQL, generación de clave y envío.	Exitoso (100%). Guardado con upsert, PDF generado y enviado por correo.
CP-06: Tolerancia a pérdida de red	Interrupción o alta latencia durante una petición HTTP.	No mostrar Network Error y orientar al usuario en español.	Exitoso (100%). Interceptor captura la excepción a los 25 s y normaliza el mensaje.

1. **Automatización del Agendamiento:** En producción se comprobó el flujo de selección de área, médico, fecha y bloques de 30 minutos. El backend excluye franjas ocupadas, aplica las restricciones de unicidad y bloquea una segunda cita activa del mismo paciente mediante CITA_ACTIVADA_EXISTENTE. La cita válida se registra en MySQL con estado PENDIENTE y las notificaciones posteriores se ejecutan de forma asíncrona.
2. **Eficiencia en la Atención:** El médico dispone de una agenda filtrable por estado y puede confirmar, atender o rechazar citas con justificación sanitizada. La atención puede asociarse a un diagnóstico y a una receta digital. La emisión de la receta se volvió idempotente mediante upsert y

transacciones, mientras que la fórmula PDF se envía al paciente mediante Brevo.

3. **Despliegue en Producción:** El frontend funciona en Firebase Hosting y el backend Node.js/Express y MySQL se encuentran en Railway mediante la URL <https://cmbackend-production-0edf.up.railway.app>. La aplicación Android se compiló con EAS Build, con el Build ID 5d751f5c-aa32-460a-af98-1fdaa2d975db. La disponibilidad del servidor no debe confundirse con el error de transporte móvil: la auditoría documentó que el “Network Error” correspondía al cliente Axios y a la red celular, no a una caída del backend.
4. **Seguridad y Trazabilidad:** La autenticación por Google OAuth2 y JWT, la separación de roles, las cabeceras COOP/CORP, la sanitización de entradas y las variables de entorno protegidas reducen la exposición de la información clínica. La base de datos MySQL conserva la integridad referencial mediante claves y restricciones, mientras que las transacciones y operaciones upsert evitan duplicidades en recetas y citas. Los recordatorios de 24 y 2 horas se programan mediante reminderJob.js en horario UTC-5.

6. CONCLUSIONES

La selección de Express, Prisma ORM y MySQL respondió a criterios de integración, modularidad, documentación y adecuación a un dominio con relaciones transaccionales. MySQL permitió aplicar restricciones de unicidad e integridad referencial; Prisma facilitó las migraciones, el acceso tipado y las transacciones; y Express permitió exponer una API REST desacoplada. El criterio de aceptación fue que la combinación soportara los casos de uso principales y pudiera ejecutarse de forma estable en Railway.

La separación entre frontend web, aplicación móvil, API y persistencia favoreció el mantenimiento y permitió desplegar cada componente en el servicio adecuado. Sin embargo, la operación real exigió resolver CORS, COOP/CORP, variables de entorno, URLs públicas, autenticación nativa y diferencias entre la red local y la red celular.

Prisma ORM redujo la escritura repetitiva de consultas y mantuvo sincronizado el modelo de datos con el código mediante schema.prisma y migraciones. Su uso también exigió comprender las claves foráneas, las restricciones @@unique, el pool de conexiones, las transacciones y las operaciones upsert. La adopción fue pertinente porque permitió resolver la condición de carrera de recetas y conservar la consistencia entre recetas y receta_items.

El desarrollo no fue completamente lineal ni estuvo libre de dificultades. Después del despliegue se corrigieron el falso “Network Error” en redes celulares, el bloqueo de puertos SMTP en Railway, las respuestas 204 sin contenido, las restricciones de popups mediante COOP y la incompatibilidad de Expo Go con la autenticación nativa. Estas situaciones hicieron necesario adoptar interceptores defensivos, Brevo por HTTPS, respuestas JSON explícitas, cabeceras de aislamiento, Expo Development Build y validaciones transaccionales.

7. RECOMENDACIONES

- **Módulo de Telemedicina:** Ante la creciente demanda de atención remota, el sistema podría expandirse para incluir videoconsultas integradas directamente en la plataforma web y móvil.
- **Análisis de Datos e Inteligencia de Negocios:** Se sugiere la creación de un dashboard gerencial que utilice los datos almacenados para generar estadísticas sobre las enfermedades más comunes, los tiempos de atención y el rendimiento del personal médico, facilitando la toma de decisiones institucionales.
- **Auditoría Avanzada:** Implementar tablas de logs de eventos (event_log) o triggers en la base de datos para mantener un historial detallado de todas las modificaciones realizadas en los registros clínicos, fortaleciendo aún más la seguridad y el cumplimiento normativo.

8. REFERENCIAS BIBLIOGRÁFICAS

Congreso de Colombia. (2015). LEY ESTATUTARIA 1751 DE 2015. Recuperado de secretariassenado:

http://www.secretariassenado.gov.co/senado/basedoc/ley_1751_2015.html

JWT.io. (s.f.). JSON Web Token Introduction. Recuperado de JSON Web Tokens: <https://jwt.io/introduction>

Naciones Unidas. (2015). Objetivo 3: Garantizar una vida sana y promover el bienestar para todos en todas las edades. Recuperado de www.un.org: <https://www.un.org/sustainabledevelopment/es/health/>

Naciones Unidas. (2015). Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación. Recuperado de www.un.org: <https://www.un.org/sustainabledevelopment/es/infrastructure/>

Ñaupari Vergara, M. P. (2023). Desarrollo de aplicación móvil para gestión de citas odontológicas. Recuperado de bibdigital.epn.edu.ec: <https://bibdigital.epn.edu.ec/bitstream/15000/24973/1/CD%2013734.pdf>

Paladines Morán, K. A. (2022). Sistema web para la administración de citas médicas en el área de bienestar estudiantil de la UNESUM. Recuperado de repositorio.unesum.edu.ec: <http://repositorio.unesum.edu.ec/handle/53000/3550>

Sisa Sisa, Á. I. (2022). Gestión de inventarios mediante PWA en entorno hospitalario. Recuperado de dspace.unach.edu.ec:

<http://dspace.unach.edu.ec/bitstream/51000/9017/1/Informe%20Final%20del%20Proyecto%20de%20Investigaci%3%b3n.pdf>

Zambrano Rosero, A. A. (2023). Sistema automático basado en tecnologías inalámbricas para gestión de pacientes. Recuperado de red.uao.edu.co:

<https://red.uao.edu.co/server/api/core/bitstreams/fc0f8a96-7976-4b8c-a46a-fac23d94fc4d/content>

9. APÉNDICES

No se agregan apéndices al trabajo de grado.

10. ANEXOS

Anexo A. Diagrama entidad-relación de la base de datos del sistema, correspondiente al modelo relacional normalizado hasta la Tercera Forma Normal (3FN). Este diagrama se presenta en el cuerpo del documento en la sección 4.1 Diseño de la Base de Datos.

Anexo B. Diagrama de arquitectura general del sistema implementado, enviado también por correo electrónico junto con los demás diagramas de soporte. Corresponde a la Figura 2 del presente documento, en la sección 5. Resultados.

Anexo C. Diagrama de flujo funcional del módulo de agendamiento de citas, enviado por correo electrónico. Corresponde a la Figura 3 del presente documento, en la sección 5. Resultados.

Anexo D. Diagrama de flujo de despliegue de los componentes del sistema, enviado por correo electrónico. Corresponde a la Figura 4 del presente documento, en la sección 5. Resultados.