

**Plataforma para el monitoreo de alertas y la toma de decisiones en sistemas de alerta
temprana**

Juan Sebastián Cortes Botina

Juan José Lebaza Muñoz

Asesor

Paola Andrea Mateus Abaunza

Universidad Nacional Abierta y a Distancia UNAD
Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI
Ingeniería en Telecomunicaciones y Electrónica

2026

Nota de Aceptación

Nombre Director de Trabajo de Grado

Jurado

Jurado

Dedicatoria

A mi abuela que desde el cielo sigue guiando mis pasos y protegiéndome con su amor eterno; a mi mamá por ser mi sostén, mi apoyo incondicional y por toda su crianza llena de paciencia, cuidado y sacrificio; y a mi novia, por darme fuerzas, por abrazar mis procesos, por caminar conmigo y estar presente en todo momento... gracias a ustedes tres, porque con su amor yo nunca camino solo. Ustedes son mi equilibrio, mis razones y mi impulso. Son mi hogar en todas sus formas. Y cada logro que hoy se escribe aquí... también les pertenece **Juan Sebastián Cortes**

Botina

A mis padres y hermanos, a mi pareja y a mi hija, por ser el pilar más firme en mi vida. Gracias por creer en mí desde el inicio, por acompañarme en los buenos y en los difíciles momentos, y por recordarme siempre que no hay meta imposible cuando se tiene amor y apoyo sincero. Cada logro alcanzado y cada meta cumplida llevan un pedacito de ustedes, porque este camino lo recorrimos juntos. Ustedes son mi fuerza, mi inspiración y el motivo por el que nunca me rindo.

Juan José Lebaza Muñoz

Agradecimientos

Agradecemos primeramente a Dios, por brindarnos la sabiduría, la fortaleza y la salud necesarias para culminar con éxito este trabajo de grado aplicado. Su guía y bendición fueron fundamentales en cada etapa del proceso.

De igual manera, expresamos nuestro sincero agradecimiento a la Universidad, cuyo modelo de formación virtual nos permitió desarrollar nuestras capacidades académicas y profesionales con flexibilidad, acceso a recursos de calidad y acompañamiento constante. Esta modalidad ha representado una oportunidad significativa para integrar el aprendizaje con nuestras actividades personales y laborales, fortaleciendo nuestro compromiso con la excelencia.

Nuestro reconocimiento especial a los directores del proyecto, Diego Enrique Guzmán Villamarín y Paola Andrea Mateus Abaunza, por su valioso apoyo, orientación y aportes técnicos que enriquecieron el desarrollo de esta investigación. Su experiencia y dedicación fueron pilares fundamentales para alcanzar los objetivos propuestos.

Finalmente, extendemos nuestro agradecimiento a todas las personas que colaboraron en la aplicación de las encuestas, aportando su tiempo y disposición para contribuir con información clave al desarrollo de este trabajo.

Resumen

Este artículo presenta el diseño y desarrollo de una plataforma tecnológica de bajo costo orientada al fortalecimiento de los Sistemas de Alerta Temprana (SAT) en zonas rurales de Colombia, donde la alta vulnerabilidad frente a desastres naturales contrasta con la limitada infraestructura tecnológica disponible. La arquitectura propuesta integra una base de datos geoespacial robusta en PostgreSQL/PostGIS, una API REST desarrollada en Kotlin/Ktor y un panel interactivo en Flutter/Dart, permitiendo la adquisición, procesamiento y visualización de datos ambientales en tiempo real.

El sistema garantiza la integridad y confidencialidad de la información mediante protocolos de comunicación seguros y mecanismos de autenticación de usuarios. Asimismo, se diseñó un manual de usuario, facilitando su adopción en entornos operativos. La metodología implementada se estructuró en cinco fases: diagnóstico, diseño, desarrollo, pruebas y evaluación, asegurando coherencia técnica y escalabilidad. Los resultados experimentales, validados mediante simulaciones en Android y Python y pruebas de carga con Apache JMeter, evidenciaron un flujo de datos estable, con una tasa de éxito del 96,9 % y una latencia promedio de 9,88 s bajo 2000 solicitudes concurrentes.

La plataforma propuesta mejora la detección temprana de amenazas como inundaciones y deslizamientos, fortalece la capacidad de respuesta local y contribuye a la resiliencia comunitaria. Además, su arquitectura modular y de código abierto la convierte en una solución replicable en contextos con recursos limitados, ofreciendo un modelo de referencia viable para la modernización de infraestructuras SAT en América Latina.

Palabras clave: Sistemas de Alerta Temprana, PostgreSQL, Flutter, sensores ambientales, integración IoT, API Ktor.

Abstract

This article presents the design and development of a low-cost technological platform aimed at strengthening Early Warning Systems (EWS) in rural areas of Colombia, where high vulnerability to natural disasters contrasts with the limited technological infrastructure available. The proposed architecture integrates a robust geospatial database in PostgreSQL/PostGIS, a REST API developed in Kotlin/Ktor, and an interactive dashboard built with Flutter/Dart, enabling the acquisition, processing, and real-time visualization of environmental data.

The system ensures data integrity and confidentiality through secure communication protocols and user authentication mechanisms. Additionally, a user manual was designed to facilitate its adoption in operational environments. The implemented methodology was structured into five phases—diagnosis, design, development, testing, and evaluation—ensuring technical coherence and scalability. Experimental results, validated through simulations in Android and Python and load testing with Apache JMeter, demonstrated a stable data flow, with a success rate of 96.9% and an average latency of 9.88 seconds under 2,000 concurrent requests.

The proposed platform enhances early detection of threats such as floods and landslides, strengthens local response capacity, and contributes to community resilience. Moreover, its modular and open-source architecture makes it a replicable solution in resource-limited contexts, offering a viable reference model for the modernization of EWS infrastructures in Latin America.

Keywords: Early Warning Systems, PostgreSQL, Flutter, environmental sensors, IoT integration, Ktor API.

Tabla de Contenido

Introducción	13
Justificación	15
Objetivos.....	17
Objetivo General.....	17
Objetivos Específicos	17
Gestión del Riesgo de Desastres y la Importancia de los SAT	20
Pluviómetro.....	21
Limnómetro.	21
Distanciómetro.....	21
Inclinómetro.....	21
Fisurómetro.....	22
Modelo de Base de Datos	22
Lógica de Procesamiento en el Back-end.....	22
Seguridad de la Información.....	23
Marco teórico	24
Diagnóstico y Recolección de Requisitos	28
Integración de Sensores Ambientales.....	29

Visualización en Tiempo Real	30
Gestión de Datos Georreferenciados.	30
Seguridad y Autenticación.....	31
Escalabilidad.....	31
Diseño Del Sistema.....	32
Selección de la Tecnología	32
Backend: Kotlin/Ktor.....	32
Compatibilidad Multiplataforma.	32
Simplicidad en la Sintaxis y Mantenimiento.	33
Rendimiento del Servidor.	33
Base de datos: PostgreSQL/PostGIS	33
Interfaz Web: Flutter/Dart	33
Rendimiento Nativo.....	34
Actualización en tiempo real.	34
Compatibilidad con APIs REST.....	34
Modelado del sistema	35
Arquitectura General	35

Modularidad	36
Escalabilidad	36
Mantenibilidad	36
Desarrollo Tecnológico.....	37
Base de Datos	38
City.....	38
SensorType.	39
Sensor.....	40
Measurement.....	41
Frontend (Interfaz Web)	42
Inicio de Sesión.....	43
Página de Inicio.....	44
Visualización de Sensores.....	44
Visualización de Alarmas.	45
Histórico de Medidas de Sensores y Descarga:	46
Integración del Sistema	47
Pruebas Y Validación	49

	10
Pruebas Funcionales	49
Pruebas de Rendimiento con Apache JMeter	50
Validación Integral del Sistema.....	52
Evaluación de Usabilidad de la Plataforma Web	53
Recomendaciones de Mejora.....	55
Recomendaciones en Futuras Versiones	56
Evaluación final	58
Resultados	59
Discusión.....	60
Conclusión	61
Referencias Bibliográficas	62

Lista de Figuras

Figura 1 <i>Tabla City SQL</i>	39
Figura 2 <i>Tabla tipo de Sensor SQL</i>	40
Figura 3 <i>Tabla de sensor SQL</i>	41
Figura 4 <i>Tabla de medidas SQL</i>	42
Figura 5 <i>Pantalla Inicio de Sesión</i>	43
Figura 6 <i>Pantalla Página de inicio</i>	44
Figura 7 <i>Pantalla de visualización de sensores</i>	45
Figura 8 <i>Pantalla de visualización de Alarmas</i>	46
Figura 9 <i>Pantalla de Histórico de medidas</i>	47
Figura 10 <i>Pantalla Aplicación móvil Android</i>	49
Figura 11 <i>Prueba con Apache Jmeter</i>	51
Figura 12 <i>Resultados Apache Jmeter</i>	52

Lista de Tablas

Tabla 1 <i>Sensores utilizados y su función dentro del SAT</i>	29
Tabla 2 <i>Resultado promedio de la encuesta de usabilidad (n=10)</i>	54
Tabla 3 <i>Resultados Obtenidos</i>	59

Introducción

Colombia es uno de los países más propensos a desastres naturales en América Latina debido a su ubicación geográfica, condiciones topográficas y variabilidad climática. Durante la última década, eventos como inundaciones, deslizamientos y crecientes súbitas han afectado gravemente a la población, especialmente en zonas rurales (Peñas, 2017). Según datos del Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM, 2023), entre 2010 y 2022 se registraron más de 12.000 eventos naturales, lo que evidencia la necesidad de fortalecer los mecanismos de monitoreo y respuesta ante emergencias.

Los Sistemas de Alerta Temprana (SAT) son una herramienta fundamental en la gestión del riesgo, ya que permiten la detección y comunicación oportuna de amenazas naturales (Moreno et al., 2014). Sin embargo, en Colombia aún se presentan limitaciones relacionadas con la cobertura tecnológica, los altos costos de implementación y la falta de integración entre los sistemas existentes (Viloria & Quintero, 2016). Tal como se menciona en estudios recientes, la efectividad de un SAT depende de la calidad de los datos, la confiabilidad de los sensores y la capacidad del sistema para procesar información en tiempo real (López-García et al., 2017). Ante esta problemática, el uso de tecnologías abiertas y multiplataforma ha demostrado ser una solución viable para el fortalecimiento de los SAT. La adopción de herramientas como PostgreSQL/PostGIS, Kotlin/Ktor y Flutter/Dart permite desarrollar plataformas tecnológicas escalables, seguras y de bajo costo, que mejoran la recopilación, procesamiento y visualización de datos ambientales (Leal, 2025; García Solís, 2025; Hernández, 2024). Estas tecnologías, al ser de código abierto, facilitan además la interoperabilidad y replicabilidad de los sistemas en diferentes contextos territoriales.

En este trabajo se presenta el desarrollo de una página web y un conjunto de API REST orientadas al monitoreo de alertas y la toma de decisiones en Sistemas de Alerta Temprana. El backend fue implementado en Kotlin/Ktor, incorporando diversos endpoints para la gestión de la información ambiental, tales como /cities para la consulta de municipios, /sensors y /sensors/all para la visualización de sensores activos, /measurement para el registro y consulta de mediciones, /sensor/measurement/last para obtener la última lectura registrada y /login para la autenticación de usuarios. Estos servicios permiten establecer la comunicación directa entre los sensores, la base de datos y la interfaz web, garantizando la transmisión eficiente y segura de los datos.

La plataforma integra una base de datos geoespacial en PostgreSQL/PostGIS, la cual almacena de forma estructurada y georreferenciada la información proveniente de los sensores, y una interfaz desarrollada en Flutter/Dart que permite la visualización dinámica de los datos en tiempo real. Esta arquitectura tecnológica facilita el seguimiento de variables ambientales críticas y la generación automática de alertas, contribuyendo al fortalecimiento de la gestión del riesgo y la resiliencia comunitaria (IDEAM, 2023; López-García et al., 2017).

En conjunto, la propuesta busca ofrecer una solución tecnológica accesible y replicable que apoye la labor de instituciones que monitorean riesgos en Colombia, promoviendo una respuesta más oportuna ante amenazas naturales y fortaleciendo los procesos de prevención en comunidades rurales.

Justificación

De acuerdo con la Unidad Nacional para la Gestión del Riesgo de Desastres (2016) la creciente frecuencia e intensidad de desastres naturales en Colombia, atribuida en gran parte al cambio climático, ha puesto en evidencia la fragilidad de los mecanismos actuales de monitoreo y alerta en el país. Si bien existen esfuerzos institucionales para enfrentar esta problemática, como los Sistemas de Alerta Temprana (SAT), estos enfrentan importantes limitaciones en cobertura, interoperabilidad y acceso a tecnologías eficientes, especialmente en comunidades rurales donde el riesgo es mayor y la infraestructura es más débil.

En este contexto, se vuelve crucial diseñar soluciones tecnológicas que permitan superar las barreras existentes en el procesamiento y visualización de datos ambientales. Una plataforma unificada y flexible, que integre información proveniente de sensores diversos y que esté orientada a facilitar la toma de decisiones en tiempo real, representa una respuesta innovadora y necesaria para fortalecer la Gestión del Riesgo en Colombia.

Este proyecto se justifica por su enfoque integral: no solo propone una solución técnica, sino también una herramienta inclusiva, pensada para adaptarse a las realidades locales, con una interfaz intuitiva y accesible. Además, al incorporar principios de diseño centrado en el usuario, procesamiento eficiente de datos y seguridad en la información, se asegura un sistema robusto y confiable que contempla las principales necesidades para la Unidad Nacional para la Gestión del Riesgo de Desastres (UNGRD).

Más allá de su aplicabilidad inmediata, esta plataforma tiene el potencial de escalarse y replicarse en múltiples territorios del país, generando impactos sostenibles en términos de reducción de riesgos, prevención de pérdidas humanas y materiales, y empoderamiento de las comunidades frente a las amenazas naturales. La articulación entre tecnología, accesibilidad y

gestión del riesgo convierte este proyecto en una apuesta pertinente, oportuna y alineada con las prioridades del desarrollo territorial sostenible. (López-García, J. D., Carvajal-Escobar, Y., & Enciso-Arango, A. M, 2017).

Objetivos

Objetivo General

Proponer una plataforma flexible para la recepción y visualización de datos de Sistemas de Alerta Temprana (SAT) a través de una API, que permita la detección temprana de amenazas naturales para la toma de decisiones de manera oportuna en la gestión del riesgo.

Objetivos Específicos

Desarrollar la estructura del back-end de la plataforma, mediante un diseño modular que permita la adquisición flexible y eficiente de datos.

Diseñar un front-end tipo dashboard que presente los datos de los sistemas SAT, por medio de tecnologías web interactivas que faciliten la visualización en tiempo real e interpretación intuitiva de las variables monitoreadas.

Diseñar una API que facilite la integración y transmisión estructurada de datos, utilizando protocolos estandarizados que conecten base de datos y dashboard de visualización de manera segura y escalable.

Verificar la flexibilidad de la herramienta, por medio de pruebas con datos de al menos dos fuentes diferentes que emulen el comportamiento del sistema SA.

Planteamiento del Problema

Colombia es un país altamente vulnerable a los desastres naturales provocados por fenómenos hidrometeorológicos y geodinámicos, cuya frecuencia e intensidad se han visto agravadas por el cambio climático. Eventos como lluvias torrenciales, inundaciones y movimientos en masa representan amenazas constantes para la población, especialmente en regiones rurales con baja capacidad de respuesta. Según datos del IDEAM, entre 2010 y 2022 se registraron más de 12.000 eventos de este tipo, que ocasionaron más de 3.000 muertes y afectaron a más de 4 millones de personas. Esta situación evidencia una necesidad urgente de fortalecer los sistemas de monitoreo y alerta para anticipar y mitigar los impactos de estos fenómenos. (IDEAM, 2023).

La falta de estandarización en la recolección y transmisión de datos, sumada a la carencia de herramientas tecnológicas accesibles, limita la capacidad predictiva y de respuesta de las comunidades. Esto se agrava por la ausencia de plataformas unificadas que permitan visualizar de manera integrada los diferentes parámetros críticos para la gestión del riesgo. Casos como el desastre en Mocoa (2017) en donde se produjo el desbordamiento de 3 ríos (Mullato, Sangoyaco y Taruca) por las fuertes lluvias cobrando la vida de más de 200 personas y miles de personas damnificadas, adicionalmente decenas de personas desaparecidas. Esto nos demuestra cómo fallas en la transmisión de datos y las desmedidas acciones humanas sobre el ecosistema fluvial pueden tener consecuencias trágicas. (Peñas, V, 2017).

A pesar de la existencia de algunos Sistemas de Alerta Temprana (SAT) en Colombia, su funcionamiento enfrenta múltiples limitaciones. Entre ellas se encuentran la cobertura insuficiente, el alto costo de los equipos, la dependencia de conectividad en tiempo real y la fragmentación de los datos recolectados. En muchas zonas rurales, la infraestructura tecnológica

es precaria, lo que impide el uso de soluciones convencionales de monitoreo, que operan bajo distintos protocolos e interfaces, dificultando su integración en una plataforma común. Como resultado, los datos se recogen de forma aislada, con baja frecuencia, y en muchos casos se pierden o no se actualizan a tiempo, lo que reduce la eficacia de los SAT. (Unidad Nacional para la Gestión del Riesgo de Desastres, 2016).

¿qué características debe tener una plataforma de alerta temprana, que permita el análisis y visualización de datos para la gestión del riesgo por desastres naturales en zonas rurales de Colombia?

Marco Conceptual

La creciente frecuencia e intensidad de los desastres naturales en Colombia, especialmente aquellos asociados a fenómenos hidrometeorológicos y geodinámicos, ha evidenciado la necesidad urgente de fortalecer los mecanismos de prevención y respuesta ante emergencias, particularmente en zonas rurales con alta vulnerabilidad y limitada infraestructura tecnológica. En este contexto, los Sistemas de Alerta Temprana (SAT) se consolidan como herramientas fundamentales en la gestión del riesgo de desastres, al permitir la detección oportuna de amenazas y la activación de medidas preventivas que pueden salvar vidas y mitigar daños materiales.

Gestión del Riesgo de Desastres y la Importancia de los SAT

La gestión del riesgo de desastres se sustenta en un enfoque preventivo que reconoce la relación entre amenazas naturales, vulnerabilidad social y capacidad de respuesta. Los SAT forman parte de esta estrategia al ofrecer mecanismos que permiten monitorear condiciones del entorno, interpretar señales de alerta y comunicar dicha información a las comunidades y autoridades. De acuerdo con Aguirre Buenaventura (2022), un SAT eficaz debe integrar cuatro componentes esenciales: conocimiento del riesgo, monitoreo y predicción, difusión de alertas, y capacidad de respuesta.

En Colombia, si bien existen SAT implementados por instituciones como el IDEAM y la Unidad Nacional para la Gestión del Riesgo de Desastres, su alcance es limitado debido a factores como la cobertura insuficiente, la desconexión entre sistemas, el alto costo de equipos especializados y la falta de plataformas integradas para el análisis de datos. Estos desafíos se agravan en territorios rurales donde la conectividad es intermitente, la alfabetización digital es

baja y la infraestructura técnica es escasa. (Unidad Nacional para la Gestión del Riesgo de Desastres, 2016)

Arquitectura del Sistema: Sensores, Base de Datos y Procesamiento

El proyecto plantea una plataforma tecnológica de bajo costo y flexible, capaz de recibir, almacenar y visualizar datos generados por sensores ambientales en tiempo real. La arquitectura del sistema se compone de, sensores de bajo costo, seleccionados por su funcionalidad, disponibilidad en el mercado y facilidad de implementación. Estos incluyen:

Pluviómetro. “El pluviómetro es un instrumento que sirve para medir la cantidad de lluvia. Su uso es fundamental en estudios ambientales, hidrológicos y meteorológicos, permitiendo analizar riesgos, el comportamiento de cuencas y la calidad del suelo”. (Higuera & Quevedo, 2015)

Limnómetro. El Limnómetro es una regla graduada que permite medir las fluctuaciones del nivel del agua en ríos, lagos u otras masas de agua. Se instala verticalmente en el lecho del cuerpo de agua y debe ubicarse en un lugar bien visible para facilitar la lectura desde un puente o la orilla. Esta lectura proporciona la profundidad o tirante del agua, información esencial para el monitoreo hidrológico y la prevención de inundaciones. Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI, 2008).

Distanciómetro. “El distanciómetro es un instrumento de medición que utiliza tecnología láser o ultrasónica para calcular la distancia entre dos puntos. Emite un haz de luz o sonido que rebota en el objetivo y regresa al dispositivo, calculando la distancia con gran precisión”. (Equipos Topografía, 2024).

Inclinómetro. “El inclinómetro es un instrumento utilizado para medir los cambios de inclinación en el suelo o estructuras, proporcionando datos sobre la magnitud y dirección del

desplazamiento. Está compuesto por una sonda que se inserta en un tubo instalado previamente en el terreno a monitorear”. (MDH, 2024)

Fisurómetro. El fisurómetro es un instrumento diseñado para medir y monitorear los cambios en la apertura de grietas o fisuras en estructuras. Consiste en dos placas superpuestas, una con una escala graduada y otra con líneas de referencia, que se fijan a ambos lados de la fisura. Al producirse movimientos en la estructura, las placas se desplazan entre sí, permitiendo registrar la magnitud y dirección del desplazamiento. Este seguimiento es esencial para evaluar la estabilidad estructural y prevenir posibles fallos. (Thingtrack, 2025)

Modelo de Base de Datos

“Diseñado para almacenar tanto datos estructurados como no estructurados, garantizando eficiencia, escalabilidad y seguridad. El modelo considera la recolección heterogénea de datos, su normalización y posterior análisis”. (Xiloj López, R. A., 2025)

Lógica de Procesamiento en el Back-end

“Permite validar la información recibida, identificar patrones anómalos y activar alertas cuando se superan umbrales críticos predefinidos. Este componente es esencial para la automatización del sistema y su operatividad en tiempo real”. (García Solís, J. C., 2025)

Visualización y Experiencia de Usuario

La visualización de la información representa un componente crucial para garantizar la utilidad del SAT, especialmente en contextos con baja alfabetización digital. Por ello, el sistema contempla el desarrollo de un dashboard interactivo que muestre los datos en tiempo real de manera clara, comprensible y georreferenciada. Esta interfaz gráfica permitirá a los usuarios acceder a:

Mapas interactivos con puntos de monitoreo.

Gráficos históricos y en tiempo real.

Indicadores visuales de alerta (colores, íconos, notificaciones).

Reportes descargables para análisis comunitario o institucional.

Según Leal (2024), la satisfacción informativa es crucial, ya que determina la facilidad con la que los usuarios pueden procesar la información presentada en los Dashboards. Si los datos son precisos y claros, los usuarios experimentan una mayor satisfacción, lo que a su vez simplifica la tarea de analizar información compleja.

Seguridad de la Información

Dado que el sistema manejará datos sensibles sobre amenazas naturales y usuarios registrados, se integrarán mecanismos de seguridad como autenticación de usuarios, control de acceso por roles. Esto garantizará la integridad de los datos, la protección contra accesos no autorizados. (DataSunrise, 2024)

Marco Teórico

Se presenta la descripción de varios proyectos relacionados con plataformas web para Sistemas de Alerta Temprana (SAT), los cuales fueron identificados a través de una búsqueda realizada en Google Académico utilizando la palabra clave plataforma web SAT. Esta búsqueda permitió localizar publicaciones científicas que documentan experiencias relevantes en el diseño, implementación y aplicación de SAT en distintos contextos geográficos, con énfasis en herramientas tecnológicas orientadas a la gestión del riesgo por inundaciones y otros eventos climáticos extremos.

El primer proyecto desarrollado por Coll, M. A. (2013) de título Sistemas de alerta temprana (SAT) para la reducción del riesgo de inundaciones súbitas y fenómenos atmosféricos en el área metropolitana de Barranquilla. Consiste en el diseño y desarrollo de un prototipo de Sistema de Alertas Tempranas (SAT) basado en tecnologías de la información y comunicación (TIC), con el objetivo de monitorear en tiempo real el comportamiento de los arroyos en la ciudad de Barranquilla, especialmente el arroyo "La Brigada", para alertar a la ciudadanía sobre niveles de riesgo por inundaciones durante la temporada de lluvias. Esta iniciativa interdisciplinaria, desarrollada por programas de ingeniería, busca mitigar los efectos de la urbanización descontrolada y el cambio climático, que han provocado el colapso del sistema de drenaje urbano. El sistema integra sensores, redes inalámbricas tipo Zigbee, plataformas de hardware libre como Arduino y una interfaz web para la visualización de datos sobre caudal, nivel y fuerza del agua. Además, se complementa con un proyecto paralelo de radar meteorológico Doppler y polarimétrico para detectar precipitaciones y fenómenos atmosféricos, permitiendo identificar zonas inundables en el área metropolitana de Barranquilla y reforzar la gestión del riesgo a nivel local y regional.

El segundo proyecto realizado por Vilorio, A., & Quintero, W. (2016) de título Gestión de información sat río Manzanares. Tiene como sede Santa Marta, este consiste en el diseño e implementación de un Sistema de Alerta Temprana (SAT) para monitorear en tiempo real variables como nivel del agua, humedad y precipitaciones en el río Manzanares, con el fin de emitir alertas oportunas a la población vulnerable y a las autoridades competentes, mitigando así el riesgo de inundaciones. Este sistema combina una red de sensores físicos, una plataforma web para la gestión de proyectos y análisis de datos, y una aplicación Android para la recepción de notificaciones de alerta, integrando tecnologías como JSF, EJB, Hibernate, Glassfish y Google Cloud Messaging. El SAT se apoya en modelos de monitoreo, análisis, juicio y acción, y busca fortalecer la gestión del riesgo en el contexto urbano de Santa Marta, permitiendo una respuesta más eficiente ante eventos climáticos extremos.

El tercer proyecto realizado por Moreno, D., Quiñones, É., & Tovar, L. C. (2014) de título Los Sistemas de Alerta Temprana, SAT, una herramienta para la prevención de desastres por inundación y efectos del cambio climático. Aborda la compleja problemática de las inundaciones en Colombia, destacando tanto las causas naturales; como lluvias intensas, huracanes, el fenómeno de la niña, causas antrópicas, incluyendo la ocupación inadecuada de zonas inundables y el mal manejo del recurso hídrico, factores que han generado graves impactos sociales, económicos y ambientales, especialmente en el Caribe colombiano donde el cambio climático ha exacerbado riesgos como el ascenso del nivel del mar y la erosión costera. Se presentan antecedentes históricos y datos técnicos que ilustran la magnitud del problema, enfatizando la necesidad urgente de fortalecer la gestión del riesgo, integrar el conocimiento hidrológico y climático en la planificación territorial y aplicar políticas públicas efectivas. En el ámbito internacional, se revisan diversas iniciativas de prevención mediante Sistemas de Alerta

Temprana (SAT), con ejemplos en América Latina como el proyecto PREDESANDÍA en Perú y el SIATA en Colombia, así como plataformas regionales como PROHIMET y SATCA en Centroamérica, que combinan tecnologías de monitoreo y comunicación para mejorar la capacidad de respuesta ante eventos hidrometeorológicos. También se destacan proyectos en África, Europa y Asia, como el plan holandés “Room for the River” y sistemas basados en WebGIS en China, que integran innovación tecnológica y gestión ambiental. Finalmente, se resalta el papel fundamental de las entidades ambientales nacionales y regionales, especialmente las Corporaciones Autónomas Regionales y el Sistema Nacional Ambiental en Colombia, en la implementación, monitoreo y sostenibilidad de estos sistemas para reducir la vulnerabilidad y proteger a las comunidades frente a las inundaciones.

El cuarto proyecto realizado por Peralta, G., De Elia, E. A., Lanfri, M. A., Porcasi, X., Frutos, N., Rotela, C., & Scavuzzo, M. (2011) de título Geomática aplicada a un Sistema de Alerta Temprana. Es una plataforma informática desarrollada por el Instituto Gulich (CONAE – UNC) para la estratificación del riesgo de dengue en Argentina, que integra datos geoespaciales, ambientales y epidemiológicos en una arquitectura modular y escalable. Desde el punto de vista geológico y territorial, el sistema permite analizar variables como temperatura, humedad, presencia del vector y condiciones del entorno físico, integrándolas en modelos multiescalares que generan mapas de riesgo a nivel nacional y urbano. La plataforma se basa en tecnologías open source como GeoServer y Geomajas, que permiten la gestión, edición y visualización de capas geográficas mediante estándares OGC (WMS, WFS), con una interfaz web robusta y amigable. Utilizando patrones de diseño como MVC a través de Spring, el sistema automatiza el flujo de datos desde la carga hasta la generación de alertas, garantizando interoperabilidad con

otras herramientas GIS y permitiendo el análisis espacial desde cualquier navegador, con soporte para imágenes satelitales como las SAR para enriquecer los análisis del entorno físico.

El quinto proyecto realizado por Alexander, R. V. J., Derley, R. J., & Fernando, A. D., (2018) de título diseño de un sistema de alerta temprana ante inundaciones, sequía e incendios forestales como medida adaptación a la variabilidad y cambio climático. Es un sistema de Alerta Temprana (SAT) en el departamento de Norte de Santander, Colombia, surge como respuesta a la alta vulnerabilidad frente a eventos extremos como inundaciones, deslizamientos e incendios forestales, intensificados por fenómenos de variabilidad y cambio climático. La región, caracterizada por una geografía compleja y cuencas binacionales como la del río Zulia y Pamplonita, carecía de infraestructura adecuada para monitoreo climático en tiempo real. El proyecto implementó una red de estaciones meteorológicas e hidrológicas automáticas, una plataforma WebSIG y modelos climáticos simples, fortaleciendo la capacidad de anticipación y respuesta frente a amenazas. Además, se promovió la apropiación institucional y comunitaria del sistema mediante procesos de socialización, capacitación y difusión, integrando tecnología y saber local en la gestión del riesgo y la adaptación climática.

Metodología

La metodología implementada en este proyecto se basó en un enfoque cuantitativo, aplicado y de desarrollo tecnológico, orientado a la creación de una plataforma web para el monitoreo de alertas en Sistemas de Alerta Temprana (SAT). El proceso metodológico se estructuró en cinco fases: diagnóstico y levantamiento de requisitos, diseño del sistema, desarrollo tecnológico, pruebas y validación, y evaluación final. Cada fase fue ejecutada de manera secuencial, garantizando la coherencia técnica y funcional del sistema propuesto.

Diagnóstico y Recolección de Requisitos

La fase de diagnóstico se centró en analizar las condiciones actuales de los Sistemas de Alerta Temprana (SAT) en Colombia, con el propósito de establecer los requerimientos técnicos necesarios para el desarrollo de una plataforma web orientada a la recolección, procesamiento y visualización de datos ambientales en tiempo real.

De acuerdo con el Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM, 2023), Colombia presenta una alta recurrencia de eventos hidrometeorológicos como inundaciones, deslizamientos y avenidas torrenciales, que afectan de manera recurrente a comunidades rurales. Por su parte, la Unidad Nacional para la Gestión del Riesgo de Desastres (UNGRD, 2016) destaca la importancia de fortalecer las capacidades de observación, monitoreo y comunicación dentro de los SAT, promoviendo el uso de herramientas tecnológicas accesibles y sostenibles.

Con base en estas directrices, se identificaron diversas limitaciones en los sistemas actuales, tales como la baja cobertura de sensores, la falta de integración entre plataformas, la carencia de acceso a datos en tiempo real y los altos costos de instalación y mantenimiento. Para abordar estas problemáticas, se establecieron los requerimientos técnicos y funcionales del

sistema, priorizando la integración de sensores ambientales, la interoperabilidad entre módulos, la seguridad de los datos y la escalabilidad del software.

Integración de Sensores Ambientales

El sistema propuesto contempla la implementación de una red de sensores heterogéneos con capacidad para registrar variables críticas asociadas a la ocurrencia de fenómenos naturales. Cada sensor cumple una función específica que contribuye al fortalecimiento del monitoreo ambiental dentro del SAT, tal como se describe en la Tabla 1.

Tabla 1

Sensores utilizados y su función dentro del SAT

Sensor	Importancia Técnica en el SAT
Pluviómetro	Permite registrar la cantidad de lluvia caída durante un periodo determinado. Es fundamental para el análisis de riesgos por inundaciones, estudios hidrológicos y evaluación del comportamiento de las cuencas (Higuera & Quevedo, 2015).
Limnómetro	Permite medir las fluctuaciones del nivel de agua, facilitando la detección temprana de crecientes o desbordamientos. Es una herramienta esencial en el monitoreo hidrológico y la prevención de inundaciones (SENAMHI, 2008).
Distanciómetro	Utiliza tecnología láser o ultrasónica para obtener mediciones precisas. Se emplea para complementar las lecturas del limnómetro y mejorar la exactitud en el control del nivel del agua (Equipos Topografía, 2024).

Inclinómetro	Detecta cambios en la inclinación del terreno que pueden indicar movimientos de masa. Es fundamental para la prevención de deslizamientos de tierra en zonas de ladera (MDH, 2024).
Fisurómetro	Permite monitorear la evolución de fisuras en suelos o estructuras, ayudando a evaluar la estabilidad y prevenir fallas estructurales (Thingtrack, 2025).

Nota. La tabla describe los sensores ambientales integrados en el Sistema de Alerta Temprana, junto con su importancia técnica y función específica dentro del monitoreo hidrometeorológico y geotécnico. Esta información permite comprender el aporte de cada sensor en la detección temprana de amenazas naturales. Fuente: *Autoría propia*.

La integración de estos dispositivos garantiza una visión completa del entorno físico, permitiendo detectar variaciones significativas en las condiciones hidrometeorológicas y geotécnicas. Además, el sistema está diseñado para operar con protocolos de comunicación inalámbricos (Wi-Fi, LoRa o GSM), asegurando la transmisión continua de los datos hacia el servidor central.

Requerimientos Funcionales del Sistema

Durante la fase de diagnóstico se establecieron los principales requerimientos funcionales que deben cumplirse para asegurar la efectividad del sistema:

Visualización en Tiempo Real. La interfaz web debe mostrar las mediciones actualizadas mediante gráficos, mapas interactivos y alertas visuales que faciliten la interpretación de los datos por parte de usuarios no técnicos.

Gestión de Datos Georreferenciados. La base de datos debe almacenar las lecturas de los sensores junto con su ubicación espacial, permitiendo una correlación precisa entre los

eventos y su localización.

Seguridad y Autenticación. El sistema debe incluir control de usuarios, validación de credenciales y manejo de sesiones para proteger la información recolectada.

Escalabilidad. La arquitectura debe permitir la incorporación de nuevos sensores o zonas de monitoreo sin alterar la estructura principal del sistema.

La selección de los sensores y las especificaciones funcionales responde a la necesidad de desarrollar un Sistema de Alerta Temprana modular, replicable y de bajo costo, adecuado para comunidades rurales del país. “La integración de variables como precipitación, nivel del agua, inclinación del terreno y apertura de fisuras permite generar alertas tempranas frente a inundaciones y deslizamientos, reduciendo el impacto de los desastres naturales”. (IDEAM, 2023; UNGRD, 2016).

Asimismo, “la implementación de tecnologías abiertas facilita la sostenibilidad del sistema y fomenta su adaptación en diferentes regiones, en concordancia con las estrategias planteadas por la UNGRD para el fortalecimiento de los SAT comunitarios en Colombia”. (UNGRD, 2016).

Diseño del Sistema

El diseño del sistema se desarrolló siguiendo una arquitectura modular que permite la interacción entre el backend, la base de datos y la interfaz web, garantizando la comunicación fluida y segura entre los diferentes componentes. Esta arquitectura se estructuró considerando los requerimientos funcionales definidos en la fase de diagnóstico y las necesidades identificadas por la Unidad Nacional para la Gestión del Riesgo de Desastres (UNGRD, 2016) para fortalecer los Sistemas de Alerta Temprana (SAT).

El proceso de diseño se apoyó en la elaboración de diagramas de flujo y diagramas entidad-relación, los cuales sirvieron como base para modelar los procesos internos del sistema, la gestión de datos y la experiencia del usuario.

Selección de la Tecnología

Backend: Kotlin/Ktor

El desarrollo del backend se realizó en Kotlin utilizando el framework Ktor, debido a su eficiencia, escalabilidad y capacidad de manejar múltiples peticiones simultáneamente con bajo consumo de recursos. A diferencia de otras opciones como Node.js o Django, Ktor ofrece una integración nativa con entornos Java y herramientas multiplataforma, permitiendo construir servicios REST de alto rendimiento y bajo acoplamiento.

Además, Ktor facilita la implementación de rutas personalizadas y el control de seguridad mediante autenticación y manejo de sesiones, lo cual fue esencial para la creación de endpoints como /login, /cities, /sensors, /measurement, y /sensor/measurement/last.

La elección de Kotlin/Ktor se basó en los siguientes criterios técnicos:

Compatibilidad Multiplataforma. Kotlin permite la interoperabilidad con Java y frameworks Android, lo que favorece futuras expansiones hacia aplicaciones móviles.

Simplicidad en la Sintaxis y Mantenimiento. Su estructura moderna reduce la complejidad del código y mejora la legibilidad.

Rendimiento del Servidor. Ktor optimiza la concurrencia mediante corrutinas, lo que incrementa la velocidad de respuesta ante múltiples solicitudes (Leal, 2025; Hernández, 2024).

Base de datos: PostgreSQL/PostGIS

La base de datos se implementó utilizando PostgreSQL con la extensión PostGIS, seleccionada por su robustez, capacidad para manejar datos geoespaciales y compatibilidad con software de código abierto. A diferencia de otros sistemas como MySQL o SQLite, PostgreSQL ofrece soporte nativo para coordenadas, mapas y consultas espaciales, lo cual es esencial en un sistema de monitoreo ambiental.

El diseño de la base de datos se representó mediante el Diagrama Entidad–Relación (ER), donde se establecen las relaciones entre las tablas principales: Sensor, SensorType, City, Measurement y SensorRange. Este modelo permite:

- Registrar la ubicación geográfica de cada sensor.

- Asociar cada tipo de sensor con su ciudad y su rango de medición.

- Controlar los valores mínimos y máximos permitidos para activar alertas de riesgo.

El uso de PostGIS facilitó la visualización de las mediciones sobre mapas dinámicos, permitiendo analizar espacialmente los datos recolectados en tiempo real (DataSunrise, 2024).

Interfaz Web: Flutter/Dart

La interfaz web fue desarrollada utilizando Flutter/Dart, framework de código abierto desarrollado por Google, que permite generar aplicaciones multiplataforma con un solo código base. Esta elección se realizó por su alto rendimiento gráfico y su capacidad para crear interfaces responsivas y dinámicas, adaptables tanto a equipos de escritorio como dispositivos móviles.

A diferencia de frameworks como React o Angular, Flutter ofrece una arquitectura basada en widgets, lo que permite personalizar cada componente visual de manera eficiente y coherente.

Entre sus principales ventajas destacan:

Rendimiento Nativo. El código se compila directamente, evitando dependencias con navegadores externos.

Actualización en Tiempo Real. Permite observar los cambios del diseño durante el desarrollo (“hot reload”).

Compatibilidad con APIs REST. Se integró directamente con los endpoints desarrollados en Ktor, garantizando la comunicación constante entre el usuario y la base de datos.

La interfaz incluye módulos para inicio, inicio de sesión, configuración de perfil, visualización de sensores, historial de datos y alertas, todos diseñados a partir de diagramas de flujo que detallan la interacción del usuario con el sistema.

Modelado del Sistema

El modelado del sistema se llevó a cabo mediante la elaboración de diagramas de flujo funcionales que describieron los principales procesos de interacción entre el usuario y la plataforma. Estos diagramas fueron fundamentales durante la etapa de diseño, ya que permitieron definir la secuencia lógica de las operaciones, la navegación entre módulos y la relación entre las acciones del usuario y las respuestas del sistema.

En total, se desarrollaron cinco diagramas principales correspondientes a los módulos de inicio, autenticación, configuración, monitoreo de sensores e historial de mediciones. Cada uno de ellos fue diseñado utilizando herramientas de modelado digital, sirviendo como guía estructural para el desarrollo del backend, la interfaz web y la base de datos.

De manera complementaria, se elaboró un diagrama entidad-relación (ER) que permitió representar la estructura lógica de la base de datos y las relaciones entre las tablas principales. Este modelo facilitó la implementación en PostgreSQL/PostGIS y la gestión de datos georreferenciados asociados a los sensores ambientales.

En conjunto, los modelos gráficos desarrollados durante esta fase garantizaron la coherencia técnica y la trazabilidad entre las etapas de diseño, desarrollo y validación del sistema, contribuyendo a una arquitectura modular y escalable.

Arquitectura General

El sistema se estructuró bajo una arquitectura cliente-servidor, donde el cliente (interfaz web) consume los servicios proporcionados por el servidor (backend Ktor). El backend actúa como intermediario entre los sensores físicos y la base de datos, procesando las solicitudes mediante APIs REST y devolviendo los resultados al usuario en tiempo real. Esta arquitectura garantiza:

Modularidad. Cada componente puede ser actualizado o reemplazado sin afectar los demás.

Escalabilidad. Permite integrar nuevos sensores o funcionalidades con facilidad.

Mantenibilidad. El uso de tecnologías abiertas facilita futuras mejoras o adaptaciones del sistema

Desarrollo Tecnológico

En el desarrollo tecnológico del sistema se estructuró en tres componentes principales: backend, base de datos e interfaz web, los cuales fueron integrados bajo un enfoque modular que garantiza la escalabilidad, el rendimiento y la mantenibilidad del sistema. Para su implementación se utilizaron herramientas profesionales como IntelliJ IDEA para el desarrollo del backend, pgAdmin para la administración de la base de datos y Visual Studio Code para la creación del frontend.

Backend

El backend fue desarrollado en Kotlin, empleando el framework Ktor debido a su eficiencia, bajo consumo de recursos y compatibilidad con la Máquina Virtual de Java (JVM) (Leal, 2025; Hernández, 2024). Este lenguaje permite un manejo óptimo de peticiones concurrentes mediante corrutinas, lo que mejora significativamente el rendimiento del sistema y reduce la latencia.

El desarrollo se realizó en el entorno IntelliJ IDEA, el cual ofrece soporte completo para Kotlin y facilita la depuración y ejecución del servidor local. El backend cumple la función de intermediario entre la base de datos y la interfaz web, gestionando las solicitudes HTTP provenientes del cliente y devolviendo las respuestas en formato JSON mediante la librería Gson.

Para ello se diseñaron diversas API RESTful, encargadas de la comunicación entre módulos. Las rutas más relevantes fueron:

`/cities`: devuelve el listado de ciudades registradas.

`/sensors`: obtiene los sensores asociados a una ciudad específica.

`/measurement`: retorna las mediciones obtenidas de los sensores.

/sensor/measurement/all: muestra el historial completo de mediciones.

/sensor/measurement/range: filtra registros según rango de fechas.

/sensor/measurement/notifications: presenta las alertas generadas.

/login: autentica a los usuarios registrados en el sistema.

Cada API fue implementada bajo un enfoque modular, lo que permite agregar nuevas funcionalidades sin alterar la estructura central del servidor. Gracias a esta arquitectura, el backend actúa como un puente seguro y eficiente entre la capa de almacenamiento y la de presentación, garantizando el flujo constante de información entre los sensores, la base de datos y la interfaz web (Leal, 2025; Hernández, 2024).

Base de Datos

La base de datos se implementó en PostgreSQL, empleando la herramienta pgAdmin para su diseño, modelado y gestión. Además, se integró la extensión PostGIS, que facilita el almacenamiento y procesamiento de datos geoespaciales, permitiendo representar de manera precisa la ubicación de los sensores y los puntos de monitoreo ambiental (DataSunrise, 2024)

El modelo de datos se representó mediante un diagrama entidad–relación, en el que se definieron las siguientes tablas principales:

City. Almacena las localidades o zonas donde se ubican los sensores (figura 1).

Figura 1

Tabla City SQL

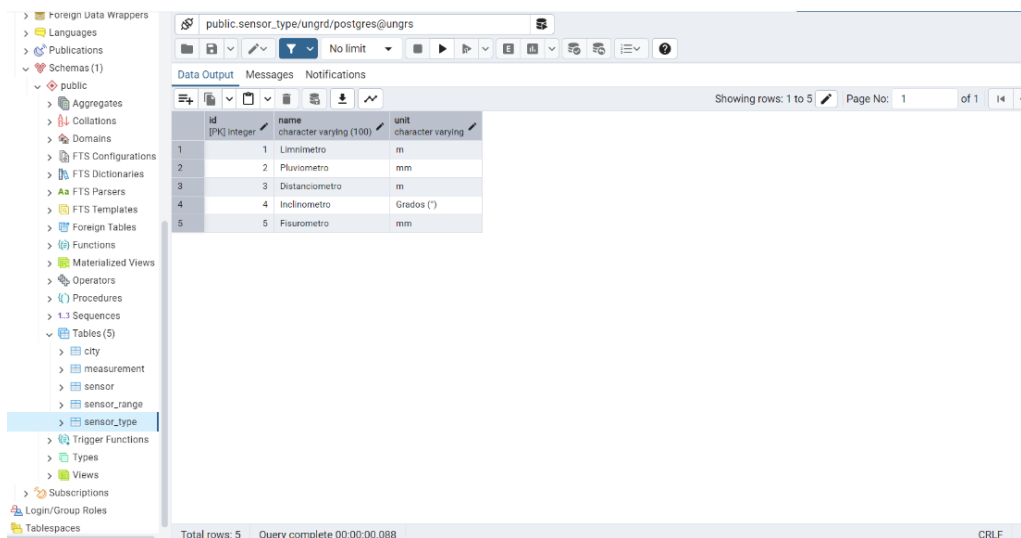
id	name	latitude	longitude
[PK] integer	character varying (100)	double precision	double precision
1	Quibdo	5.6947	-76.6611
2	Timbiquí	2.7757	-77.6582
3	Villa Rica	3.2361	-76.4667
4	Soacha	4.5794	-74.2167
5	Popayan	2.43823	-76.61316

Nota. La figura muestra la estructura de la tabla *City* implementada en PostgreSQL, la cual almacena la información básica de las ciudades donde se ubican los sensores del sistema, incluyendo el identificador único, nombre del municipio y sus coordenadas geográficas (latitud y longitud). Esta tabla permite la georreferenciación de los puntos de monitoreo dentro del Sistema de Alerta Temprana. Fuente: *Autoría propia*.

SensorType. Registra los tipos de sensores utilizados (pluviómetro, limnómetro, distanciómetro, inclinómetro y fisurómetro) (figura 2).

Figura 2

Tabla tipo de Sensor SQL



id	name	unit
1	Limnómetro	m
2	Pluviómetro	mm
3	Distanciómetro	m
4	Inclinómetro	Grados (°)
5	Fisurómetro	mm

Nota. La figura presenta la estructura de la tabla *SensorType*, donde se registran los diferentes tipos de sensores utilizados en el sistema, junto con su identificador y unidad de medida. Esta tabla permite clasificar y estandarizar las variables ambientales monitoreadas por la plataforma.

Fuente: *Autoría propia.*

Sensor. Asocia cada sensor con su tipo y ciudad, asignándole un identificador único, define los umbrales mínimo y máximo de cada sensor para determinar condiciones de alerta (figura 3)

Figura 3

Tabla de sensor SQL

id [PK]	city_id	sensor_type_id	latitude	longitude	min	max
1	1	1	5.69483678007111	-76.66043146044638	1	15
2	2	2	2.7758489385541414	-77.6529201537484	1	100
3	3	3	3.2458525507309393	-76.46197331999123	1	10
4	4	4	4.5833132309625295	-74.2155532262853	1	10
5	5	1	5.702374573752701	-76.65805439123002	1	10
6	6	2	2.784714948315019	-77.6510545787187	1	10
7	7	3	3.238229986270055	-76.45859889642415	1	10
8	8	4	2.4585989290724657	-74.21520013177795	1	10
9	9	1	3.5697851668482972	-76.65331426603814	1	10
10	10	2	2.777724950369832	-77.65393625059032	1	10
11	11	3	3.2455503765987315	-76.46456711118851	1	10
12	12	4	4.579550691342639	-74.20953394808701	1	10
13	13	1	5.701106268268699	-76.65738863951813	1	10
14	14	2	4.27822100367089413	-77.64907407588882	1	10
15	15	3	3.243127070510202	-76.45989355670694	1	10
16	16	4	4.587984589364804	-74.21180885555037	1	10
17	17	1	5.696014180092554	-76.6553202213519	1	10
18	18	2	5.2776345904245283	-77.65174900266261	1	10
19	19	3	3.239817944932226	-76.46084946646998	1	10
20	20	4	4.580193941032965	-74.21326234873669	1	10
21	21	5	5.69483678007111	-76.66043146044638	1	20

Nota. La figura ilustra la estructura de la tabla *Sensor*, en la cual se asocia cada sensor con su tipo y ciudad correspondiente, además de almacenar su ubicación geográfica y los rangos mínimo y máximo de medición definidos para la generación de alertas. Fuente: *Autoría propia.*

Measurement. Almacena las mediciones de las variables captadas por los sensores (figura 4).

Figura 4

Tabla de medidas SQL

id [PK] integer	sensor_id integer	value double precision	timestamp timestamp without time zone
1	6420	16	8 2025-09-10 12:51:32.071815
2	6421	7	2 2025-09-10 12:51:32.070813
3	6422	2	7 2025-09-10 12:51:32.076823
4	6423	8	8 2025-09-10 12:51:32.073814
5	6424	19	7 2025-09-10 12:51:32.070813
6	6425	10	10 2025-09-10 12:51:32.073814
7	6426	3	5 2025-09-10 12:51:32.10106
8	6427	12	2 2025-09-10 12:51:32.082057
9	6428	17	10 2025-09-10 12:51:32.076823
10	6429	16	5 2025-09-10 12:51:32.103059
11	6430	20	14 2025-09-10 12:51:32.098059
12	6431	2	3 2025-09-10 12:51:32.087058
13	6432	15	9 2025-09-10 12:51:32.075817
14	6433	12	7 2025-09-10 12:51:32.099059
15	6434	3	11 2025-09-10 12:51:32.10106
16	6435	19	15 2025-09-10 12:51:32.090059
17	6436	10	14 2025-09-10 12:51:32.105058
18	6437	11	8 2025-09-10 12:51:32.122598
19	6438	19	9 2025-09-10 12:51:32.091058
20	6439	9	15 2025-09-10 12:51:32.234232
21	6440	9	4 2025-09-10 12:51:32.128598

Total rows: 5608 Query complete 00:00:00.161

Nota. La figura muestra la estructura de la tabla *Measurement*, encargada de almacenar las mediciones captadas por los sensores, incluyendo el identificador del sensor, el valor registrado y la estampa de tiempo. Esta información es fundamental para el análisis histórico y la detección de condiciones de riesgo. Fuente: *Autoría propia*.

El uso de pgAdmin permitió crear, visualizar y verificar la consistencia de las tablas mediante consultas SQL. Este diseño relacional asegura la integridad y trazabilidad de la información, además de permitir la consulta en tiempo real por parte del sistema de monitoreo. La estructura también facilita el análisis histórico de los datos y la generación de alertas ante valores críticos, contribuyendo así a la operatividad del Sistema de Alerta Temprana (DataSunrise, 2024).

Frontend (Interfaz Web)

La interfaz web fue desarrollada con el framework Flutter, utilizando el lenguaje Dart

dentro del entorno de desarrollo Visual Studio Code. Flutter fue seleccionado por su capacidad para generar aplicaciones multiplataforma de alto rendimiento, su diseño adaptable y su excelente integración con servicios web (García-Solís, 2025).

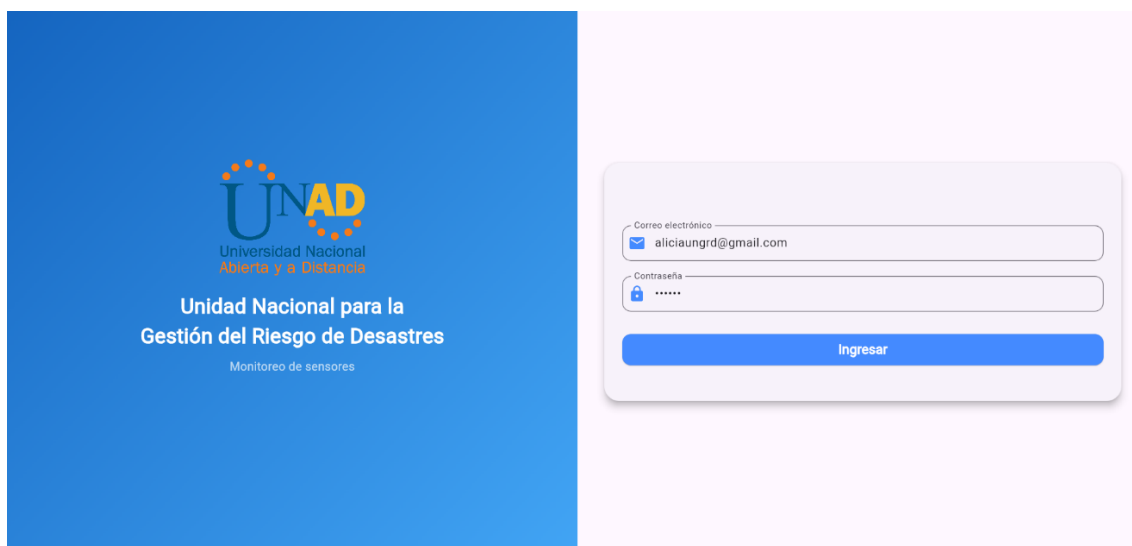
Este framework facilita la creación de interfaces interactivas mediante componentes llamados *widjets*, que permiten un desarrollo rápido y organizado. Además, Flutter ofrece una conexión eficiente con las API REST implementadas en el backend desarrollado con Kotlin/Ktor, lo cual posibilita la actualización dinámica de la información y la comunicación fluida entre las capas del sistema.

Los módulos principales que componen la interfaz web son los siguientes:

Inicio de Sesión. Permite a los usuarios acceder al sistema mediante la validación de credenciales, gestionando la autenticación y garantizando la seguridad de la información (figura 5).

Figura 5

Pantalla Inicio de Sesión



Nota. La figura corresponde a la pantalla de inicio de sesión de la plataforma web, donde los usuarios ingresan sus credenciales para acceder al sistema, garantizando la autenticación y la

seguridad de la información. Fuente: *Autoría propia*.

Página de Inicio. presenta un mensaje de bienvenida y la información general de la plataforma, destacando su enfoque sostenible, el monitoreo en tiempo real y la confiabilidad de los datos (figura 6).

Figura 6

Pantalla Página de inicio

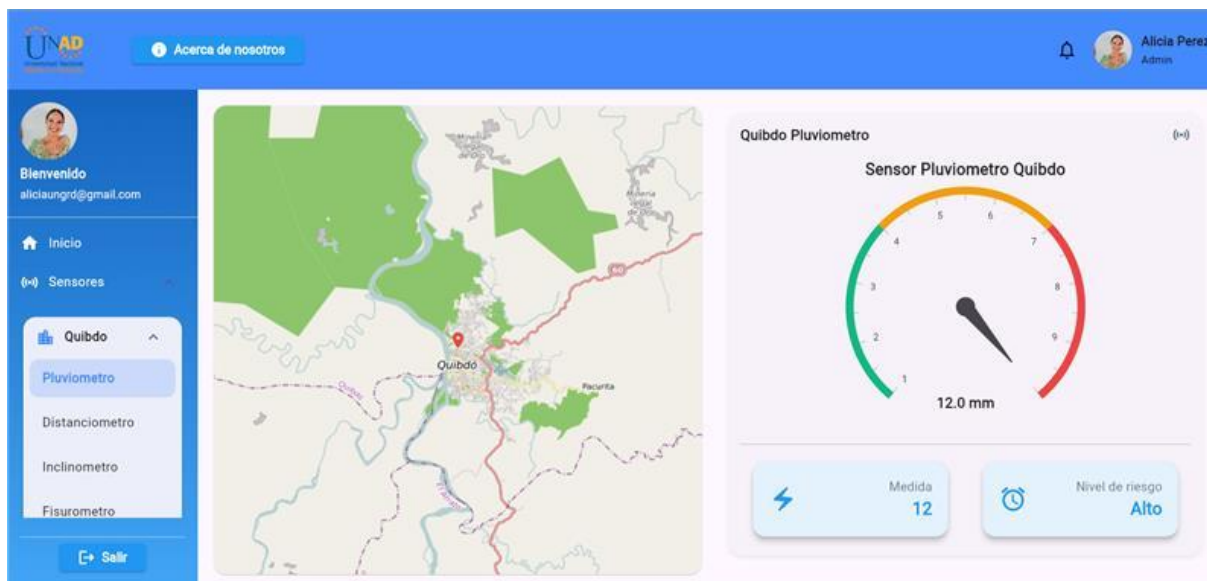


Nota. La figura presenta la pantalla principal de la plataforma web, en la cual se visualiza información general del sistema y se brinda acceso a los diferentes módulos de monitoreo, visualización y gestión de datos. Fuente: *Autoría propia*.

Visualización de Sensores. Se muestran, por ciudad, los valores captados por los diferentes sensores (pluviómetro, limnómetro, distanciómetro, inclinómetro y fisurómetro), junto con su respectivo nivel de riesgo ambiental. Además, se visualiza en el mapa la ubicación geográfica de cada sensor (figura 7.)

Figura 7

Pantalla de visualización de sensores

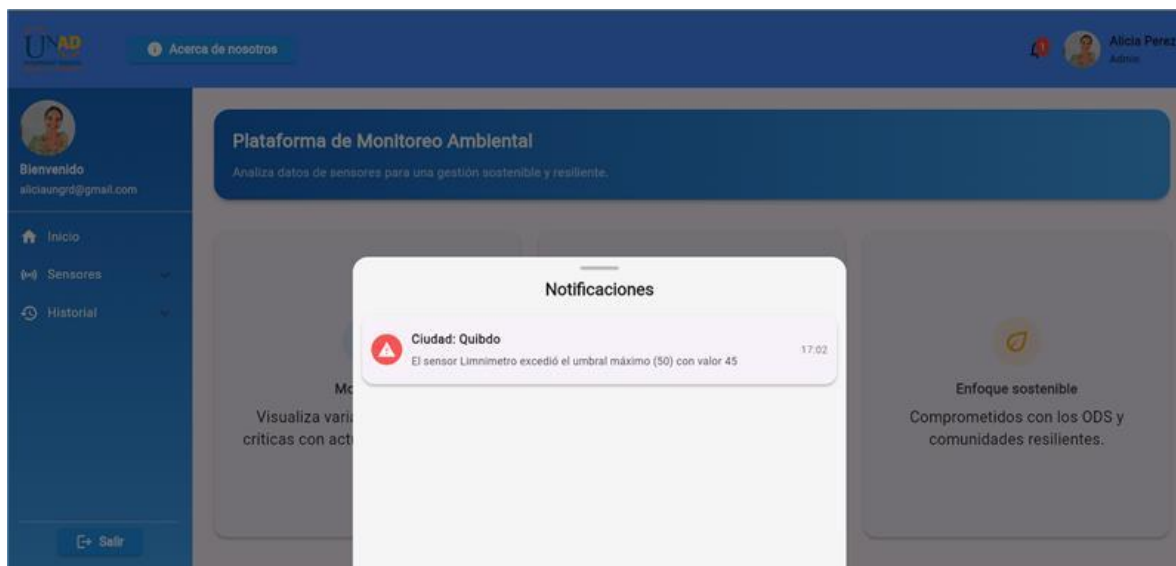


Nota. La figura muestra el módulo de visualización de sensores, donde se presentan los valores registrados por los sensores ambientales asociados a una ciudad específica, junto con su ubicación geográfica y nivel de riesgo correspondiente. Fuente: *Autoría propia.*

Visualización de Alarmas. Presenta las alertas generadas en función de los umbrales establecidos para cada sensor, permitiendo la identificación oportuna de eventos críticos (figura 8).

Figura 8

Pantalla de visualización de Alarmas



Nota. La figura ilustra el módulo de alertas del sistema, en el cual se notifican los eventos críticos generados cuando los valores de los sensores superan los umbrales establecidos, facilitando la toma de decisiones oportunas. Fuente: *Autoría propia.*

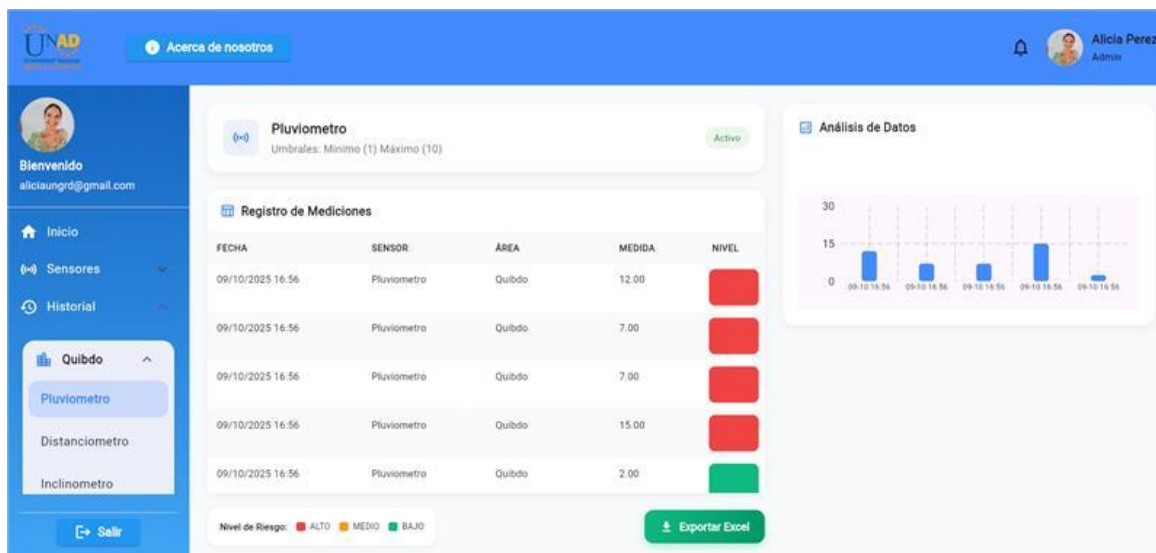
Histórico de Medidas de Sensores y Descarga

Permite consultar los registros históricos de las mediciones captadas por los diferentes sensores, filtrarlas por rango de fechas y descargar los resultados en formato .csv para su análisis posterior.

En la parte superior de la interfaz se visualizan los umbrales de medición del sensor, mostrando su valor mínimo y máximo permitidos. Además, se presentan las gráficas de análisis de datos que muestran la variación de las mediciones en el tiempo, y en la parte inferior se incluye un índice de riesgo ambiental representado por colores: rojo (alto), naranja (medio) y verde (bajo), facilitando la interpretación visual del nivel de riesgo registrado (figura 9).

Figura 9

Pantalla de Histórico de medidas



Nota. La figura presenta el módulo de histórico de mediciones, que permite consultar registros anteriores, visualizar gráficas de comportamiento en el tiempo y descargar los datos para análisis posterior, apoyando el seguimiento de variables ambientales. Fuente: *Autoría propia.*

Fuente. Autoría propia, Se visualiza el menú del historial, en donde se puede apreciar diferentes secciones como: Umbral del sensor, Últimos 6 registros del sensor, Diagrama de barras, Descarga de datos.

Cada módulo fue diseñado siguiendo los diagramas de flujo elaborados durante la fase de diseño (inicio, sensores, configuración, historial y alarmas), garantizando una navegación intuitiva y una experiencia de usuario amigable. Además, la estructura modular de Flutter permite que la plataforma sea escalable, facilitando la incorporación de nuevos sensores o funcionalidades en el futuro (García-Solís, 2025).

Integración del Sistema

La integración final del sistema se realizó interconectando los tres componentes principales: el backend, la base de datos y el frontend. El servidor desarrollado en Ktor actúa

como núcleo central, gestionando las solicitudes provenientes de la interfaz y ejecutando consultas a la base de datos PostgreSQL a través de los repositorios implementados.

El flujo de información se ejecuta en tiempo real, permitiendo registrar, procesar y visualizar las mediciones obtenidas por los sensores de forma continua. Esta integración asegura la funcionalidad completa del sistema dentro del contexto de los Sistemas de Alerta Temprana (SAT), permitiendo identificar y analizar condiciones de riesgo en diferentes regiones del país (Higuera & Quevedo, 2015).

El uso combinado de tecnologías modernas, entornos profesionales y herramientas de código abierto garantiza la sostenibilidad del sistema, su capacidad de expansión y la posibilidad de ser replicado en otros contextos territoriales o educativos.

Pruebas y Validación

La fase de pruebas y validación tuvo como objetivo comprobar el correcto funcionamiento de la arquitectura cliente–servidor propuesto, verificando la interoperabilidad de los componentes, la eficiencia del procesamiento y la estabilidad del sistema ante múltiples peticiones concurrentes.

Pruebas Funcionales

Para validar el flujo de comunicación, se desarrolló una aplicación móvil Android en kotlin/jetpack (Figura 10) que permite seleccionar la ciudad, el tipo de sensor y registrar manualmente valores de medición. La información se envía en formato JSON al endpoint /measurement de la API implementada en Kotlin/Ktor, donde se procesa y almacena en la base de datos PostgreSQL/PostGIS.

Figura 10

Pantalla Aplicación móvil Android



Nota. La figura muestra la interfaz de la aplicación móvil desarrollada en Android, utilizada para el envío manual y simulado de datos de sensores hacia la API REST, con el fin de validar la integración y el flujo de información del sistema. Fuente: *Autoría propia.*

Simultáneamente, se creó un script en Python (archivo ungrd.py), que genera lecturas simuladas desde sensores ambientales virtuales y las envía periódicamente al mismo servicio web. Este procedimiento permitió evaluar la capacidad del sistema para recibir datos desde diferentes medios y confirmar la correcta integración entre los módulos del backend, la base de datos y la interfaz web.

Los resultados mostraron una recepción estable de los datos provenientes tanto de la aplicación móvil como del simulador Python, sin pérdidas de paquetes ni errores de codificación. Las mediciones fueron visualizadas en tiempo real en el dashboard, confirmando la sincronización entre las capas cliente–servidor–base de datos.

Pruebas de Rendimiento con Apache JMeter

Con el propósito de medir el desempeño del servidor y la capacidad de respuesta de la API, se realizaron pruebas de carga utilizando Apache JMeter 5.6.3. Se configuró un plan de pruebas que ejecutó 2000 solicitudes HTTP POST al endpoint /measurement, simulando el envío concurrente de datos desde múltiples sensores (Figura. 11).

Figura 11

Prueba con Apache Jmeter

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	11:55:47.222	Prueba 1-1	envio de me...	150	Success	342	364	150	3
2	11:55:47.228	Prueba 1-5	envio de me...	179	Success	343	365	178	1
3	11:55:47.230	Prueba 1-6	envio de me...	250	Success	343	365	250	1
4	11:55:47.244	Prueba 1-21	envio de me...	396	Success	343	365	396	2
5	11:55:47.234	Prueba 1-11	envio de me...	409	Success	343	365	409	1
6	11:55:47.240	Prueba 1-15	envio de me...	466	Success	343	365	466	1
7	11:55:47.230	Prueba 1-7	envio de me...	479	Success	342	364	479	1
8	11:55:47.232	Prueba 1-9	envio de me...	495	Success	342	364	495	0
9	11:55:47.228	Prueba 1-4	envio de me...	528	Success	343	365	528	1
10	11:55:47.241	Prueba 1-16	envio de me...	531	Success	343	365	531	1
11	11:55:47.225	Prueba 1-3	envio de me...	578	Success	342	364	578	0
12	11:55:47.241	Prueba 1-18	envio de me...	573	Success	343	365	573	1
13	11:55:47.249	Prueba 1-23	envio de me...	578	Success	343	365	578	2
14	11:55:47.250	Prueba 1-27	envio de me...	597	Success	343	365	597	1
15	11:55:47.251	Prueba 1-28	envio de me...	634	Success	343	365	634	1
16	11:55:47.245	Prueba 1-19	envio de me...	646	Success	342	364	646	1
17	11:55:47.231	Prueba 1-8	envio de me...	660	Success	342	364	660	1
18	11:55:47.259	Prueba 1-29	envio de me...	634	Success	342	364	634	1
19	11:55:47.224	Prueba 1-2	envio de me...	670	Success	342	364	670	1
20	11:55:47.234	Prueba 1-10	envio de me...	660	Success	343	365	660	1

Nota. La figura presenta la configuración y ejecución de las pruebas de carga realizadas con Apache JMeter, en las cuales se simularon múltiples solicitudes concurrentes al servidor para evaluar el rendimiento de la API REST. Fuente: *Autoría propia.*

El reporte resumido (Figura. 12) evidenció los siguientes resultados promedio:

Número total de muestras: 2000

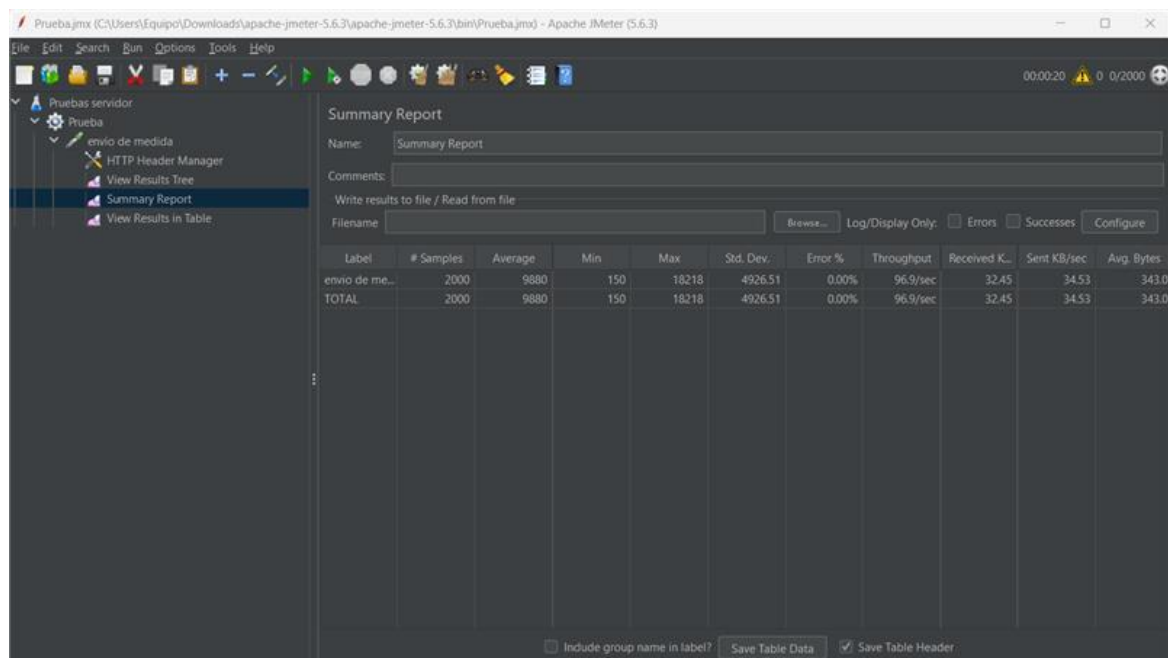
Latencia promedio: 9.88 s

Latencia mínima: 150 ms

Latencia máxima: 18.2 s

Tasa de éxito: 96.9 %

Tasa de transferencia (Throughput): 0.005 req/s

Figura 12*Resultados Apache Jmeter*

Nota. La figura muestra el reporte de resultados obtenido tras las pruebas de carga con Apache JMeter, evidenciando métricas como latencia, tasa de éxito y número de solicitudes procesadas, utilizadas para validar la estabilidad del sistema. Fuente: *Autoría propia.*

Estos resultados demostraron que la API mantiene estabilidad bajo carga moderada, procesando múltiples peticiones sin interrupciones del servicio ni pérdidas de conexión. La dispersión observada en los tiempos de respuesta se atribuye a la configuración local del entorno de pruebas y al uso de un servidor no dedicado.

Validación Integral del Sistema

Tras integrar los módulos de la arquitectura (aplicación Android, API REST, base de datos y dashboard) se verificó la consistencia del flujo de información en tiempo real. Las pruebas confirmaron que el sistema:

Recibe y almacena mediciones desde distintas fuentes (móvil y simulador).

Actualiza los valores en el panel de monitoreo sin retrasos perceptibles.

Genera alertas automáticas al superar umbrales críticos definidos por sensor.

Mantiene la integridad de los datos dentro del modelo relacional PostgreSQL/PostGIS.

La validación integral evidencia que la plataforma es funcional, escalable y tecnológicamente viable para fortalecer los Sistemas de Alerta Temprana (SAT) de la UNGRD, permitiendo futuras ampliaciones a entornos productivos reales.

Evaluación de Usabilidad de la Plataforma Web

Con el propósito de evaluar la usabilidad, accesibilidad y facilidad de navegación de la plataforma web desarrollada, se aplicó una encuesta a un grupo de 10 participantes pertenecientes al Centro de Control de la Empresa de Energía del Departamento del Cauca.

El grupo estuvo conformado por 2 ingenieros jefes, 4 ingenieros operadores y 4 auxiliares técnicos, todos con amplia experiencia en el manejo de sistemas SCADA (Supervisory Control and Data Acquisition).

Aunque los participantes no trabajan directamente en la gestión de riesgo o atención de emergencias, su experiencia en plataformas de monitoreo en tiempo real resulta altamente pertinente, dado que las interfaces de los sistemas SCADA comparten similitudes funcionales y visuales con la página desarrollada en este proyecto. Por esta razón, se seleccionó este grupo como base de la evaluación, considerando que su retroalimentación aporta una visión técnica sólida y una perspectiva crítica sobre la operatividad de sistemas de alerta y visualización de datos.

Adicionalmente, con el propósito de evaluar la facilidad de uso y la comprensión del sistema por parte de usuarios no especializados, se aplicó la misma encuesta a un segundo grupo conformado por 10 personas sin conocimientos previos en sistemas de alerta temprana. Este

grupo estuvo integrado por habitantes de comunidades ubicadas en zonas de riesgo, cercanas a ríos o áreas propensas a deslizamientos en el municipio de Popayán departamento del Cauca, seleccionados con el fin de validar la intuitividad, accesibilidad y claridad general de la plataforma web.

Durante la prueba, los usuarios interactuaron con los principales módulos de la plataforma (inicio de sesión, visualización de sensores, alertas e historial de mediciones) y posteriormente diligenciaron un breve cuestionario de valoración. El instrumento permitió identificar fortalezas y aspectos susceptibles de mejora en cuanto a diseño, estructura visual, velocidad de carga, y comprensión de los indicadores de riesgo.

El instrumento estuvo compuesto por 15 ítems distribuidos en cuatro dimensiones: Experiencia con la plataforma, Eficiencia y rendimiento, Utilidad y confiabilidad, y Satisfacción general.

Las respuestas se calificaron en una escala tipo Likert de 1 (totalmente en desacuerdo) a 5 (totalmente de acuerdo), incluyendo además dos preguntas abiertas sobre mejoras y funcionalidades sugeridas (tabla 2).

Tabla 2

Resultado promedio de la encuesta de usabilidad (n=10)

Dimensión Evaluada	Grupo A (con conocimiento SCADA)	Grupo B (sin conocimiento técnico)	Promedio General
Experiencia con la Plataforma	4.9	4.6	4.7
Eficiencia y Rendimiento	4.9	4.6	4.7

Utilidad y	4.9	4.7	4.8
Confiabilidad			
Satisfacción General	4.9	4.6	4.7

Nota. La tabla presenta los resultados promedio obtenidos a partir de la encuesta de usabilidad aplicada a los usuarios, diferenciando entre participantes con conocimiento técnico y sin conocimiento previo, lo que permitió evaluar la percepción general sobre la eficiencia, utilidad y satisfacción con la plataforma. Fuente: *Autoría propia.*

Recomendaciones de Mejora

En cuanto al diseño e interfaz, los participantes con experiencia en sistemas recomendaron implementar un deslizable o carrusel en la página principal, mejorar la paleta de colores optando por tonos más oscuros o sofisticados, y hacer la visualización más dinámica y atractiva. Por su parte, los usuarios sin experiencia sugirieron incluir una breve explicación inicial que permita comprender el propósito de la plataforma desde el primer acceso, así como un diseño más intuitivo que reduzca la posibilidad de confusión al momento del registro o la consulta de datos. Ambos grupos coincidieron en que la interfaz resulta funcional, pero podría beneficiarse de una presentación más moderna y orientada a la comprensión inmediata del usuario.

En el componente de funcionalidad operativa, los usuarios con experiencia resaltaron la importancia de incluir roles diferenciados y asignaciones específicas dentro de la plataforma, especialmente para los monitores o responsables de cada comunidad o ciudad. También propusieron incorporar notificaciones emergentes que alerten cuando un sensor supere los límites establecidos, y habilitar la opción de corregir o eliminar datos erróneos registrados desde la aplicación. Los usuarios sin experiencia, en cambio, manifestaron el deseo de que la plataforma

esté abierta al público para su visualización general, promoviendo la transparencia y el acceso libre a la información ambiental.

En cuanto a la gestión y almacenamiento de la información, los participantes con experiencia hicieron énfasis en la necesidad de garantizar la agilidad en la consulta y la estabilidad del sistema a largo plazo, especialmente cuando el volumen de datos aumente. Así mismo, señalaron la importancia de prever posibles fallos de conectividad en zonas rurales, recomendando analizar cómo se guarda la información cuando no hay señal o conexión a internet. Los usuarios sin experiencia reforzaron esta idea sugiriendo la inclusión de un diccionario de variables que facilite la comprensión de los datos y el manejo adecuado de los términos técnicos.

Recomendaciones en Futuras Versiones

En primer lugar, varios encuestados resaltaron la importancia de fortalecer los mecanismos de visualización e interpretación de información, proponiendo la implementación automática de gráficos estadísticos que representen los datos históricos obtenidos por los sensores, con el fin de identificar tendencias, temporadas críticas o posibles comportamientos anómalos de las variables monitoreadas. En esta misma línea, se sugirió la incorporación de mapas de calor y despliegues generales de monitoreo por ciudad o departamento, que permitan un análisis territorial más amplio y visual de las condiciones ambientales registradas.

Otro grupo de participantes destacó la necesidad de mejorar la gestión y el acceso a los datos, recomendando la posibilidad de exportar la información en forma de reportes y garantizar un almacenamiento más prolongado en la base de datos, con el propósito de conservar registros históricos útiles para estudios comparativos o investigaciones futuras. Asimismo, se propuso

integrar una configuración de perfiles de acceso, que permita diferenciar roles y niveles de usuario, fortaleciendo la seguridad y el control interno del sistema.

En cuanto a la interacción y usabilidad, los encuestados sugirieron añadir funciones de retroceso o acceso rápido dentro de la interfaz, así como la creación de espacios de interacción entre usuarios o comunidades que aportan información a la plataforma. También se planteó la inclusión de novedades visuales que hagan más atractiva la experiencia de uso, y opciones orientadas a públicos específicos, como funciones diseñadas para niños, con el fin de fomentar la educación ambiental de manera lúdica y participativa.

Dentro del ámbito de alertas y monitoreo en tiempo real, algunos usuarios recomendaron integrar notificaciones con sonido para los estados de riesgo, así como la aparición de alertas en burbujas que informen de manera inmediata cualquier cambio crítico en las variables medidas. De igual manera, se sugirió permitir la visualización pública del monitoreo ambiental, promoviendo la transparencia y la participación ciudadana en la vigilancia del entorno. Otros participantes añadieron que sería valioso incluir herramientas que muestren las zonas potencialmente afectadas por fenómenos naturales, mejorando la capacidad preventiva y la toma de decisiones.

Finalmente, una recomendación relevante fue la modernización del sistema de sensores, proponiendo que sean digitales y no manuales, lo que permitiría una mayor precisión, automatización y eficiencia en la recolección de datos.

Evaluación Final

El desarrollo del proyecto permitió cumplir los objetivos propuestos, logrando la implementación de una plataforma tecnológica funcional, capaz de integrar de manera eficiente la recolección, procesamiento y visualización de datos en tiempo real. A través del uso de tecnologías modernas como Kotlin/Ktor para el backend, PostgreSQL/PostGIS para la base de datos y Flutter/Dart para la interfaz web, se consiguió un sistema estable, modular y con bajo consumo de recursos.

Durante la fase de validación, las pruebas funcionales y de rendimiento demostraron la eficiencia del sistema frente a múltiples solicitudes simultáneas, alcanzando una tasa de éxito del 96,9 % y una latencia promedio inferior a los 10 segundos. Estos resultados confirman la confiabilidad de la arquitectura cliente-servidor propuesto y su potencial para ser aplicada en diferentes contextos tecnológicos.

La integración de herramientas abiertas y multiplataforma garantiza la sostenibilidad del sistema, al mismo tiempo que facilita su adaptación a futuras mejoras o ampliaciones. De esta manera, el proyecto se consolida como una propuesta académica sólida que contribuye al fortalecimiento de soluciones tecnológicas accesibles y escalables, útiles para instituciones y comunidades que requieran sistemas de gestión de datos eficientes y seguros.

Resultados

Los ensayos experimentales permitieron obtener indicadores cuantitativos del rendimiento del sistema. Mediante el uso de Apache JMeter 5.6.3, se ejecutaron 2000 solicitudes POST simultáneas al endpoint /measurement, simulando múltiples sensores activos. (tabla 3)

Tabla 3

Resultados Obtenidos

Parámetro	Valor Promedio	Valor Mínimo	Valor Máximo	Tasa de Exito
Tiempo de Respuesta (ms)	9880	150	18218	96,9 %
Throughput (req/s)	0,005			
Bytes Recibidos	32,45 KB/s			
Bytes Enviados	34,53 KB/s			

Nota. La tabla resume los principales resultados alcanzados durante el desarrollo y validación del sistema, destacando los indicadores de desempeño, estabilidad y funcionalidad obtenidos a partir de las pruebas realizadas. Fuente: *Autoría propia*.

Durante la simulación, la API procesó todas las solicitudes sin interrupciones críticas. El servidor mantuvo estabilidad, y la base de datos gestionó correctamente la inserción de registros sin pérdidas. Las gráficas de latencia mostraron una distribución controlada, sin sobrecarga en la cola de peticiones.

Las pruebas combinadas (aplicación Android, simulador Python y JMeter) confirmaron que la arquitectura puede recibir datos de distintos orígenes, procesarlos en tiempo real y almacenarlos de forma segura, garantizando la integridad del flujo de información.

Discusión

Los resultados evidencian que la arquitectura propuesta cumple con los principios de diseño de sistemas distribuidos y resilientes aplicados a entornos de monitoreo ambiental. La utilización de una API REST en Kotlin/Ktor permitió una gestión eficiente de concurrencia gracias a las corrutinas nativas, lo cual redujo los bloqueos y mejoró la escalabilidad.

El uso de Apache JMeter como herramienta de estrés permitió identificar comportamientos reales del servidor frente a cargas múltiples, mostrando que, aun con recursos limitados, el sistema mantiene un rendimiento aceptable. Esto sugiere que la solución podría operar en entornos rurales o institucionales con infraestructura básica sin comprometer la disponibilidad del servicio.

El simulador Python demostró ser una herramienta útil para pruebas de integración continua, ya que permite generar datos sintéticos que simulan sensores físicos sin necesidad de hardware adicional. Asimismo, la aplicación Android facilita la participación de usuarios finales, generando un entorno colaborativo y ampliando el alcance del sistema.

En términos de sostenibilidad tecnológica, el uso de componentes de código abierto y herramientas multiplataforma respalda el principio de accesibilidad y replicabilidad que busca la UNAD, permitiendo que otras instituciones adopten o adapten la solución con bajo costo y mínima curva de aprendizaje.

Conclusión

El sistema desarrollado constituye una plataforma integral y funcional para el monitoreo y análisis en tiempo real de datos ambientales dentro de los Sistemas de Alerta Temprana (SAT). Su arquitectura modular compuesta por una aplicación Android, una API REST en Kotlin/Ktor, una base de datos PostgreSQL/PostGIS y un simulador en Python demostró ser interoperable, escalable y eficiente frente a cargas operacionales simuladas.

Las pruebas de rendimiento realizadas con Apache JMeter confirmaron la solidez de los servicios del backend, alcanzando una tasa de éxito del 96,9 % en 2000 solicitudes concurrentes y una latencia promedio inferior a 10 segundos, lo que valida la estabilidad y capacidad de respuesta del sistema bajo condiciones de estrés. El flujo continuo de información desde los sensores hasta el panel de visualización garantizó la trazabilidad e integridad de los datos en todas las capas del sistema.

Más allá de su factibilidad técnica, la solución propuesta representa un modelo replicable y de bajo costo para instituciones y comunidades rurales que buscan fortalecer sus capacidades de monitoreo ambiental y respuesta ante emergencias. Su diseño de código abierto promueve la sostenibilidad, transparencia y autonomía tecnológica, además de servir como base para futuras extensiones orientadas al aprendizaje automático, la fusión de datos IoT y la toma de decisiones adaptativa.

En síntesis, esta investigación demuestra que las arquitecturas abiertas, modulares y habilitadas geo espacialmente pueden mejorar significativamente la eficacia y resiliencia de los Sistemas de Alerta Temprana, contribuyendo al avance global en la reducción del riesgo de desastres mediante la innovación tecnológica.

Referencias Bibliográficas

- Aguirre Buenaventura, J. (2022). Gestión del riesgo y fortalecimiento de los sistemas de alerta temprana en contextos rurales. Editorial Académica Colombiana.
- Alexander, R. V. J., Derley, R. J., & Fernando, A. D. (2018). Diseño de un sistema de alerta temprana ante inundaciones, sequía e incendios forestales como medida de adaptación al cambio climático. *Revista de Ingeniería Ambiental*, 6(1), 22–31.
- Coll, M. A. (2013). Sistemas de alerta temprana (SAT) para la reducción del riesgo de inundaciones súbitas y fenómenos atmosféricos en el área metropolitana de Barranquilla. *Scientia et Technica*, 18(2), 303–308.
- DataSunrise. (2024a). Confidencialidad, integridad y disponibilidad: Ejemplos clave. Centro de Conocimiento DataSunrise. <https://www.datasunrise.com/es/centro-de-conocimiento/ejemplos-de-confidencialidad-integridad-disponibilidad/>
- DataSunrise. (2024b). PostgreSQL database security overview [Informe técnico].
- Equipos Topografía. (2024). Distanciómetros: Guía completa para medir distancias. <https://equipostopografia.com/equipostopograficos/distanciometros/distanciometros-guia-completa/>
- García-Solís, J. C. (2025). Diseño de interfaces multiplataforma con Flutter. Editorial Tecnológica.
- Hernández, M. (2024). Implementación de APIs RESTful en entornos Kotlin. *Revista Ingeniería y Tecnología*, 12(3), 45–58.
- Higuera, D. M., & Quevedo, M. C. C. (2015). Manual de procedimientos del pluviómetro ISCO 674 del laboratorio de servicios públicos. Universidad Distrital Francisco José de Caldas.

- IDEAM (Instituto de Hidrología, Meteorología y Estudios Ambientales). (2023). Reporte anual de eventos hidrometeorológicos en Colombia. <http://www.ideam.gov.co/>
- Leal, J. F. R. (2024). Estudio de caso: Revisión del uso de dashboards para la visualización de datos. *Revista de Ingeniería Aplicada*, 8(2), 45–53.
- Leal, J. F. R. (2025). Desarrollo backend moderno con Kotlin y Ktor. Editorial Universitaria.
- López-García, J. D., Carvajal-Escobar, Y., & Enciso-Arango, A. M. (2017). Sistemas de alerta temprana con enfoque participativo: un desafío para la gestión del riesgo en Colombia. *Luna Azul*, 44, 231–246.
- MDH. (2024). *Inclinómetros: Instrumentación geotécnica en Perú*.
<https://mdh.com.pe/inclinometros-instrumentacion-geotecnica-en-peru/>
- Moreno, D., Quiñones, É., & Tovar, L. C. (2014). Los sistemas de alerta temprana (SAT), una herramienta para la prevención de desastres por inundación y efectos del cambio climático. *Ciencia e Ingeniería al Día*, 9(1), 1–18.
- Peñas, V. (2017). La catástrofe de Mocoa (Colombia): un desastre anunciado previsible que podía haberse evitado. Facultad de Ciencias Agrarias, Universidad Nacional de Colombia.
- Peralta, G., De Elia, E. A., Lanfri, M. A., Porcasi, X., Frutos, N., Rotela, C., & Scavuzzo, M. (2011). Geomática aplicada a un sistema de alerta temprana. En II Congreso Argentino de Informática y Salud (CAIS 2011) (pp. 1–8). Córdoba, Argentina.
- SENAMHI (Servicio Nacional de Meteorología e Hidrología del Perú). (2008). Guía de hidrometría de aguas superficiales: Separata N.º 01-2008.
https://www.senamhi.gob.pe/pdf/aprendiendo_hidrometria.pdf

Thingtrack. (2025). Fisurómetro para monitoreo de grietas. <https://thingtrack.com/fisurometro-para-monitoreo-de-grietas/>

UNGRD (Unidad Nacional para la Gestión del Riesgo de Desastres). (2016). Guía para la implementación de sistemas de alerta temprana comunitarios (Vol. 9). Bogotá, Colombia.

Viloria, A., & Quintero, W. (2016). Gestión de información SAT Río Manzanares. Investigación e Innovación en Ingenierías, 4(2), 128–135.

Xiloj López, R. A. (2025). Modelo de base de datos para sistemas ambientales [Informe técnico]. Universidad Nacional Abierta y a Distancia (UNAD).