

GESTANTEAPP, APLICACIÓN MÓVIL PARA EL MANEJO DE DATOS DE GESTANTES: APOYO EN TELEMEDICINA

Jose Miguel Henao-Gómez¹ John Andersson Valencia² Juan José García-Londoño³ Jhon Fredy Narváez-Valderrama⁴ Edison Andrés Tavera⁵ Porfirio Alvarez-Arango⁶

¹, Joven investigador, Corporación Universitaria Remington, jose.henao.6236@miremington.edu.co.

²Ingeniero industrial, Msc Ingeniería industrial, Corporación Universitaria Remington, Medellín-Colombia; john.valencia@uniremington.edu.co

³Ingeniero civil, Corporación Universitaria Remington, Medellín-Colombia; juan.garcia.3750@miremington.edu.co

⁴Químico Farmacéutico, PhD Ingeniería, Corporación Universitaria Remington, Medellín-Colombia; jhon.narvaez@uniremington.edu.co

⁵Médico general, Especialista en Ginecología y obstetricia Corporación Universitaria Remington, edison.tavera@uniremington.edu.co

⁶Ingeniero en sistemas, Corporación Universitaria Remington,

Autor de correspondencia: palvarez@uniremington.edu.co

RESUMEN

La gestión de datos en estudios clínicos de cohorte representa grandes retos en custodia y suministro periódico de información por parte de participantes. Por ejemplo, la información captada de mujeres gestantes en Antioquia supone un desafío asociado con limitaciones de comunicación entre regiones, pero también una oportunidad para acceder a datos y comunicación en tiempo real entre paciente - investigador, lo que apoyaría la salud preventiva. Así, el desarrollo de una aplicación móvil usando Dart y Flutter para la interfaz y Python con Flask para el backend, se orienta al monitoreo de síntomas y la captación de información gestacional, como insumo durante los controles prenatales en especialidades gineco-obstetricas en el futuro. El desarrollo, aplicó principios de Scrum integrando contenidos científicos como autenticación segura, registro de síntomas, historial y notificaciones. En conclusión, la aplicación denominada GESTANTEAPP, permitirá el seguimiento del bienestar materno-fetal en Antioquia basado en principios de telemedicina.

Palabras clave: mHealth; Aplicaciones Móviles; Dart, Flutter; Python; Flask; Gestación.

Recibido: 2 de noviembre de 2025. Aceptado: 13 de mayo de 2026

Received: November 2, 2025. Accepted: May 13, 2026

GESTANTEAPP, MOBILE APPLICATION FOR THE MONITORING OF GESTATIONAL SYMPTOMS

ABSTRACT

Data management in research projects about cohort study represents significant challenges due to the custody and the difficulties in periodic data of information from participants. Thereby, the collected information from pregnant women in the department of Antioquia is a challenge due to limitations of communication across regions, but also an opportunity to access real-time data, support preventive health, and link the continuous contact between researchers and patients. This research paper shows the development of a mobile application built with Dart and Flutter for the interface and Python with Flask for the backend, aimed at monitoring symptoms and capturing information during pregnancy. The development applied Scrum principles and incorporated scientific content validated by gynecologist and experts. The results include secure authentication, symptom logging, history visualization, and notifications. It is concluded that GESTANTEAPP will allow the continuous monitoring and strengthens maternal-fetal well-being in different regions located in Antioquia based on telemedicine principles.

Keywords: *mHealth; Mobile Applications; Dart; Flutter; Python; Flask; Pregnancy.*

Cómo citar este artículo: J. Henao-Gómez, J. Valencia, J. García-Londoño, J. Narváez, E. Tavera, P. Alvarez-Arango. "Gestanteapp, aplicación móvil para el manejo de datos de gestantes: apoyo en telemedicina", Revista Politécnica, vol.22, no.43 pp.32-49, 2026. DOI:10.33571/rpolitec.v22n43a3

1. INTRODUCCIÓN

La gestación constituye uno de los procesos biológicos y sociales más trascendentales en la vida de la mujer, al implicar transformaciones fisiológicas, emocionales y sociales que demandan un acompañamiento integral y continuo [1]. A nivel global, la salud materna se mantiene como un indicador esencial del desarrollo humano y la equidad en los sistemas sanitarios [1,2]. La Organización Mundial de la Salud estima que diariamente mueren alrededor de 800 mujeres por causas prevenibles vinculadas con el embarazo y el parto, entre las cuales destacan las hemorragias graves, la hipertensión gestacional, las infecciones y los partos obstruidos [3]. Estas muertes se concentran principalmente en países de ingresos bajos y medianos, donde la deficiencia en el acceso a servicios de salud de calidad, la escasez de personal especializado y las desigualdades sociales incrementan el riesgo materno.

En este sentido, la Organización Panamericana de la Salud (OPS) enfatiza que fortalecer la atención prenatal, garantizar personal calificado durante el parto y asegurar el seguimiento posnatal son medidas efectivas para reducir significativamente la mortalidad materna [4]. La evidencia científica respalda que la atención continua y centrada en la gestante mejora el bienestar materno-fetal, disminuye la incidencia de complicaciones y optimiza la detección temprana de riesgos obstétricos [5,6]. Por lo tanto, el acompañamiento con ayudas tecnológicas facilita una interacción ágil con pacientes desde regiones de alta complejidad en accesibilidad.

La telemedicina se ha convertido en una herramienta clave para ampliar el acceso a servicios de salud materna, especialmente en regiones con barreras geográficas o limitaciones en la disponibilidad de profesionales. La OPS, define la telemedicina como la prestación de servicios clínicos a distancia mediante tecnologías digitales, permitiendo diagnósticos, tratamiento y seguimiento remoto [6]. En Colombia, la Ley 1419 de 2010 y el Plan Decenal de Salud Pública 2022–2031 promueven su incorporación en la atención materno-infantil, reconociendo su potencial para fortalecer la continuidad del cuidado [7]. Programas implementados por instituciones clínicas han demostrado que el monitoreo a distancia y las consultas virtuales pueden mejorar la detección temprana de riesgos como preeclampsia, diabetes gestacional o parto prematuro [8].

De manera complementaria, las aplicaciones móviles de salud (mHealth) se han posicionado como herramientas útiles para el acompañamiento durante el embarazo. Estas aplicaciones permiten registrar síntomas, llevar control de la evolución del embarazo, recordar citas prenatales y ofrecer información confiable para el autocuidado [9]. Apps ampliamente utilizadas, como Dana, Embarazo+, BabyCenter y Sproud, facilitan el seguimiento del desarrollo fetal y brindan orientación educativa [10]. Sin embargo, diversos autores señalan que muchas de estas plataformas se centran en contenidos informativos y no incluyen funcionalidades avanzadas como el registro estructurado de síntomas, la evaluación del estado emocional o la integración con sistemas clínicos, lo que limita su alcance en la atención materna [11].

Esta brecha tecnológica abre la oportunidad para desarrollar soluciones móviles más completas que permitan documentar de manera integral la experiencia gestacional, identificar tempranamente signos de alarma y, cuando corresponda, facilitar el enlace con profesionales de la salud. De esta manera, las aplicaciones móviles pueden convertirse en un recurso esencial para reforzar la seguridad materna, mejorar el autocuidado y promover un acompañamiento continuo a lo largo del embarazo [12]. Adicionalmente, la interacción con maternidades permite el acceso a muestreos importantes para monitorear el estado de salud de las pacientes.

2. METODOLOGIA

2.1 Enfoque metodológico: Buenas prácticas basadas en la Metodología Ágil Scrum

Para el desarrollo de la aplicación se incorporaron elementos de la metodología ágil Scrum, un marco de trabajo ampliamente utilizado en proyectos de software que requieren flexibilidad, adaptación continua y entregas incrementales. Aunque la metodología Scrum, no se implementó de forma estricta, se aplicaron componentes esenciales como: el trabajo por iteraciones cortas (sprint), la priorización continua y la revisión frecuente del avance (Product Backlog), lo que permitió integrar mejoras de manera progresiva a partir de la retroalimentación por el equipo de trabajo (Incluyendo áreas ambientales, médicas, desarrolladores) [13]

La elección de estos componentes se fundamentó en la naturaleza dinámica del proyecto, dado que el desarrollo de una aplicación dirigida a mujeres gestantes exige ajustes continuos tanto en funcionalidades como la interfaz según las necesidades detectadas por las usuarias y el personal médico. Este enfoque permitió generar versiones funcionales parciales en ciclos de trabajo breves, favoreciendo la validación continua de la utilidad y usabilidad de lo implementado sin esperar a la culminación total del desarrollo [14].

Aunque la construcción de aplicaciones es un proceso continuo debido a su adaptación a los nuevos requerimientos, en una etapa inicial se construyeron funcionalidades esenciales, como el registro e inicio de sesión, con el fin de una base sólida de autenticación y seguridad. A medida que se recolectaron retroalimentaciones preliminares derivados de las pruebas internas, se integraron nuevas características como el registro diario de síntomas, la visualización del historial, las notificaciones personalizadas y las configuraciones del perfil.

Este enfoque incremental permitió evaluar, mejorar y estabilizar cada módulo antes de avanzar al siguiente, asegurando la estabilidad del sistema y evitando la acumulación de errores. Asimismo, facilitó la incorporación temprana de mejoras clínicas sugeridas por el personal experto en ginecología y obstetricia fortaleciendo tanto la pertinencia funcional como la experiencia del usuario. En coherencia con los principios tomados de Scrum, cada ciclo de trabajo permitió entregar una versión parcial, evaluarla y refinada de acuerdo con las necesidades reales detectadas [13,14].

2.2 Flujo general de la aplicación



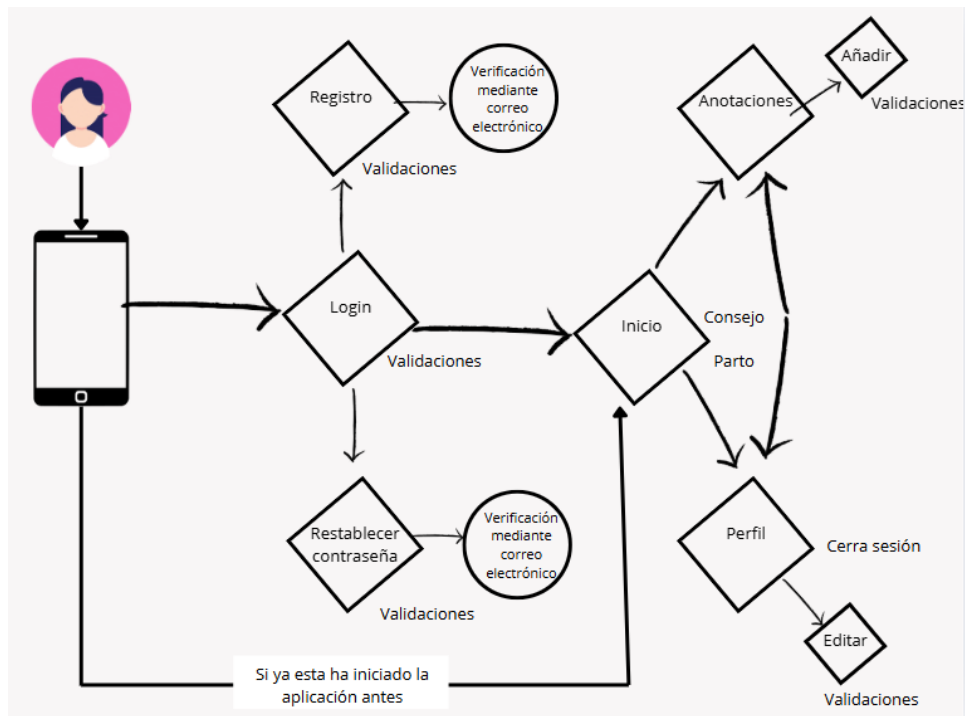


Figura 1. Flujo general de la aplicación (Fuente: Elaboración propia)

La Figura 1 muestra el flujo de funcionamiento general de la aplicación móvil desarrollada para el registro de síntomas durante el embarazo recolectadas por la aplicación GESTANTEAPP. El proceso inicia cuando la usuaria accede a la pantalla de inicio de sesión (login), donde se presentan tres opciones principales: ingresar con usuario y contraseña, registrarse o restablecer la contraseña.

En caso de seleccionar Registrarse, la aplicación redirige a una pantalla con los campos de registro personal y clínico, los cuales son validados antes de su envío al sistema. Como medida de seguridad, se incluyó un proceso de verificación de identidad mediante correo electrónico, en el que la usuaria ingresa un código de confirmación enviado a su dirección antes de completar el registro. Una vez finalizado el proceso, el sistema retorna a la pantalla de inicio de sesión.

Si la usuaria elige la opción Restablecer contraseña, se muestra una pantalla donde se solicita el correo electrónico asociado a la cuenta. Tras validar la existencia del usuario, el sistema envía un código temporal de recuperación que debe ser introducido para verificar la identidad. Una vez confirmado, se habilita la opción para establecer una nueva contraseña, tras lo cual la aplicación redirige nuevamente al inicio de sesión.

Al ingresar correctamente las credenciales, la usuaria accede a la pantalla principal, la cual presenta un consejo general relacionado con el embarazo (Consejos suministrados por gineco-obstetricia) y un botón destinado a notificar el parto al equipo médico. Estos consejos son generales con orientación hacia la visita médica en caso de síntomas mayores, lo cual evita enmascarar problemas serios dentro de la gestación. Esta pantalla actúa como punto de partida para la navegación del sistema, que se organiza mediante una barra inferior de navegación con acceso a las secciones Mis Anotaciones y Perfil.

En la sección Mis Anotaciones, la usuaria puede visualizar las entradas previas de síntomas registradas, así como acceder a la función de agregar una nueva anotación en su diario personal. Finalmente, en la sección Perfil, se muestran los datos personales y clínicos registrados, permitiendo su actualización o modificación según sea necesario.

2.3 Búsqueda y análisis de aplicaciones existentes

Se realizó un análisis de aplicaciones móviles centradas en el embarazo, disponibles en tiendas digitales (Google Play y App Store). Este estudio permitió identificar de manera sistemática las fortalezas y limitaciones de herramientas como Dana, Embarazo+, BabyCenter y Sproud, sirviendo como punto de referencia para establecer las funcionalidades distintivas de la aplicación propuesta.

A partir de este análisis, se definieron las características diferenciadoras de la aplicación, resaltando la integración del registro físico y emocional de síntomas, el seguimiento personalizado y la comunicación con el personal médico.

2.4 Selección del lenguaje de programación

En la etapa de selección tecnológica se realizó una búsqueda comparativa de lenguajes y frameworks de programación para aplicaciones móviles. El objetivo fue identificar la opción más adecuada en términos de rendimiento, escalabilidad, facilidad de desarrollo, soporte comunitario y compatibilidad con los requerimientos del proyecto.

- Kotlin. Lenguaje moderno desarrollado por JetBrains en 2011 y adoptado oficialmente por Google en 2017 como el lenguaje principal para el desarrollo de aplicaciones Android.
- Java. Lenguaje orientado a objetos y multiplataforma creado en 1995, base del ecosistema Android desde 2008.
- Swift. Lenguaje oficial de Apple para iOS y macOS desde 2014.
- React Native. Framework de código abierto creado por Meta en 2015 para desarrollo multiplataforma con JavaScript.
- Flutter. Framework de Google lanzado en 2017 que permite desarrollar aplicaciones nativas para Android y iOS con un único código base.

Tras analizar las diferentes alternativas, se eligió Flutter como herramienta de desarrollo por su capacidad de generar aplicaciones móviles nativas tanto para Android como para iOS desde un único código base [15].

2.5 Identificación de herramientas de desarrollo

Para la implementación del sistema se definieron las siguientes herramientas tecnológicas:

- Frontend: Flutter para la interfaz gráfica (15).
- Backend: Flask como framework ligero para la construcción de servicios API [16,17]
- Base de datos: PostgreSQL para el almacenamiento estructurado de la información [18,19].
- Seguridad: Implementación de JWT (JSON Web Tokens) bajo estándares internacionales como HIPAA y GDPR [20,21].
- Notificaciones: Firebase Cloud Messaging (FCM) para envío de mensajes y notificaciones push [22].

2.6 Diseño de la interfaz y páginas de la aplicación

El diseño de la interfaz de la aplicación se desarrolló siguiendo principios del diseño centrado en la usuaria, para ofrecer una experiencia visual intuitiva, accesible y empática hacia las mujeres gestantes. Se elaboraron prototipos de cada una de las pantallas que conforman la aplicación, garantizando coherencia estética y simplicidad en la navegación.



Estas decisiones se sustentan en los principios de usabilidad y experiencia de usuario descritos en la literatura [23,24]. De acuerdo con los lineamientos establecidos [25], el diseño de interfaces eficaces se fundamenta en la consistencia, jerarquía visual y retroalimentación inmediata. Asimismo, se resalta que el uso de colores suaves, el lenguaje comprensible y los elementos visuales amigables fortalecen la experiencia emocional del usuario [26]. Finalmente, se consideraron las recomendaciones de accesibilidad del W3C [27].

2.7 Integración y arquitectura de datos

Se diseñó la arquitectura de datos utilizando PostgreSQL como sistema de gestión relacional, debido a su robustez y escalabilidad, fundamentales para garantizar la integridad de los registros clínicos. La estructura de la base de datos se organizó en tablas normalizadas que almacenan información de las usuarias, síntomas registrados, fechas y configuraciones personalizadas, lo que optimiza el rendimiento del sistema [18]. La comunicación entre la aplicación móvil y el servidor se estableció mediante servicios API RESTful desarrollados con Flask, permitiendo una transferencia ligera y estructurada de datos en formato JSON [16].

2.8 Seguridad y protección de datos

En materia de seguridad, cada solicitud enviada desde la aplicación incluirá un token de autenticación JWT, generado durante el inicio de sesión y validado en el servidor antes de conceder acceso a los recursos protegidos. Este proceso garantizará que solo las usuarias autenticadas puedan acceder a su información personal o realizar modificaciones [28].

2.9 Pruebas y validación

Se llevaron a cabo pruebas piloto con un grupo reducido de usuarios (Para analizar descarga de aplicación), con el propósito de evaluar la usabilidad general de la aplicación, la precisión en el registro de síntomas y la efectividad del sistema de notificaciones. Para garantizar la continuidad del servicio en contextos de conectividad limitada, se implementó el uso del token de sesión JWT como mecanismo temporal de almacenamiento local lo que permite mantener ciertas funcionalidades operativas incluso sin conexión estable [28]. Adicionalmente, se realizó una validación clínica con un ginecólogo profesional con el fin de estimar la pertinencia clínica de las funcionalidades desarrolladas. Las mejoras resultantes fortalecieron la confiabilidad del sistema, incrementando su nivel de preparación previa antes de su implementación a mayor escala.

Por otro lado, se hicieron pruebas de rendimiento de la API por medio de los métodos de POST y GET para entender los tiempos de respuesta que tendrá el servidor usado a una cantidad de consultas establecidas bajo determinadas condiciones.

3. RESULTADOS

3.1. Descripción general del sistema

La aplicación desarrollada corresponde a una herramienta móvil para el monitoreo personalizado de síntomas durante el embarazo, orientada a brindar acompañamiento digital a mujeres gestantes a través de un entorno accesible, seguro y fácil de usar. Su propósito es permitir que las usuarias registren sus síntomas diarios, consulten su historial y reciban notificaciones personalizadas que fortalezcan el autocuidado y la adherencia a los controles prenatales.

3.1.1 Módulos desarrollados



El sistema se organizó en diversos módulos funcionales que aseguran una interacción fluida entre la usuaria y la aplicación, permitiendo un seguimiento completo de la salud materna:

- Registro e inicio de sesión: módulo responsable de la autenticación segura de las usuarias mediante credenciales personales y verificación por correo electrónico. Se implementó un sistema de validación de identidad basado en tokens JWT, asegurando la protección de datos y la gestión de sesiones activas.
- Registro diario de síntomas: permite registrar información sobre el estado físico y emocional de la gestante (como dolor abdominal, estado de ánimo, presión arterial o movimientos fetales). Los datos se guardan con marca temporal y se almacenan de forma estructurada en la base de datos.
- Visualización del historial: proporciona una interfaz para consultar cronológicamente los registros previos de síntomas, facilitando la identificación de variaciones o tendencias a lo largo del tiempo.
- Perfil y configuración: permite modificar datos personales, preferencias de notificación y ajustes de la cuenta.
- Mapa: incorpora el servicio de Google Maps SDK for Android para permitir la visualización de un mapa de Google Maps y la posibilidad de geolocalizar en el momento exacto el lugar o buscar una dirección en específico
- Notificaciones: incorpora el servicio Firebase Cloud Messaging (FCM) para enviar alertas y recordatorios automáticos sobre citas, controles o recomendaciones de salud [22].

3.1.2 Pruebas de funcionamiento

La aplicación fue probada en dispositivos Android, validando el correcto funcionamiento de los módulos desarrollados, la conexión estable con el servidor y la persistencia de datos. El desarrollo de la aplicación fue realizado bajo un entorno Android debido a que en Colombia más del 70% de los dispositivos móviles funcionan en este entorno [Referenciar]. No obstante, el uso del framework Flutter asegura la portabilidad multiplataforma, permitiendo su adaptación a sistemas iOS con mínimos ajustes en la configuración y el código fuente [15].

3.1.3 Infraestructura del servidor y alojamiento de la API

La API del sistema se alojó en las instalaciones de la Corporación Universitaria Remington, en un servidor dedicado configurado con el sistema operativo Ubuntu Server, un entorno estable, seguro y optimizado para la ejecución de servicios web [29].

El servidor opera bajo una arquitectura cliente-servidor, donde la aplicación móvil (cliente) se comunica con una API RESTful desarrollada en Flask, encargada de procesar solicitudes, gestionar la lógica de negocio y administrar la iteración con la base de datos [30].

La seguridad en la comunicación entre cliente y servidor se asegura mediante OpenSSL y el protocolo TLS (Transport Layer Security), que cifran los datos transmitidos, garantizando su integridad y confidencialidad. Adicionalmente, el servidor está configurado con un cortafuegos (UFW) que restringe el acceso únicamente a los puertos y servicios esenciales.

3.1.4 Gestión y almacenamiento de datos

El sistema utiliza PostgreSQL como Sistema de Gestión de Bases de Datos Relacional (RDBMS), seleccionado por su estabilidad, escalabilidad y cumplimiento con las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) [27].

La base de datos se gestiona mediante SQLAlchemy y su extensión Flask-SQLAlchemy, la cual proporcionan un mapeo objeto-relacional (ORM) para facilitar la interacción entre el código Python y las tablas del sistema.

El modelo lógico contempla la existencia de las siguientes entidades principales:



- Usuario: almacena información básica de las gestantes (nombre, correo, edad, contraseña cifrada).
- Síntoma: registra los datos físicos o emocionales ingresados por la usuaria, con fecha y hora.
- Historial: agrupa los registros diarios permitiendo su organización y consulta estructura posterior.
- Notificación: contiene la programación y estado de los mensajes enviados a la usuaria.

Cada registro se asocia a un identificador único, y las relaciones entre tablas (uno a muchos) permiten mantener la integridad referencial. Gracias al uso del control de concurrencia multiversión (MVCC) de PostgreSQL, el sistema puede atender múltiples solicitudes simultáneamente sin comprometer la consistencia ni la integridad de los datos.

3.2. Resultados funcionales

En esta sección se presentan los resultados funcionales obtenidos tras la implementación y despliegue de la aplicación móvil, desarrollada bajo un enfoque centrado en la usuaria y orientada a garantizar simplicidad visual, accesibilidad y una interacción intuitiva en cada módulo. La aplicación se encuentra completamente operativa en su entorno Android y ha sido probada en diferentes dispositivos de esta plataforma, asegurando consistencia en su comportamiento. Todas las pantallas del sistema comparten un tema visual unificado diseñado específicamente para la aplicación, lo que proporciona coherencia estética y facilita la navegación a lo largo de los distintos componentes.

El funcionamiento interno de la aplicación se sostiene mediante una arquitectura cliente–servidor en la que la API procesa las solicitudes recibidas desde las interfaces móviles, gestiona la lógica de negocio y se encarga de validar, registrar o consultar la información necesaria. La base de datos almacena de manera estructurada todos los datos de las usuarias, sus anotaciones, síntomas, direcciones y procesos asociados, garantizando integridad y disponibilidad.

A continuación, se describen las pantallas, flujos de navegación y procesos funcionales clave del sistema, así como su interacción con la API y la base de datos, apoyados en los diagramas que ilustran el comportamiento general de la aplicación. Ver figura 2

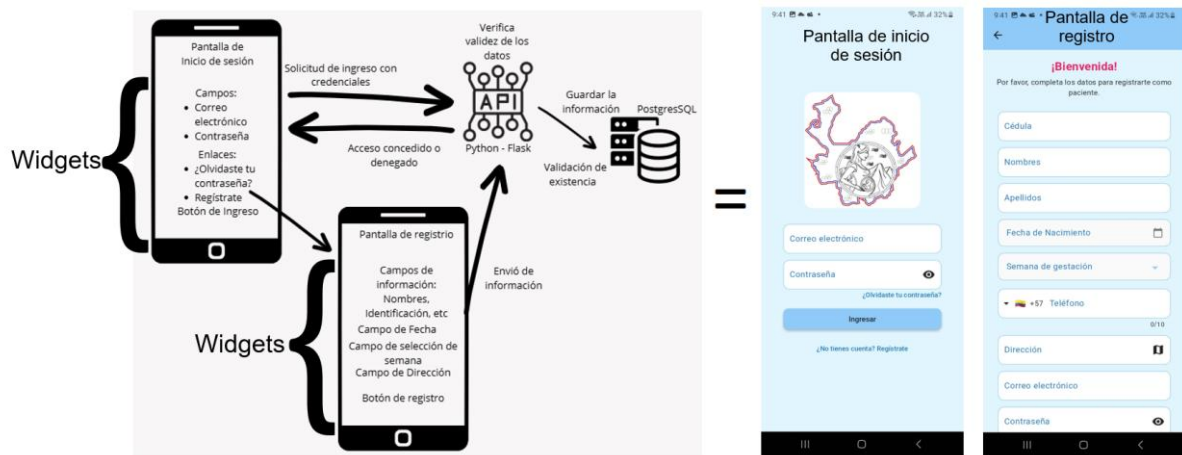


Figura 2. Flujo de operación en Registro e Inicio de sesión (Fuente: Elaboración propia)

La figura 2 presenta el proceso general que siguen los usuarios al interactuar con las pantallas iniciales de la aplicación. La pantalla de inicio de sesión incluye los widgets principales para este proceso: campos de texto para ingresar el correo electrónico y la contraseña, el botón para enviar las credenciales, y los enlaces que permiten acceder tanto a la pantalla de registro como a la opción de restablecer contraseña.

En la pantalla de registro, los usuarios completan los campos necesarios para la creación de su cuenta mediante los widgets correspondientes, entre ellos el campo de Dirección que redirige a una pantalla adicional descrita más adelante. Esta pantalla incorpora también el botón de registro, el cual inicia un proceso de validación mediante un código enviado al correo electrónico ingresado; este mecanismo se explicará con detalle más adelante, ya que también es utilizado en el flujo de recuperación de contraseña. Una vez validado el código, la API almacena los datos en la base de datos y finaliza el registro.

En cuanto al proceso de inicio de sesión, la usuaria utiliza los campos de texto para ingresar sus credenciales y el botón correspondiente para enviarlas a la API, que verifica su autenticidad y otorga o rechaza el acceso según corresponda. Además, la API proporciona el acceso a la aplicación mediante un token seguro que integra información relevante del usuario, incluyendo datos personales y configuraciones básicas, y permite cargar elementos iniciales como el registro de síntomas y otros datos vinculados al perfil de la gestante. Ver figura 3

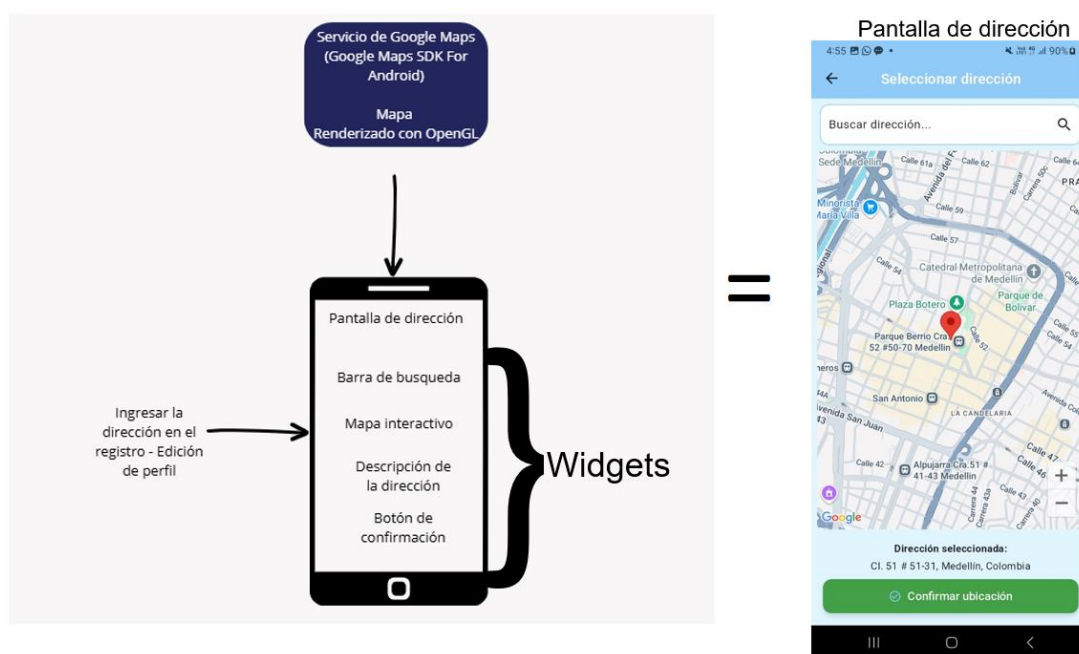


Figura 3. Flujo de operación de pantalla de dirección (Fuente: Elaboración propia)

La figura 3 muestra el flujo correspondiente a la pantalla de Dirección, a la cual se accede desde la pantalla de registro mediante el campo destinado a capturar la ubicación de la usuaria. Esta pantalla integra el servicio de Google Maps a través de Maps SDK for Android, permitiendo seleccionar una dirección de manera visual e interactiva.

La interfaz está compuesta por varios widgets que facilitan la búsqueda y confirmación de la ubicación: una barra de búsqueda que permite ingresar manualmente una dirección o punto de referencia, el mapa interactivo donde la usuaria puede desplazarse y seleccionar la ubicación exacta, un resumen en texto que muestra la dirección interpretada por el sistema y un botón de confirmación que permite registrar la información seleccionada.

Una vez confirmada la dirección, la aplicación regresa a la pantalla de registro para continuar con el proceso de creación de la cuenta. Posteriormente, cuando la usuaria confirma el registro, esta información será enviada a la API para su almacenamiento en la base de datos como parte del perfil registrado. Ver figura 4



Figura 4. Flujo de operación de Restablecer contraseña y Validación de códigos de verificación (Fuente: Elaboración propia)

La figura 4 presenta el flujo correspondiente al proceso de restablecimiento de contraseña, al cual se accede desde la pantalla de inicio de sesión mediante el enlace destinado para esta función. En esta primera pantalla, la usuaria encuentra un campo para ingresar el correo electrónico con el que se registró en el sistema y un botón para solicitar el código de verificación. La aplicación envía esta información a la API, que valida la existencia del correo en la base de datos y, si es correcto, genera y envía el código correspondiente.

En la pantalla de validación del código —mecanismo que también se utiliza en el proceso de registro— la usuaria ingresa el código recibido en su correo electrónico dentro del campo habilitado para ello. Este código posee una validez temporal limitada. Mediante el botón de verificación, la aplicación envía el dato a la API, que confirma su validez y permite continuar con el proceso.

Una vez verificado el código, se muestra una pantalla con dos campos para ingresar y confirmar la nueva contraseña, junto con un botón para finalizar el proceso. Tras superar las validaciones correspondientes, la API actualiza la contraseña en la base de datos y la aplicación redirige automáticamente a la pantalla de inicio de sesión. Ver figura 5



Figura 5. Flujo de navegación entre pantallas principales (Fuente: Elaboración propia)

La figura 5 presenta el flujo de navegación entre las pantallas principales de la aplicación una vez la usuaria ha iniciado sesión. Al autenticarse correctamente, el sistema dirige a la usuaria a la pantalla de Inicio. En esta vista, en la parte superior se encuentra un cuadro que muestra el “Consejo del día”, el cual es solicitado a la

API y actualizado dinámicamente. En la parte inferior de la pantalla se incluye un botón destinado al registro de un evento de parto, cuya acción envía la información correspondiente a la API para su almacenamiento.

En la pantalla de *Mis anotaciones*, la aplicación solicita a la API el listado de anotaciones realizadas previamente por la usuaria. Estas se muestran de manera organizada por fecha, cada una dentro de un cuadro donde se detallan los síntomas registrados. En la parte inferior derecha se incorpora un botón con el símbolo “+”, el cual permite acceder a la pantalla de Registro de síntoma explicada más adelante. En la parte inferior se incluye un semáforo de síntomas, un widget que clasifica la condición reportada según su nivel de urgencia y ofrece recomendaciones sobre la acción adecuada, como consulta prioritaria o atención inmediata.

Por su parte, la pantalla de *Mi Información* obtiene desde la API los datos personales de la usuaria y los presenta en campos dedicados para facilitar su lectura. Entre los elementos disponibles se encuentran varios botones de acción, entre ellos Registrar acompañante, Cerrar sesión y el botón de Editar información, este último representado con el ícono de un lápiz. Este ícono, ubicado en la esquina inferior derecha de la pantalla, permite acceder a la interfaz de edición de datos personales, la cual se describe más adelante.

Un elemento común presente en todas estas pantallas es la barra de navegación inferior, que actúa como componente central de desplazamiento y permite a la usuaria cambiar entre las secciones principales de manera sencilla y directa. Ver figura 6

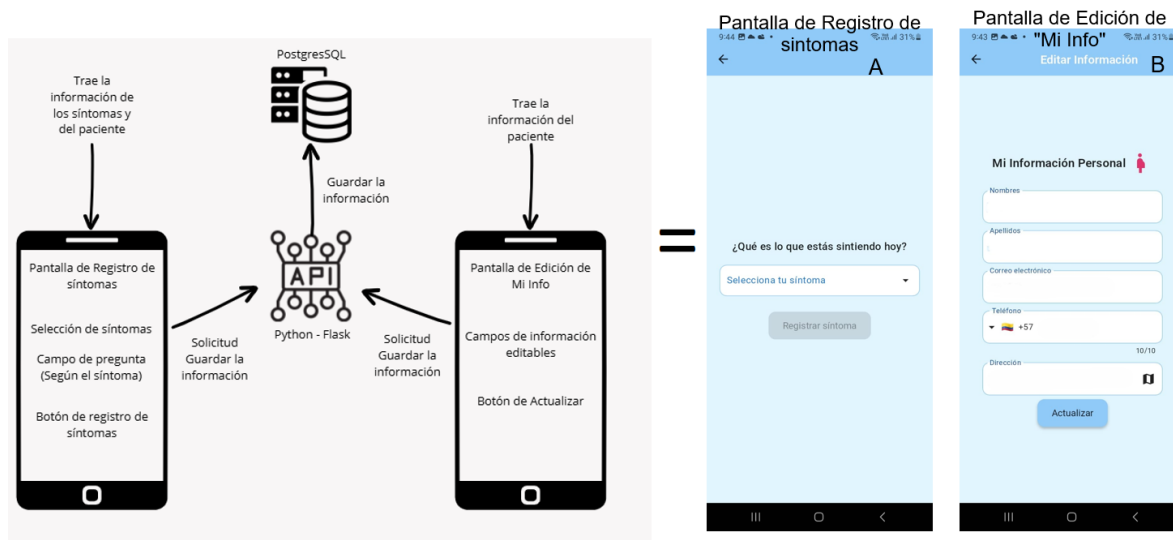


Figura 6. Flujo de procesos de registro de síntoma y actualización de perfil (Fuente: Elaboración propia)

La figura 6 presenta el flujo correspondiente a la pantalla de *Registro de síntoma*, a la cual se accede desde la pantalla de *Mis anotaciones*. Esta pantalla permite registrar un nuevo síntoma utilizando los datos de referencia ya manejados por la aplicación, los cuales son cargados desde el inicio de sesión a través de la API, que incluye en el token seguro la información del usuario y el conjunto de síntomas disponibles. La interfaz incorpora un selector de síntomas con función de búsqueda para facilitar la selección del tipo de síntoma. Dependiendo de la opción elegida, se despliega una pregunta complementaria con respuesta tipo Sí o No. Finalmente, mediante el botón de registro, la usuaria envía la información a la API, que almacena el nuevo síntoma en la base de datos y devuelve el control a la pantalla de *Mis anotaciones*, donde la nueva anotación aparece reflejada.

Asimismo, la figura 6 incluye el flujo de la pantalla de Edición de información, a la cual se accede desde la sección *Mi Información*. Esta pantalla carga los datos actuales del perfil del paciente y permite modificar únicamente ciertos campos autorizados. Tras realizar los cambios, la usuaria utiliza el botón de actualización para enviar la información modificada a la API. Una vez validados y almacenados los cambios en la base de datos, la aplicación redirige nuevamente a la pantalla de *Mi Información*, mostrando los datos actualizados. Ver figura 7

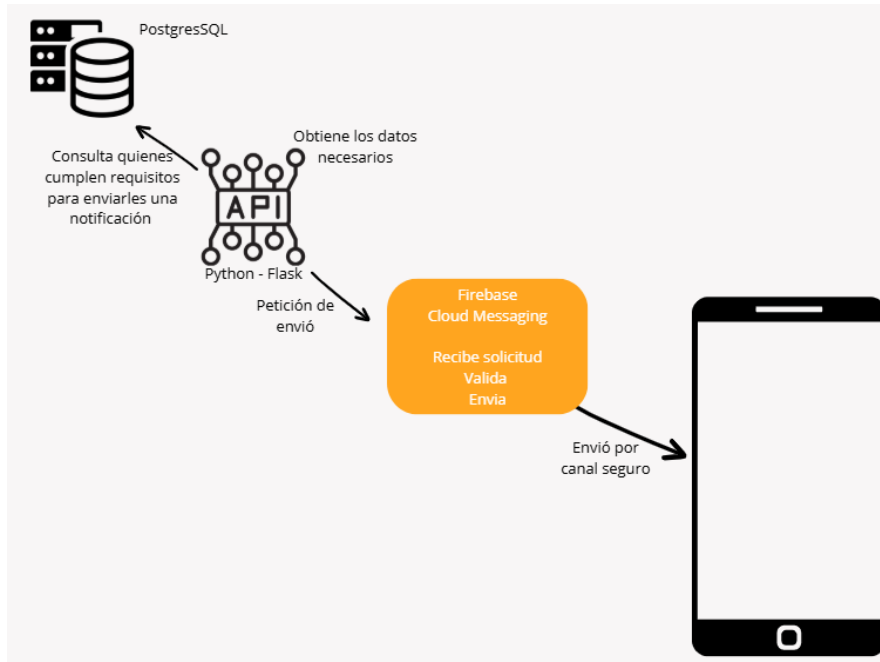


Figura 7. Flujo de proceso de notificaciones (Fuente: Elaboración propia)

La figura 7 muestra el proceso seguido por el sistema para el envío automático de notificaciones a las usuarias. En primer lugar, la API consulta la base de datos para identificar a las usuarias que cumplen con las condiciones establecidas, como la ausencia de un registro de síntoma en un periodo determinado u otros criterios definidos.

Una vez obtenida la información, la API prepara los datos necesarios y envía la solicitud de notificación a un canal del servicio Firebase Cloud Messaging (FCM). Este servicio valida el mensaje y gestiona su distribución mediante un canal seguro, enviándolo directamente al dispositivo móvil de la usuaria.

3.3. Pruebas del sistema y funcionamiento

Con el propósito de evaluar el desempeño técnico del sistema, se realizaron pruebas de funcionamiento sobre los principales métodos implementados en la API desarrollada con Flask, alojada en el servidor institucional de la Corporación Universitaria Remington. Estas pruebas tuvieron como objetivo analizar la eficiencia en el procesamiento de solicitudes, la correcta integración entre la API y la base de datos PostgreSQL, y la validación del mecanismo de autenticación mediante JSON Web Tokens (JWT).

Durante la fase de validación se probaron los métodos POST y GET, correspondientes a las operaciones de registro de síntomas, consulta de consejo, consulta de anotaciones realizadas y consulta de síntomas. Los resultados obtenidos evidencian un tiempo de respuesta promedio entre 28 y 36 milisegundos (ms) por solicitud, manteniendo una estabilidad adecuada incluso bajo múltiples peticiones consecutivas. Este rango refleja un rendimiento óptimo para aplicaciones móviles conectadas a servicios RESTful, garantizando una experiencia fluida para las usuarias.

La siguiente tabla presenta un resumen general de los tiempos de respuesta registrados durante las pruebas técnicas:

Tabla 1. Resultados de las pruebas de rendimiento de la API (Fuente: Elaboración propia)

Método HTTP	Operación evaluada	Descripción general	Tiempo promedio (ms)	Estado de respuesta
GET	Consulta	Consulta de los síntomas registrados en la base de datos	36	200
GET	Consulta	Consulta de consejo para la madre gestante	32	200
GET	Consulta	Consulta de las anotaciones realizadas por una usuaria	28	200
POST	Registro	Registro de una anotación	33	200

En cuanto a la integración entre la API y la base de datos, las pruebas confirmaron que los datos se almacenan correctamente en las tablas correspondientes, garantizando la persistencia y consistencia de la información. Cada solicitud registrada en los logs del servidor Flask reflejó una respuesta exitosa, sin errores de conexión o pérdida de datos.

Asimismo, se comprobó el funcionamiento del sistema de autenticación JWT, verificando que los tokens se generaran correctamente al iniciar sesión y se validaran antes de acceder a recursos protegidos. Los tokens presentan un tiempo de expiración limitado, reforzando la seguridad del sistema y evitando accesos no autorizados. Finalmente, las pruebas de funcionamiento relacionadas con el número de descargas y el reporte de dificultades realizado con 30 usuarios indica una aceptación cercana al 90% y una evaluación preliminar de las razones asociadas con la no descarga. Ver figura 8

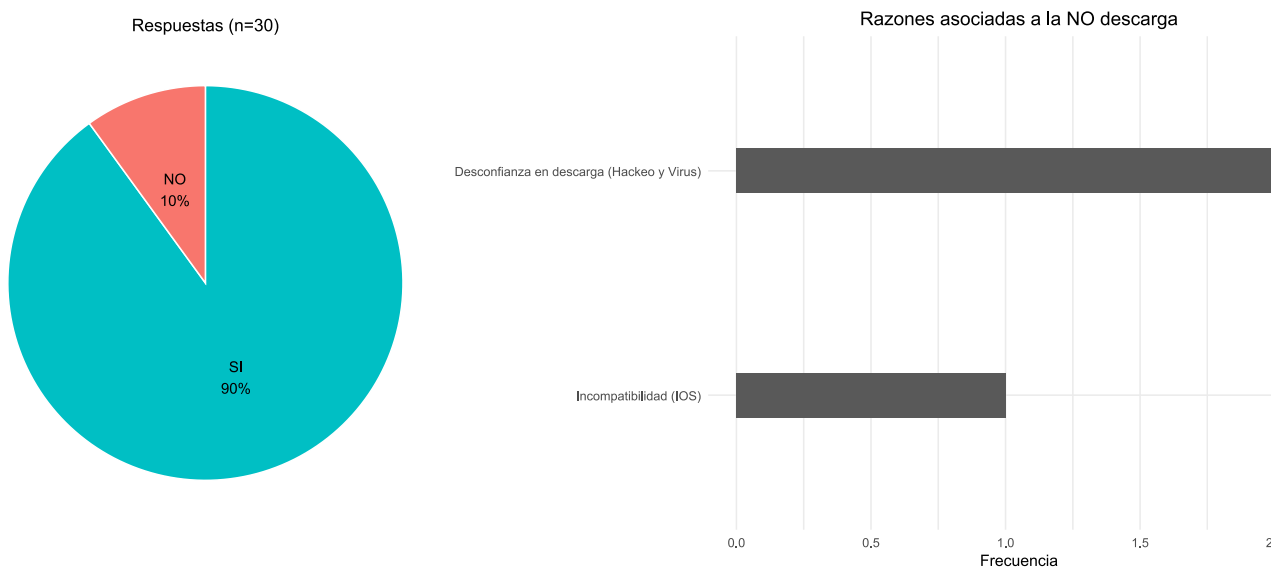


Figura 8. Análisis de funcionamiento y descarga de la aplicación. n=30

De acuerdo con la figura 8, la desconfianza es el primer factor que reduce el número de descargas en los dispositivos. Adicionalmente un porcentaje menor al 5% presenta dificultades en la instalación debido al desarrollo de la aplicación solo para entorno Android, lo que demuestra que el desarrollo en dicho entorno garantiza una población superior al 95% como fue presentado con anterioridad para la justificación del desarrollo inicial en Android para la aplicación GESTANTEAPP.



4. DISCUSIÓN

4.1. Evaluación del cumplimiento de objetivos

El desarrollo de la aplicación móvil para el monitoreo personalizado de síntomas durante el embarazo logró cumplir los objetivos planteados tanto en los aspectos funcionales como técnicos y de experiencia de usuaria.

Desde el punto de vista funcional, la aplicación facilita el registro y monitoreo diario de síntomas físicos y emocionales, la consulta del historial, la administración del perfil personal y el envío de notificaciones automáticas sobre cuidados prenatales. Cada módulo fue evaluado mediante pruebas de uso en dispositivos Android, asegurando estabilidad, navegabilidad y consistencia visual del sistema.

En el ámbito técnico, el uso combinado del framework Flutter para el frontend y Flask para la construcción de la API permitió un flujo de comunicación eficiente entre la aplicación móvil y el servidor. El tiempo promedio de respuesta, entre 28 y 36 milisegundos, evidencia un rendimiento estable y adecuado para entornos móviles. El uso de PostgreSQL como sistema gestor de base de datos y la implementación del esquema de autenticación mediante JSON Web Tokens (JWT) aseguraron la persistencia, consistencia y protección de los datos de las usuarias.

En términos de experiencia de usuaria, la interfaz se diseñó siguiendo principios de usabilidad y accesibilidad, garantizando una interacción intuitiva y empática para las gestantes. Los íconos, colores suaves y mensajes claros ayudaron a minimizar la carga cognitiva y a fortalecer la percepción de un acompañamiento constante.

A nivel comparativo, la aplicación presenta aportes diferenciadores concretos frente a otras soluciones existentes como Embarazo+ o BabyCenter (12). A diferencia de estas, que se centran principalmente en la entrega de contenido educativo, la aplicación desarrollada integra un registro estructurado de síntomas físicos y emocionales, una clasificación de riesgo tipo semáforo, y una sincronización segura con un servidor propio, optimizando la disponibilidad y el análisis de la información incluso en entornos de conectividad limitada.

4.2. Comparación con trabajos previos

A partir del análisis comparativo de aplicaciones móviles disponibles en tiendas digitales, como Dana, Embarazo+, BabyCenter y Sproud, se determinó que la mayoría de estas herramientas se centran principalmente en proporcionar contenido educativo o un seguimiento general del embarazo, sin incluir un registro sistemático de síntomas físicos y emocionales. En contraste, la aplicación desarrollada combina estas funcionalidades en un entorno digital único, facilitando un seguimiento personalizado, la documentación sistemática del bienestar materno y una interacción remota segura con profesionales de la salud, lo que constituye un avance significativo en la autogestión y el acompañamiento digital de las gestantes.

Los estudios previos revisados, como los de Lee y Moon [11] y Fonseca, Carpintero y Romero-Acosta [12], muestran que las soluciones existentes se concentran principalmente en brindar información general o guías sobre cambios fisiológicos durante el embarazo. En contraste, la propuesta presentada prioriza la autogestión activa del bienestar materno-fetal, mediante un sistema interactivo que integra el registro diario de síntomas, la visualización histórica de patrones y la emisión de alertas automáticas basadas en criterios clínicos predefinidos.

Adicionalmente, la adopción de una arquitectura cliente-servidor con almacenamiento en PostgreSQL y comunicación cifrada mediante TLS proporciona una ventaja técnica notable en comparación con aplicaciones que dependen únicamente del almacenamiento local o de servicios externos de terceros. Esta infraestructura permite escalar el sistema y, en versiones futuras, incorporar funcionalidades analíticas y predictivas alineadas con estándares internacionales de interoperabilidad sanitaria como HL7 FHIR y HIPAA. Por lo tanto, se podría realizar escalabilidad a largo plazo, permitiendo también una proyección desde las historias clínicas interoperables hasta su aplicabilidad en el sector salud.



Finalmente, en el contexto local, la aplicación constituye un aporte innovador dentro del ecosistema digital colombiano, al combinar estrategias de mHealth con un enfoque clínico y ambiental contextualizado en el Valle de Aburrá, donde los factores atmosféricos han sido asociados con variaciones en la salud materna (8). Esta contextualización fortalece el potencial de la herramienta para apoyar decisiones clínicas y promover el bienestar materno en escenarios locales de salud pública.

4.3. Análisis técnico

Desde la perspectiva técnica, la arquitectura del sistema demostró eficiencia, estabilidad y seguridad durante todo el proceso de pruebas.

El servidor institucional, configurado en Ubuntu Server, demostró estabilidad bajo una carga simultánea de peticiones, con un tiempo de respuesta promedio menor a 40 milisegundos. El empleo del protocolo TLS y el cifrado OpenSSL aseguraron la protección de los datos transmitidos entre cliente y servidor, mientras que la validación de tokens JWT reforzó los mecanismos de control de acceso y gestión de expiración de sesiones.

El sistema manejó la concurrencia de forma eficiente gracias al modelo MVCC (Multiversion Concurrency Control) de PostgreSQL, permitiendo procesar múltiples solicitudes sin pérdida de datos ni bloqueos. Además, los registros de logs del servidor Flask mostraron una tasa de éxito del 100 % en las transacciones realizadas durante las pruebas de integración.

En cuanto al rendimiento de la aplicación móvil, Flutter mostró un desempeño eficiente en dispositivos Android de gama media y alta, con un consumo de memoria reducido y tiempos de carga mínimos. El sistema funcionó adecuadamente en condiciones de conectividad limitada, manteniendo la sesión activa mediante el almacenamiento temporal del token JWT, garantizando así la continuidad del servicio.

Finalmente, se considera la posibilidad de expandir la compatibilidad del sistema a entornos iOS, aprovechando la naturaleza multiplataforma de Flutter, y de explorar la incorporación de sensores biomédicos (frecuencia cardíaca, presión arterial o temperatura corporal). Esta ampliación permitirá automatizar parte del registro de síntomas y consolidar un ecosistema integral de monitoreo materno-fetal, alineado con tendencias emergentes de salud digital. Además con el fin de reducir la desconfianza en la instalación de la aplicación para los usuarios, esta aplicación deberá alojarse en una plataforma oficial de distribución digital que permita una interacción entre las futuras usuarias gestantes y el desarrollo propuesto.

5. CONCLUSIONES

El desarrollo de la aplicación móvil para el monitoreo personalizado de síntomas durante el embarazo permitió demostrar la viabilidad técnica y funcional de una herramienta digital orientada al acompañamiento materno desde un enfoque preventivo y de autocuidado. La integración de Flutter y Flask, junto con la gestión de datos en PostgreSQL, garantizó un sistema estable, seguro y con tiempos de respuesta adecuados para su uso en entornos reales.

Desde el punto de vista funcional, la aplicación alcanzó los objetivos establecidos, integrando módulos para el registro de síntomas, consulta de historial, gestión del perfil de usuaria y envío de notificaciones automáticas, creando un entorno accesible y empático con las necesidades de las gestantes. La estructura modular permitió validar la interacción entre los componentes del sistema, destacando una usabilidad adecuada y coherencia visual.

A nivel técnico, las pruebas realizadas mostraron un rendimiento óptimo del servidor, con tiempos de respuesta entre 28 y 36 milisegundos y sin pérdida de datos en las operaciones de consulta o registro. Asimismo, el uso del protocolo TLS, la autenticación mediante JWT y la arquitectura cliente-servidor fortalecieron la seguridad de las comunicaciones y el control de acceso.

La aplicación presenta un avance significativo en el campo de la salud materna digital, al proporcionar un sistema capaz de registrar, analizar y comunicar información relevante sobre el bienestar físico y emocional de la gestante, incluso en contextos con conectividad limitada.



Como perspectivas futuras, se propone extender la aplicación a plataformas iOS y su integración con dispositivos biomédicos portátiles, como relojes inteligentes (smartwatch), para ampliar la captura de variables fisiológicas y facilitar la detección temprana de posibles complicaciones durante el embarazo. Esta evolución tecnológica busca adaptar la aplicación para operar directamente en dichos dispositivos, fortaleciendo el monitoreo continuo y la prevención de riesgos en tiempo real. Adicionalmente, la aplicación podría incorporar una función con respecto al acompañamiento con inteligencia artificial (IA) alimentado por un panel de expertos en el tema y ofrecer información valiosa a las usuarias. Esto fortalecería territorios con difícil acceso terrestre, donde las pacientes podrían tener un apoyo remoto durante controles prenatales debido a su ubicación geográfica compleja sin alejarse de los controles presenciales cuando los síntomas lo indiquen durante la gestación.

6. AGRADECIMIENTOS

Los autores agradecen al programa titulado: “Salud obstétrica-perinatal en Antioquia y su respuesta frente a la transformación de la calidad del aire” (Código 1365949110240), financiado por el Ministerio de Ciencia, Tecnología e Innovación (Minciencias). Convocatoria 949 – Misión Soberanía Sanitaria y Bienestar Social: Territorios Garantes de la Salud.

Para este proyecto fue fundamental el apoyo de la compañera Yileni Argoti Ospina, por su apoyo en la identificación de fallas, la realización de pruebas continuas y la formulación de sugerencias clave para mejorar la experiencia de uso de la aplicación. Asimismo, se extiende un especial reconocimiento al ginecólogo Edison Tavera, cuyas observaciones clínicas permitieron fortalecer la pertinencia de la herramienta para el acompañamiento a mujeres gestantes.

Finalmente, se agradece a la Corporación Universitaria Remington por el respaldo institucional brindado y por facilitar los espacios necesarios para el despliegue y funcionamiento de la API que soporta la solución desarrollada.

7. APOORTE DE CADA UNO DE LOS AUTORES

Investigación, escritura del borrador original (José Miguel Henao Gómez), (John Andersson Valencia), Supervisión y adquisición de la financiación (Jhon Fredy Narvaez Valderrama), Validación y edición (Juan José García Londoño), Conceptualización (Edison Andrés Tavera) y Metodología y escritura del borrador original (Porfirio Álvarez),

8. REFERENCIAS BIBLIOGRÁFICAS

- [1]. WHO. WHO Recommendations on Antenatal Care for a Positive Pregnancy Experience: Summary. The Lancet. 2016;387(10017). Available from: <https://www.who.int/publications/i/item/9789241549912>
- [2]. Narváez-Valderrama JF, Alzate-B SV, Correa-Gil V, García-L JJ, Bedoya-Soto JM, Molina-P FJ, et al. Traffic and Industrial Contributions of Particle-Bound PAHs during an Air Pollution Event in the Metropolitan Area of Medellin-Colombia: Inhalation Intake Risk during Pregnancy. Vol. 15, Atmosphere. 2024. Available from: <https://doi.org/10.3390/atmos15020173>
- [3]. World Health Organization. Managing Complications in Pregnancy and Childbirth (MCP): A Guide for Midwives and Doctors Background and Purpose. WhoInt. 2017;(May). Available from: <https://www.who.int/publications/i/item/9789241565493>
- [4]. Organización Panamericana de la Salud. Salud en las Americas 2007 Volumen I - Regional. No 622. 2007;(622). Available from: <https://lildbi.fcm.unc.edu.ar/lildbi/tesis/Salud-Americas-2007-vol-1.pdf>

- [5]. Say L, Chou D, Gemmill A, Tunçalp Ö, Moller AB, Daniels J, et al. Global causes of maternal death: A WHO systematic analysis. *Lancet Glob Health*. 2014;2(6). Available from: [https://doi.org/10.1016/S2214-109X\(14\)70227-X](https://doi.org/10.1016/S2214-109X(14)70227-X)
- [6]. Gómez-Sánchez PI, Arévalo-Rodríguez I, Rubio-Romero JA, Guío JA, Osorio-Castaño JH, Buitrago-Gutiérrez G, et al. Guías de práctica clínica para la prevención, detección temprana y tratamiento de las complicaciones del embarazo, parto o puerperio: Introducción y metodología. Vol. 64, *Revista Colombiana de Obstetricia y Ginecología*. 2013. Available from: <https://doi.org/10.18597/issn.0034-7434>
- [7]. World Health Organization. (2021). WHO global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. World Health Organization. Available from: <https://www.who.int/publications/i/item/9789240034228>
- [8]. Johanna Bermúdez Lopera. Presencia de hidrocarburos aromáticos policíclicos en material particulado fino en el centro de Medellín [Internet]. [Medellín]: Universidad Nacional de Colombia; 2022 [cited 2025 Nov 30]. Available from: <https://repositorio.unal.edu.co/handle/unal/83837>
- [9]. WHO. Consolidated telemedicine implementation guide. World Health Organization. 2022. Available from: <https://www.who.int/publications/i/item/9789240059184>
- [10]. Clínica Universitaria Bolivariana y la Universidad Pontificia Bolivariana [Internet]. 2022. HADA Domiciliaria, un proyecto pionero en la atención a distancia de mujeres en embarazo de alto riesgo.
- [11]. DeNicola N, Grossman D, Marko K, Sonalkar S, Butler Tobah YS, Ganju N, et al. Telehealth Interventions to Improve Obstetric and Gynecologic Health Outcomes: A Systematic Review. Vol. 135, *Obstetrics and Gynecology*. 2020. Available from: <https://doi.org/10.1097/AOG.0000000000003646>
- [12]. Jockusch J, Schneider S, Hochuli A, Joerger-Messerli MS, Surbek D, Radan AP. Healthcare Professionals' Perceptions and Acceptance of Telemonitoring During Pregnancy and Early Labor: A Single-Center Survey. *Int J Environ Res Public Health* [Internet]. 2025 Nov 19;22(11):1753. Available from: <https://doi.org/10.3390/ijerph22111753>
- [13]. Schwaber K, Sutherland J. *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*. 2020.
- [14]. *Software engineering: A practitioner's approach*. *Advances in Engineering Software* (1978). 1983;5(3).
- [15]. Tashildar A, Shah N, Gala R, Giri T, Chavhan P. APPLICATION DEVELOPMENT USING FLUTTER. *International Research Journal of Modernization in Engineering Technology and Science @International Research Journal of Modernization in Engineering*. 2020;02(08).
- [16]. Relan K. *Beginning with Flask*. In: *Building REST APIs with Flask*. 2019.
- [17]. Aggarwal S. *Flask Framework Cookbook*. Vol. 53, *Journal of Chemical Information and Modeling*. 2019.
- [18]. Juba S, Volkov A. *Learning PostgreSQL 11*. Pack Publishing. 2019.
- [19]. The PostgreSQL Global Development Group. *What Is PostgreSQL?* PostgreSQLOrg. 2021;
- [20]. Jones M, Bradley J, Sakimura N. RFC 7519: Json web token (JWT). Internet Engineering Task Force (IETF). 2015.
- [21]. Hoofnagle CJ, Sloot B van der, Borgesius FZ. The European Union general data protection regulation: What it is and what it means. *Information and Communications Technology Law*. 2019;28(1). Available from: <https://doi.org/10.1080/13600834.2019.1573501>



- [22].Albertengo G, Debele FG, Hassan W, Stramandino D. On the performance of web services, google cloud messaging and firebase cloud messaging. Digital Communications and Networks. 2020;6(1).
- [23].Tenner E. The Design of Everyday Things by Donald Norman (review). Technol Cult. 2015;56(3).
- [24].Quiñones D, Rusu C, Rusu V. A methodology to develop usability/user experience heuristics. Comput Stand Interfaces. 2018;59.
- [25].Shneiderman B. Designing the user interface strategies for effective human-computer interaction. ACM SIGBIO Newsletter. 1987;9(1).
- [26].Karen E., Moreno P, Mayerly GK, Gómez M. Nueve meses, una experiencia: Diseño, evaluación y mejora de la experiencia de usuario en una aplicación móvil para mujeres gestantes [Internet]. [Bogotá]: Universidad Nacional Abierta y a Distancia; 2025 [cited 2025 Nov 30]. Available from: <https://repository.unad.edu.co/handle/10596/74540>
- [27].Wcag. Web Content Accessibility Guidelines (WCAG) Overview. Web Accessibility Initiative. 2012;(December 2008).
- [28].Jones M, Bradley J, Sakimura N. JSON Web Token (JWT). Internet Engineering Task Force (IETF). 2015;
- [29].Guide US. Ubuntu Server Guide. Methods. 2010;
- [30].Flask. Foreword. 2010. Flask web development, one drop at a time.

