Universidad Nacional de Ingeniería

Dirección de Área de Conocimiento de Tecnología de la Información y Comunicación

“Diseño de un sistema de monitoreo de ubicación en tiempo real de visitantes usando la plataforma Firebase en la UNI - RUSB”

Trabajo Monográfico para optar al título de Ingeniero Electrónico

Elaborado por:

Br. Mario Moisés López Escoto
Carnet: 2006-24136

Br. Wandy Yubran Jarquín Vega
Carnet: 2016-0862U

Br. Luis Enrique Robleto Rodríguez
Carnet: 2016-0420U

Tutor:

Msc. Saúl Otoniel Núñez Zeledón

06 de mayo de 2026
Managua, Nicaragua

DEDICATORIA

“Dedico este trabajo a Dios, por las bendiciones recibidas y por permitirme culminar esta etapa con éxito.

A mi padre, por su ejemplo de perseverancia y por ser mi pilar fundamental en este proceso. Gracias por cada palabra de aliento.

A mi mamá, que hoy me cuida desde la eternidad. Tu ausencia duele, pero tu recuerdo me da las fuerzas necesarias para seguir adelante. Gracias por los valores que sembraste en mí; espero que este triunfo la haga sentir orgullosa, aunque ya no esté físicamente conmigo”

Mario Moisés López Escoto

DEDICATORIA

“Dedico el presente trabajo Dios, por ser mi guía constante, por darme la fortaleza en los momentos más difíciles y por iluminar cada paso de este camino. Sin su voluntad, nada de esto habría sido posible.

A mi madre, Maira Rodríguez, que hoy me acompaña desde el cielo, pero vive eternamente en mi corazón. Este logro es también tuyo, porque en cada esfuerzo estuvo tu amor, tus enseñanzas y el recuerdo de tu voz impulsándome a no rendirme. Espero que donde estés, te sientas orgullosa de mí.

A mi padre, por su apoyo, su ejemplo y por estar presente a lo largo de mi vida, brindándome las herramientas y valores necesarios para seguir adelante.

A mi amada Melissa, por su amor incondicional, su paciencia y su apoyo en cada etapa de este proceso. Gracias por creer en mí incluso cuando yo dudaba, por ser mi refugio y mi motivación para seguir adelante.

A mis hermanos, por su compañía, sus palabras de aliento y por ser parte fundamental de mi vida. Este logro también les pertenece.”

Luis Enrique Robleto Rodríguez

DEDICATORIA

“Dedico el presente trabajo, en primer lugar, a Dios, por brindarme sabiduría, paz y conocimiento a lo largo de este proceso, guiando cada uno de mis pasos y permitiéndome alcanzar esta meta tan importante.

A mis padres, Humberto Jarquín y Eveling Vega, quienes han sido el pilar fundamental en mi vida. Gracias por su amor incondicional, sus sacrificios y sus enseñanzas, que han sido clave en mi formación personal y profesional.

Finalmente, dedico este logro a mi esposa, Anielka Valle, quien se ha convertido en una pieza esencial en esta nueva etapa de mi vida. Su apoyo, comprensión y compañía han sido fundamentales para seguir adelante y no rendirme en el camino.”

Wandy Yubran Jarquín Vega

RESUMEN

El presente proyecto desarrolla un sistema de monitoreo de visitantes en tiempo real para el campus UNI–RUSB, utilizando tecnologías modernas como GPS y la plataforma Firebase. La solución propuesta responde a la necesidad de mejorar el control de acceso y la supervisión de personas dentro de un entorno universitario dinámico.

El sistema se basa en una aplicación web progresiva (PWA) que permite a los visitantes registrarse mediante un código QR, capturar su ubicación geográfica desde dispositivos móviles y transmitirla en tiempo real hacia una base de datos en la nube (Firestore). Esto permite visualizar trayectorias y ubicaciones en un panel de monitoreo accesible para el personal autorizado.

Metodológicamente, el proyecto se desarrolló en tres etapas: diseño y configuración, pruebas funcionales y evaluación de viabilidad. Se validó el funcionamiento mediante diferentes escenarios, confirmando una actualización en tiempo real eficiente y una correcta trazabilidad de los visitantes.

Los resultados evidencian que el sistema es viable, escalable y de bajo costo, al no requerir hardware adicional. Además, mejora significativamente la seguridad institucional y optimiza los procesos de supervisión.

En conclusión, el sistema representa una alternativa tecnológica moderna que fortalece el control interno mediante herramientas digitales accesibles, contribuyendo a la transformación digital de la institución.

Contenido

1. Introducción 1
2. Antecedentes 2
3. Objetivos 4
   3.1. General 4
   3.2. Específicos 4
4. Justificación 5
5. Marco teórico 6
   5.1. Sistemas de monitoreo 6
   5.2. GPS 6
   5.3. Plataforma Firebase como backend administrado 7
      5.3.1. Firestore (NoSQL, sincronización en tiempo real) 7
      5.3.2. Firebase Hosting (despliegue por HTTPS y CDN) 7
   5.4. Tecnologías web para la implementación (HTML, CSS y JavaScript) 8
      5.4.1. HTML (estructura de la interfaz) 8
      5.4.2. CSS (presentación responsiva y usabilidad móvil) 8
      5.4.3. JavaScript (lógica, geolocalización y actualización dinámica) 9
   5.5. Aplicaciones Web Progresivas (PWA) 10
   5.6. Identificación de visitas mediante códigos QR 10
   5.7. Visualización cartográfica y trayectorias en la web – Leaflet 11
   5.8. Marco normativo sobre el uso de tecnologías de geolocalización 12
6. Metodología del proyecto 14
   6.1. Enfoque metodológico 14
   6.2. Estructura de la metodología por etapas 15
      6.2.1. Etapa 1: Diseño y configuración Inicial 15
      6.2.2. Etapa 2: Simulación funcional y pruebas del prototipo 16
      6.2.3. Etapa 3: Evaluación de viabilidad 17
   6.3. Viabilidad técnica 17
7. Desarrollo del proyecto 21

7.1. Estructura del proyecto 22
7.1.1. Configurar Firebase Hosting 24
7.1.2. Instalación de Firebase CLI 24
7.1.3. Iniciar sesión en Firebase 25
7.1.4. Inicializar Firebase Hosting dentro del proyecto 25
7.1.5. Crear los archivos del proyecto 27
7.1.6. Descripción de los archivos 27
7.1.7. Código principal de la aplicación 29
7.2. Generación del código QR para el acceso del visitante 37
7.3. Presentación y análisis de resultados 38
7.3.1. Preguntas guía 38
7.3.2. Valores obtenidos 38
8. Conclusiones 41
9. Recomendaciones 42
10. Referencias Bibliográficas 43
11. Anexos 44

Índice de figuras

Figura 1 Constelación de satélites expandibles de 24 ranuras, como se define en el Estándar de rendimiento de SPS. 7
Figura 2 Servicios contenidos en Firebase 8
Figura 3 Arquitectura Web con HTML, CSS y JavaScript 9
Figura 4 Arquitectura y Beneficios de las PWA 10
Figura 5 Control de Acceso con Códigos QR 11
Figura 6 Mapas y Trayectorias Web con Leaflet 12
Figura 7 Arquitectura de Geolocalización con Firebase 14
Figura 8 Diagrama de flujo de proceso 21
Figura 9 Regla de Redirección en Hosting 33
Figura 10 Visualización de interfaz vista desde un dispositivo móvil 35
Figura 11 Visualización de Panel de Monitoreo 36
Figura 12 Acceso a panel de visitante mediante QR 37
Figura 13 Datos obtenidos en prueba 1 39
Figura 14 Trayectoria vista en mapa 39
Figura 15 Fragmento de código para obtención de ubicación 45
Figura 16 Escaneo de código QR 46
Figura 17 Prueba de registro desde dispositivo móvil 46

1. Introducción

La Universidad Nacional de Ingeniería (UNI) cuenta con una población aproximada de 13,500 estudiantes en sus diferentes niveles académicos, según datos del SIESCA. En el campus UNI–RUSB convergen alrededor de 3,500 estudiantes y 400 colaboradores, incluyendo personal docente, administrativo y miembros del Cuerpo de Protección Física (CPF). Este flujo constante de personas, junto con la entrada diaria de visitantes externos, convierte al campus en un entorno altamente dinámico donde el control de acceso y la supervisión de desplazamientos se vuelve un proceso complejo y de importancia estratégica.

El Sistema de Posicionamiento Global (GPS), integrado de forma nativa en los dispositivos móviles modernos, ofrece una solución eficiente para la obtención de coordenadas geográficas en tiempo real, habilitando aplicaciones de navegación, rastreo y seguridad sin necesidad de hardware adicional. Al combinar esta tecnología con plataformas en la nube capaces de sincronizar datos de forma instantánea, es posible implementar sistemas de monitoreo más ágiles y precisos que los métodos tradicionales basados en registros manuales.

En este contexto, el presente proyecto desarrolla un sistema de monitoreo de visitantes basado únicamente en software, utilizando una aplicación web progresiva (PWA) que recopila la ubicación GPS del dispositivo móvil y la transmite en tiempo real mediante Firebase. El acceso al sistema se realiza a través de un código QR, lo que permite identificar cada visita de forma rápida y estandarizada.

La propuesta ofrece una alternativa moderna, escalable y de bajo costo para fortalecer los procesos de control interno del CPF, proporcionando una herramienta que facilita la supervisión continua, mejora la trazabilidad de los visitantes y contribuye al fortalecimiento de la seguridad institucional en el campus.

2. Antecedentes

El uso de tecnologías de geolocalización para el rastreo y seguimiento en tiempo real ha sido ampliamente abordado en investigaciones tanto a nivel nacional como internacional. Estos estudios demuestran la versatilidad del GPS en aplicaciones orientadas a movilidad, seguridad y gestión operativa, aportando bases conceptuales para el desarrollo de sistemas de monitoreo basados en dispositivos móviles y plataformas digitales.

En Nicaragua, se han desarrollado diversos trabajos que aprovechan el GPS como herramienta para la obtención y visualización de trayectorias. Entre ellos, destaca el proyecto "**Desarrollo de aplicación para teléfono Android que brinde información sobre el trayecto del transporte urbano colectivo de Managua**" (Garcia Carballo & Malespín Cabrera, 2015) el cual implementa una aplicación móvil para consultar rutas y paradas del transporte urbano utilizando los sensores GPS integrados en los teléfonos inteligentes. Este estudio evidencia la aplicabilidad del GPS móvil para la navegación y la representación de recorridos en tiempo real, sin necesidad de hardware especializado adicional.

De igual manera, el trabajo "**Diseño, desarrollo y programación de sistema GPS utilizando módulo PMB688 y microcontrolador ATMEGA328P para rastreo vehicular**" (Tapia Moraga & Aguilar Díaz, 2016) demuestra la capacidad de los sistemas de rastreo para registrar y transmitir información de localización, en este caso aplicado a flotas vehiculares. Aunque integra componentes electrónicos y módulos GPS externos, aporta un marco conceptual relevante sobre los mecanismos de captura y procesamiento de datos de geolocalización, aplicables a soluciones más modernas basadas únicamente en dispositivos móviles.

A nivel internacional, numerosas investigaciones han explorado soluciones de rastreo y monitoreo mediante tecnologías GPS integradas en dispositivos o sistemas híbridos. Un ejemplo es el proyecto "**Diseño de un equipo de rastreo satelital usando tecnologías GPS y GSM**" (Henao Melo, 2014), que propone un dispositivo capaz de transmitir su ubicación en tiempo real mediante redes móviles y satelitales. Aunque se

basa en hardware dedicado, el estudio aporta fundamentos sobre la importancia de la comunicación en tiempo real para sistemas de monitoreo.

Asimismo, el trabajo “**Prototipo de sistema de localización por GPS para salvamento marítimo**” (Raldúa Veuthey, 2012), desarrolla una red de dispositivos capaces de emitir su posición en situaciones de emergencia. Este proyecto destaca la relevancia del GPS y de las plataformas de visualización remota para el seguimiento continuo de personas en ambientes de riesgo, reforzando la pertinencia de tecnologías de localización como herramienta para la seguridad.

3. Objetivos

3.1. General

- Diseñar un sistema de monitoreo de visitantes usando la plataforma Firebase que permita el registro, almacenamiento y visualización en tiempo real de la ubicación de los visitantes en la Universidad Nacional de Ingeniería.

3.2. Específicos

- Identificar los componentes para crear un sistema de monitoreo de visitantes en el campus UNI-RUSB a través del rastreo por GPS y basado en los servicios en la nube aplicando Firebase
- Configurar Firebase como plataforma de almacenamiento y sincronización en tiempo real de los datos generados por el sistema.
- Evaluar la viabilidad técnica y operativa del sistema propuesto considerando su implementación en el campus UNI-RUSB.

4. Justificación

El presente proyecto surge ante la necesidad de mejorar los procesos de control de acceso y supervisión de visitantes dentro del campus UNI–RUSB, el cual presenta un flujo constante de personas que dificulta el seguimiento manual y reduce la eficiencia en la seguridad institucional.

En la actualidad, los métodos tradicionales de registro no permiten una trazabilidad en tiempo real ni un control preciso de los desplazamientos, lo que puede generar riesgos y limitar la toma de decisiones del personal encargado. Por ello, se propone la implementación de un sistema de monitoreo basado en tecnologías de geolocalización (GPS) y servicios en la nube mediante Firebase.

Desde el punto de vista tecnológico, el sistema aprovecha dispositivos móviles, evitando hardware adicional y reduciendo costos. Además, el uso de una aplicación web progresiva (PWA) permite una solución accesible, escalable y fácil de implementar, garantizando la transmisión de datos en tiempo real.

En el ámbito operativo, mejora la supervisión de los visitantes al permitir conocer su ubicación y trayectoria, fortaleciendo el control interno y optimizando la seguridad. Asimismo, genera registros confiables para análisis y toma de decisiones.

Finalmente, el proyecto tiene relevancia académica y tecnológica al integrar geolocalización, desarrollo web y computación en la nube, aportando una solución innovadora que contribuye a la seguridad institucional y a la transformación digital.

5. Marco teórico

5.1. Sistemas de monitoreo

Un sistema de monitoreo es un conjunto de herramientas y procedimientos destinados a observar, registrar y presentar información sobre el estado o comportamiento de una persona, proceso o fenómeno en un entorno determinado. A diferencia de los sistemas de control, cuyo objetivo es ejecutar acciones correctivas o intervenir sobre el entorno, los sistemas de monitoreo se limitan a recopilar datos y proporcionar visualización en tiempo real para facilitar la supervisión por parte de un operador humano. (REAL ACADEMIA ESPAÑOLA, 2021)

En el contexto de entornos institucionales como campus universitarios, los sistemas de monitoreo permiten conocer la ubicación y trayectoria de personas utilizando tecnologías de geolocalización, aplicaciones móviles o web, y plataformas de almacenamiento en la nube. Su uso contribuye a mejorar la seguridad, optimizar los procesos de vigilancia y proporcionar información para la toma de decisiones sin necesidad de hardware especializado.

Con el avance de los dispositivos móviles y de las plataformas cloud, los sistemas de monitoreo han evolucionado hacia soluciones completamente basadas en software, donde la captura de datos es realizada por el propio smartphone y el procesamiento se gestiona mediante aplicaciones web y bases de datos en tiempo real. Este enfoque reduce costos, aumenta la accesibilidad y permite implementar soluciones más flexibles y escalables, como el sistema desarrollado en este proyecto.

5.2. GPS

Sistema de Posicionamiento Global (GPS por sus siglas en inglés), es un sistema conformado por constelaciones de satélites ubicadas alrededor del mundo, el cual cada satélite vuela en órbita terrestre media, estos satélites son los encargados de transmitir señales de radio a los usuarios, en lo que respecta a su uso al menos en solo Estados Unidos posee 24 satélites GPS operativos en un 95% del tiempo, (GPS.GOV, 2021). El GPS se convirtió en un sistema para empresas y ciudadanos con un acceso libre de costo alguno esto decretado por el presidente de los estados unidos Ronald Reagan.



Figura 1 Constelación de satélites expandibles de 24 ranuras, como se define en el Estándar de rendimiento de SPS.

5.3. Plataforma Firebase como backend administrado

5.3.1. Firestore (NoSQL, sincronización en tiempo real)

Firestore es una base de datos NoSQL orientada a documentos y colecciones, con capacidades de sincronización en tiempo real. Para el monitoreo, permite persistir visitas y subcolecciones de ubicaciones con marca de tiempo y precisión, y suscribirse a cambios para reflejarlos de inmediato en el panel del supervisor. (docs, 2026)

5.3.2. Firebase Hosting (despliegue por HTTPS y CDN)

Firebase Hosting publica la aplicación web por HTTPS con CDN global, control de versiones y configuración sencilla. Este servicio facilita un despliegue seguro y confiable sin operar servidores propios.



Figura 2 Servicios contenidos en Firebase

5.4. Tecnologías web para la implementación (HTML, CSS y JavaScript)

5.4.1. HTML (estructura de la interfaz)

HTML define la estructura semántica de las vistas del visitante y del panel del supervisor, organizando componentes como botones de acción, contenedores de estado, tablas de visitas y el lienzo del mapa. Esta organización favorece la accesibilidad, la integración con estilos y scripts, y la mantenibilidad de la aplicación. (W3Schools, 2026)

5.4.2. CSS (presentación responsiva y usabilidad móvil)

CSS proporciona la capa de presentación para garantizar una experiencia responsiva en dispositivos móviles: tipografías legibles, contrastes adecuados, rejillas fluidas, tamaños

táctiles mínimos y jerarquías visuales claras para apoyar tareas críticas como iniciar/finalizar una visita y supervisar ubicaciones en tiempo real. (W3Schools, 2026)

5.4.3. JavaScript (lógica, geolocalización y actualización dinámica)

JavaScript implementa la lógica del cliente: solicitud de permisos de geolocalización, captura continua de coordenadas, serialización de datos, envío a la base de datos en tiempo real y actualización del mapa y de la interfaz sin recargar la página (W3Schools, 2026). También gestiona el ciclo de vida de una visita (inicio, estado, cierre)



Figura 3 Arquitectura Web con HTML, CSS y JavaScript

5.5. Aplicaciones Web Progresivas (PWA)

Aunque la solución se utiliza directamente en el navegador y no requiere instalación, adopta principios de las Aplicaciones Web Progresivas (PWA): experiencia ligera y multiplataforma, uso de APIs del dispositivo (geolocalización) y despliegue centralizado con actualizaciones inmediatas. Esto aporta portabilidad y bajo costo de mantenimiento frente a aplicaciones nativas.



Figura 4 Arquitectura y Beneficios de las PWA

5.6. Identificación de visitas mediante códigos QR

Los códigos QR codifican la URL de acceso con un identificador único de visita (VisitID). Al escanearlo, el visitante ingresa a la interfaz web y queda vinculado a su sesión, estandarizando el proceso, reduciendo errores de digitación y habilitando la trazabilidad de extremo a extremo.



Figura 5 Control de Acceso con Códigos QR

5.7. Visualización cartográfica y trayectorias en la web – Leaflet

El uso de una librería de mapas web (p. ej., Leaflet) permite representar puntos de ubicación, dibujar polilíneas del recorrido y centrar la vista en la última posición reportada. Combinada con suscripciones a la base de datos en tiempo real, posibilita una visualización dinámica sin recargas, adecuada para la supervisión continua por parte del personal autorizado.



Figura 6 Mapas y Trayectorias Web con Leaflet

5.8. Marco normativo sobre el uso de tecnologías de geolocalización

En Nicaragua, el uso de tecnologías de geolocalización y transmisión de datos, como el Sistema de Posicionamiento Global (GPS), se encuentra regulado de manera indirecta a través de normativas relacionadas con las telecomunicaciones, la privacidad y el manejo de la información.

En este contexto, la Ley General de Telecomunicaciones Convergentes (Ley 1223) establece el marco jurídico que regula la prestación de servicios de telecomunicaciones y el uso de redes para la transmisión de datos en el país. Esta normativa contempla aspectos relacionados con la gestión, supervisión y control de las comunicaciones electrónicas, incluyendo aquellas que permiten la transmisión de información en tiempo real.

Dicha ley también establece disposiciones orientadas a la regulación del rastreo y la vigilancia de comunicaciones, garantizando que estos procesos se realicen bajo

principios de legalidad, seguridad y respeto a los derechos fundamentales de las personas. En este sentido, cualquier sistema que implique la recopilación y transmisión de datos, como es el caso del monitoreo de ubicación mediante GPS, debe operar bajo condiciones que aseguren la protección de la información y el uso adecuado de las redes de comunicación.

En relación con el sistema propuesto en este proyecto, se destaca que su funcionamiento se basa en la transmisión de datos a través de internet, haciendo uso de plataformas en la nube como Firebase, lo cual se encuentra dentro del marco de servicios regulados por esta ley. Asimismo, el sistema solicita el consentimiento del usuario para acceder a su ubicación, cumpliendo con los principios de privacidad y uso responsable de la información.

Por lo tanto, el desarrollo e implementación del sistema de monitoreo de visitantes se considera alineado con la normativa vigente en materia de telecomunicaciones en Nicaragua, garantizando un uso adecuado de las tecnologías de geolocalización dentro de un entorno institucional.

6. Metodología del proyecto

El desarrollo del Sistema de Monitoreo de Visitantes de la Universidad Nacional de Ingeniería se llevó a cabo mediante una metodología estructurada en tres etapas fundamentales, diseñadas para garantizar un proceso ordenado que permitiera el diseño, implementación, evaluación y validación del prototipo. Cada etapa se articula con los componentes descritos en el marco teórico, asegurando coherencia técnica, escalabilidad y eficiencia en el uso de los recursos disponibles.



Figura 7 Arquitectura de Geolocalización con Firebase

6.1. Enfoque metodológico

La metodología adoptada se basó en un enfoque incremental. Esto permitió verificar tempranamente la viabilidad del uso del GPS integrado en dispositivos móviles, la comunicación con Firebase en tiempo real y la visualización dinámica del recorrido en una interfaz web. A medida que se completaban pequeños módulos funcionales, estos se evaluaban en escenarios controlados para iterar sobre decisiones de diseño, estructura de datos, rendimiento y experiencia de usuario.

Este enfoque resultó adecuado debido a la naturaleza del proyecto, que combina tecnologías web, geolocalización, sincronización en tiempo real y despliegue en la nube, todas características que requieren validación progresiva.

6.2. Estructura de la metodología por etapas

El proceso metodológico se organiza en tres etapas principales:

1. **Etapa 1:** Diseño, configuración del sistema.
2. **Etapa 2:** Simulación funcional y pruebas del prototipo
3. **Etapa 3:** Evaluación de viabilidad técnica y operativa

Cada etapa se detalla a continuación.

6.2.1. Etapa 1: Diseño y configuración Inicial

Se descartó el uso de hardware adicional (Arduino, Raspberry Pi) al comprobar que los smartphones ofrecen suficiente precisión GPS, menor costo y mayor simplicidad operativa. Esta decisión permitió reducir la complejidad del sistema, facilitar su implementación y aprovechar recursos tecnológicos ya disponibles en los usuarios.

El sistema se definió como una solución basada en aplicación web + Firebase, utilizando:

• GPS del móvil

• Códigos QR con un identificador único

• Firestore para almacenamiento en tiempo real

• Hosting de Firebase para el despliegue

• Panel web para monitoreo del CPF

La aplicación web se desarrolló con HTML, CSS y JavaScript, priorizando un flujo sencillo: escaneo del QR → inicio → envío automático de coordenadas → cierre de visita. Este enfoque permite que el usuario interactúe con el sistema de forma intuitiva, sin necesidad de conocimientos técnicos.

Además, se diseñó una arquitectura orientada a la comunicación en tiempo real, donde cada dispositivo actúa como emisor de datos de ubicación hacia la base de

datos en la nube, permitiendo su visualización inmediata en el panel de monitoreo. Esto garantiza una supervisión continua y eficiente por parte del personal autorizado.

Durante esta etapa también se definieron las estructuras de datos en Firestore, organizando la información en colecciones y subcolecciones para registrar visitas y sus respectivas ubicaciones. Asimismo, se establecieron criterios básicos de funcionamiento, como la identificación única de cada visita, la actualización periódica de coordenadas y la correcta finalización del recorrido.

En conjunto, el diseño planteado permite una solución escalable, de bajo costo y adaptable a diferentes entornos, cumpliendo con los requerimientos iniciales del sistema.

6.2.2. Etapa 2: Simulación funcional y pruebas del prototipo

En esta etapa se validó el funcionamiento integral del sistema mediante la ejecución de diferentes escenarios de prueba, diseñados para simular condiciones reales dentro del entorno del campus. Estas pruebas permitieron evaluar el comportamiento del sistema en situaciones cotidianas de uso y verificar su estabilidad.

Se consideraron los siguientes escenarios:

• Recorridos cortos y extendidos dentro del campus

• Entradas y salidas de edificios

• Uso simultáneo de varios dispositivos conectados al sistema

El procedimiento consistió en la generación de códigos QR únicos para cada prueba, el inicio de la visita por parte del usuario, la captura continua de coordenadas durante el recorrido y la finalización del proceso. Durante estas pruebas, se registraron métricas relevantes como tiempos de actualización, precisión de ubicación y cantidad de puntos generados en cada trayectoria.

Adicionalmente, se evaluó la respuesta del sistema ante condiciones variables, como la pérdida parcial de señal GPS o fluctuaciones en la conectividad a internet, con el fin de analizar su tolerancia a fallos y su comportamiento en escenarios no ideales.

Los criterios de aceptación definidos fueron: actualización de datos en un tiempo menor a 1 segundo, generación de trayectorias continuas sin interrupciones, correcto cierre de las visitas y estabilidad general del sistema durante su ejecución. Los resultados obtenidos demostraron un desempeño adecuado, cumpliendo con los requisitos establecidos para su funcionamiento

6.2.3. Etapa 3: Evaluación de viabilidad

En esta etapa se evaluó la viabilidad del sistema propuesto desde los enfoques técnico, económico y operativo, con el objetivo de determinar su factibilidad de implementación en el entorno de la UNI–RUSB. Se analizaron aspectos como el desempeño del sistema, los recursos tecnológicos requeridos, los costos asociados y la facilidad de uso para los usuarios finales. Los resultados obtenidos evidencian que el sistema es funcional, accesible y adaptable a las condiciones institucionales, lo que confirma su viabilidad para una posible implementación real.

6.3. Viabilidad técnica

La viabilidad técnica del sistema propuesto es alta, debido a que se fundamenta en tecnologías actuales, accesibles y ampliamente utilizadas en el desarrollo de aplicaciones web y sistemas de monitoreo en tiempo real.

El sistema hace uso del GPS integrado en dispositivos móviles, eliminando la necesidad de hardware adicional como módulos externos o microcontroladores. Esto reduce significativamente la complejidad del sistema y facilita su implementación, ya que aprovecha recursos disponibles en la mayoría de smartphones modernos.

Se utiliza la plataforma Firebase como backend, específicamente Firestore, que permite la sincronización de datos en tiempo real. Esto garantiza que la información de ubicación

de los visitantes se actualice de manera inmediata en el panel de monitoreo, cumpliendo con los requerimientos del sistema.

La aplicación fue desarrollada utilizando tecnologías web estándar como HTML, CSS y JavaScript, lo que asegura compatibilidad con múltiples dispositivos y navegadores. Además, el uso de Firebase Hosting permite desplegar la solución en un entorno seguro mediante HTTPS y con alta disponibilidad.

Las pruebas realizadas evidenciaron una latencia baja (menor a 2 segundos), estabilidad en la transmisión de datos y correcto funcionamiento bajo múltiples usuarios simultáneos. Esto demuestra que el sistema es escalable y capaz de operar en entornos reales.

El sistema es técnicamente viable, ya que utiliza herramientas confiables, no requiere infraestructura compleja y puede ser implementado con los recursos tecnológicos disponibles en la institución.

6.3.1. Viabilidad económica

La viabilidad económica del proyecto es favorable, ya que el sistema ha sido diseñado para minimizar costos de implementación y mantenimiento. Una de las principales ventajas es que no requiere inversión en hardware adicional, debido a que utiliza dispositivos móviles existentes para la captura de ubicación mediante GPS, eliminando gastos asociados a la compra de equipos especializados.

En cuanto a la infraestructura tecnológica, Firebase ofrece un plan gratuito que permite cubrir las necesidades iniciales del sistema, incluyendo almacenamiento, sincronización en tiempo real y hosting. Esto hace posible implementar el sistema sin incurrir en costos significativos durante su fase inicial.

No obstante, con el objetivo de presentar un escenario más completo, se muestra a continuación una estimación de costos considerando el uso de servicios en su modalidad de pago y el desarrollo del sistema:

Estimación de costos del proyecto

Los costos estimados del proyecto son los siguientes:

- Desarrollo del sistema: 24 horas de programación a un costo de C$ 500 por hora (C$ 12,000).
- Uso de Firebase en plan de pago: C$ 850 mensuales.
- Servicio de generación de códigos QR en versión premium: C$ 350 mensuales.

En total, la inversión estimada asciende a C$ 12,950.

Nota. Los costos presentados son estimaciones basadas en valores promedio del mercado. El uso de plataformas en versiones gratuitas puede reducir significativamente el costo total del proyecto.

Asimismo, los únicos costos potenciales adicionales estarían relacionados con el incremento en el uso de la plataforma en caso de escalar el sistema o la contratación de servicios complementarios. Sin embargo, estos costos son variables y dependen directamente del crecimiento del sistema.

Finalmente, los costos operativos son bajos, ya que no se requiere personal técnico altamente especializado para su uso diario, sino únicamente capacitación básica para el personal encargado del monitoreo.

El proyecto es económicamente viable, ya que su implementación requiere una inversión mínima y su mantenimiento puede ser gestionado con recursos accesibles.

6.3.2. Viabilidad operativa

La viabilidad operativa del sistema es alta, ya que su diseño está orientado a facilitar su uso dentro del entorno institucional de la UNI–RUSB.

El sistema presenta un flujo de uso sencillo para los visitantes, quienes únicamente deben escanear un código QR, otorgar permisos de ubicación e iniciar su visita. Este proceso no requiere conocimientos técnicos avanzados, lo que facilita su adopción.

Por otro lado, el panel de monitoreo está diseñado para ser intuitivo, permitiendo al personal del Cuerpo de Protección Física (CPF) visualizar en tiempo real la ubicación de los visitantes, sus trayectorias y el estado de cada visita. Esto mejora la capacidad de supervisión y la toma de decisiones.

Durante las pruebas, se comprobó que el sistema puede integrarse fácilmente a los procesos actuales de control de acceso, sin necesidad de modificar significativamente la infraestructura o los procedimientos existentes.

Además, el sistema es accesible desde cualquier dispositivo con conexión a internet, lo que permite su uso tanto en computadoras como en dispositivos móviles.

El sistema es operativamente viable, ya que es fácil de usar, se adapta al entorno institucional y mejora los procesos de supervisión sin generar complejidad adicional.

7. Desarrollo del proyecto

El siguiente apartado presenta de manera detallada el proceso de construcción del sistema de monitoreo de visitantes, abarcando desde la configuración inicial del entorno hasta la implementación completa de las funcionalidades principales. En este capítulo se explica paso a paso cómo se estructuró la aplicación web, cómo se integraron los servicios de Firebase —incluyendo Hosting y Firestore— y cómo se desarrollaron tanto la interfaz del visitante como el panel de monitoreo para el supervisor. De esta forma, se busca proporcionar una descripción clara, ordenada y comprensible del desarrollo técnico que permitió dar forma al sistema final.

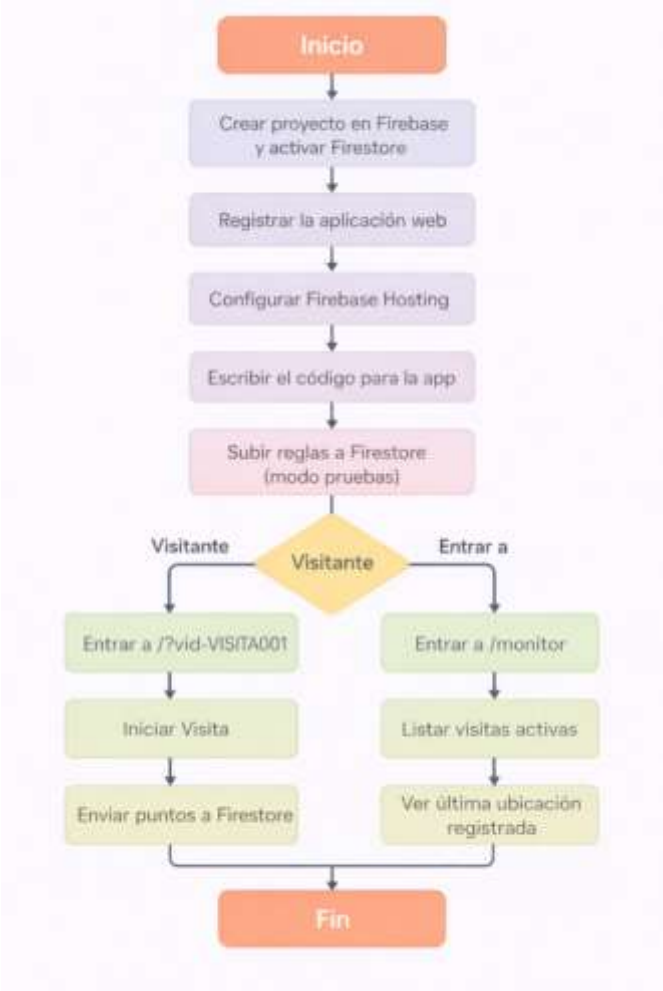


Figura 8 Diagrama de flujo de proceso

7.1. Estructura del proyecto

Los archivos se almacenan dentro de la carpeta destinada al desarrollo del sistema. En este caso, todos fueron ubicados en una ruta local.

La estructura final del proyecto queda organizada de la siguiente forma:

Tesis/

├─ index.html

├─ styles.css

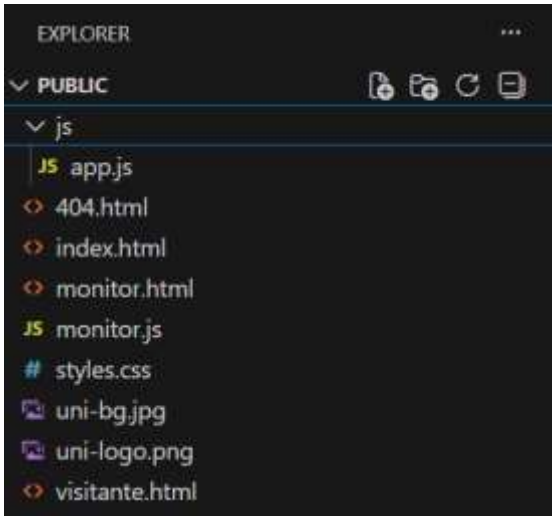
├─ Js/app.js

├─ monitor.html

├─ monitor.js

├─ firebase.json

└─ firestore.rules



En los anexos podrán encontrar el repositorio para que puedan descargar los archivos utilizados. Asi mismo, el paso a paso para crear un proyecto desde 0 en Firebase.

En este proyecto se utiliza en:

app.js

Archivo utilizado para:

- Inicializar Firebase
- Registrar la visita
- Enviar coordenadas GPS
- Guardar datos en Firestore
- Dibujar el recorrido en el mapa

monitor.js

Archivo encargado de:

- Conectarse a Firestore
- Listar visitas activas
- Consultar la última ubicación registrada
- Mostrarla en el mapa del panel de monitoreo

Una vez registrada la app web e insertada la configuración firebaseConfig en los archivos del proyecto:

- La aplicación puede comunicarse con Firestore sin restricciones.
- Se habilita el manejo de datos en tiempo real.
- El sistema queda preparado para:
    - Guardar visitas
    - Registrar ubicaciones
    - Mostrar recorridos
    - Actualizar el monitor sin recargar la página

Este paso es uno de los cimientos del proyecto, ya que, sin la app registrada y configurada, ninguna de las funciones posteriores (monitoreo, envío de ubicación, renderización de mapas) podría ejecutarse correctamente.

7.1.1. Configurar Firebase Hosting

Firebase Hosting es el servicio encargado de alojar la aplicación web y ponerla a disposición del usuario final mediante un dominio seguro y optimizado. En este apartado se describe el proceso completo para instalar las herramientas necesarias, iniciar sesión, configurar el entorno y preparar la carpeta del proyecto para su despliegue.

Instalar Node.js (recomendado LTS)

1. Ve a **https://nodejs.org**
2. Descarga la versión **LTS** (NO la Current)
3. Ejecuta el instalador
4. Deja marcada esta opción
    - Add Node.js to PATH
5. Finaliza la instalación

7.1.2. Instalación de Firebase CLI

Para poder interactuar con Firebase desde la computadora, es necesario instalar Firebase CLI (Command Line Interface). Esta herramienta permite ejecutar comandos como inicializar proyectos, desplegar contenido al hosting, actualizar reglas y consultar el estado del proyecto.

La instalación se realiza desde la terminal o símbolo del sistema(CMD) mediante el siguiente comando:

- npm install -g firebase-tools

Este comando descarga e instala Firebase CLI de manera global, permitiendo usarlo desde cualquier directorio del sistema.

Es importante contar con Node.js previamente instalado, ya que es un requisito para ejecutar estas herramientas.

7.1.3. Iniciar sesión en Firebase

Una vez instalada la CLI, se debe iniciar sesión con la cuenta de Google asociada al proyecto de Firebase. Esto se logra ejecutando:

- firebase login

El comando abrirá una ventana en el navegador solicitando autorización para que Firebase CLI pueda acceder a los proyectos disponibles en tu cuenta.

Después de confirmar el acceso, la terminal mostrará un mensaje indicando que la sesión se inició correctamente.

7.1.4. Inicializar Firebase Hosting dentro del proyecto

Con la sesión iniciada, se debe preparar la carpeta donde se encuentran los archivos del sistema.

Para ingresar a la ubicación, se utiliza:

- cd C:\Users\Localuser\Documents\ (Dependerá donde quieren guardar los archivos.)

En este caso, el proyecto se almacenó en:

- C:\Users\Luis\Documents\Tesis

Una vez dentro de la carpeta del proyecto, se ejecuta el comando:

- firebase init hosting

Este comando iniciará un asistente interactivo en la consola, el cual solicitará una serie de configuraciones.

A continuación, se detallan las opciones que deben seleccionarse en el orden correcto.

- **Selección de opciones dentro del asistente de Hosting**

Cuando aparece el menú interactivo, se deben elegir las siguientes opciones:

- **Seleccionar “Proyecto existente”**

Permite vincular la carpeta local con un proyecto ya creado en Firebase. Se mostrará una lista de proyectos asociados a la cuenta; se debe seleccionar:

- sistema-de-monitoreo-8542b (o el nombre asignado al proyecto).

2. Directorio público: . (punto)

Firebase solicita indicar qué carpeta contendrá los archivos que se subirán al Hosting.

Se debe escribir:

- . (punto)

Esto indica que **la carpeta raíz** del proyecto es la que contiene los archivos de la aplicación (index.html, app.js, etc.).

3. **No configurar SPA (Single Page Application)**

Firebase pregunta:

"¿Configurar como SPA utilizando rewrites?"

Para este sistema NO es necesario activar esta opción, ya que se utilizan rutas específicas (/, /monitor) y no un framework SPA como React o Angular.

Por ello, seleccionar:

- No

4. No sobrescribir index.html

En caso de mostrar el mensaje: "¿Desea sobrescribir index.html?"

Se debe elegir: No. Esto evita que Firebase cree un archivo nuevo y sobrescriba el que ya forma parte del proyecto.

Resultado de la inicialización

Al finalizar, Firebase CLI generará automáticamente:

- Un archivo firebase.json
- Un archivo .firebaserc
- Una estructura básica lista para desplegar

7.1.5. Crear los archivos del proyecto

Una vez configurado Firebase Hosting, el siguiente paso consiste en preparar la estructura de archivos que compondrán la aplicación web. Cada archivo cumple una función específica dentro del sistema, ya sea en la interfaz del usuario, la lógica del funcionamiento, la comunicación con Firebase o la configuración del entorno. A continuación se describen los archivos principales y su propósito dentro del desarrollo.

7.1.6. Descripción de los archivos

index.html – Interfaz del visitante

Este archivo constituye la página principal del sistema utilizada por el visitante. Incluye los elementos gráficos y funcionales que permiten visualizar el estado de la visita, mostrar el mapa de geolocalización y ejecutar la acción de inicio de visita. Su estructura HTML sirve como contenedor de los componentes dinámicos generados por JavaScript.

styles.css – Estilos generales

Contiene las reglas de estilo (CSS) que permiten definir la apariencia visual de la aplicación: colores, tipografías, tamaños, espaciados, diseño del mapa y componentes. Su función es garantizar una presentación visual uniforme tanto en la vista del visitante como en el panel del supervisor.

app.js – Lógica GPS + Firestore + mapa

Archivo crítico del sistema. En él se implementa toda la lógica del visitante, incluyendo:

- Inicialización de Firebase
- Obtención del VisitID

- Activación del GPS mediante *navigator.geolocation*
- Envío de ubicación a Firestore
- Creación de documentos y subcolecciones
- Dibujo del recorrido en un mapa usando Leaflet
- Evitar duplicación de visitas activas

Este archivo permite que el visitante pueda enviar su ubicación en tiempo real mientras realiza la visita.

monitor.html – Interfaz del supervisor

Página diseñada para el monitoreo administrativo. Aquí se muestran:

- Todas las visitas activas
- Últimas ubicaciones registradas
- Un mapa que permite centrar la vista en cualquier visitante

Sirve como una interfaz accesible desde cualquier dispositivo web para supervisores o personal de control.

monitor.js – Lógica del monitor

Archivo de JavaScript asociado al panel del supervisor. Su función principal es:

- Conectarse a Firestore
- Consultar en tiempo real las visitas activas
- Obtener la última ubicación registrada
- Actualizar el mapa al seleccionar un visitante

Trabaja de manera continua para mantener el monitoreo en tiempo real sin necesidad de actualizar la página.

firebase.json – Configuración de Hosting

Es el archivo de configuración generado por Firebase CLI que define cómo se comportará el Hosting del proyecto. Incluye:

- Carpeta pública desde la cual se servirán archivos
- Rutas especiales (por ejemplo: /monitor)
- Archivos ignorados
- Reescrituras (rewrites) necesarias para el sistema

Este archivo permite que Firebase sirva correctamente tanto la página del visitante como el panel del supervisor.

firestore.rules – Reglas de seguridad

Contiene las reglas que definen qué operaciones están permitidas dentro de Firestore. Inicialmente se utilizaron reglas abiertas para pruebas, pero en un entorno de producción deben endurecerse para limitar el acceso no autorizado.

7.1.7. Código principal de la aplicación

El sistema desarrollado se compone de varios archivos que colaboran entre sí para ofrecer las funcionalidades de monitoreo, captura de ubicación, renderización de mapas y visualización en tiempo real. Cada archivo cumple un rol específico dentro de la arquitectura del proyecto, combinando lógica de front-end, interacción con servicios de Firebase y manejo de la interfaz. A continuación, se describen los archivos principales y su función dentro del sistema.

Archivo: index.html

Este archivo constituye la **interfaz principal del visitante**, es decir, la pantalla desde donde la persona que realiza la visita inicia el proceso de monitoreo. Su estructura incluye los siguientes elementos:

- **Visualización del VisitID y estado actual.**
  El VisitID es indispensable para identificar cada visita en Firestore.

- **Botón “Iniciar Visita”.** Permite comenzar el proceso de captura de ubicación.
- **Integración de Leaflet**, una librería especializada en mapas interactivos. Esto permite mostrar el recorrido en tiempo real.
- **Referencia al archivo app.js**, que contiene la lógica completa de la captura de ubicación y comunicación con Firestore.

En términos funcionales, index.html actúa como el punto de entrada para la interacción del visitante con todo el sistema.

Archivo: app.js

Este archivo implementa la mayor parte de la lógica operativa del lado del visitante, integrando servicios de Firebase con la API de geolocalización del navegador.

Sus funciones principales son:

Obtención del VisitID desde la URL

Utiliza parámetros como ?vid=VISITA001 para identificar de manera única cada visita. Si el VisitID no está presente, el sistema lo solicita al usuario mediante un cuadro de diálogo, garantizando que cada visitante tenga un identificador válido.

Creación y actualización de documentos en Firestore

El código registra la visita dentro de la colección:

- visitas/{VISIT_ID}

Y adicionalmente almacena cada punto GPS en la subcolección:

- visitas/{VISIT_ID}/ubicaciones

Incluyendo latitud, longitud, precisión y una marca de tiempo.

Activación del GPS mediante watchPosition()

Se utiliza la API nativa del navegador para obtener la ubicación del dispositivo en intervalos continuos. Este método permite actualizaciones en tiempo real mientras el visitante se desplaza.

Dibujo del recorrido sobre un mapa

Con Leaflet, el sistema:

- coloca marcadores,
- genera una polilínea con el recorrido,
- centra el mapa en la última ubicación registrada.

Prevención de visitas duplicadas

Si un VisitID ya se encuentra con estado “activa”, el sistema evita reiniciar la visita, previniendo comportamientos duplicados o la creación de múltiples listeners de GPS.

En conjunto, app.js conecta al visitante con toda la infraestructura del sistema.

Archivo: monitor.html

Es la interfaz del supervisor o monitor, y permite visualizar el estado de todas las visitas activas.

Sus funciones principales incluyen:

- Mostrar una tabla dinámica con los VisitID activos.
- Desplegar la última ubicación registrada de cada visita.
- Botón “Ver mapa” para centrar el mapa en el visitante seleccionado.
- Zona del mapa donde se renderiza la ubicación utilizando Leaflet.

Este archivo proporciona una vista administrativa que permite supervisar el desplazamiento de los visitantes en tiempo real.

Archivo: monitor.js

Este archivo contiene la lógica del panel de monitoreo. Está diseñado para funcionar en tiempo real gracias a las capacidades de Firestore.

Sus funciones incluyen:

Conexión a Firestore

Inicializa Firebase y obtiene acceso a la colección visitas y sus subcolecciones asociadas.

Consulta en tiempo real (onSnapshot)

Permite recibir actualizaciones automáticas, sin recargar la página, cuando:

- Una visita se activa,
- se añade una nueva ubicación,
- cambia información relevante del visitante.

Esto convierte el panel en un sistema de monitoreo fluido y permanente.

Obtención de la última ubicación

Por cada VisitID, el sistema consulta la subcolección ubicaciones ordenada por tiempo, obteniendo el punto más reciente.

Centrado del mapa en el visitante seleccionado

Al presionar el botón “Ver mapa”, el mapa se posiciona sobre la última coordenada recibida.

Con este archivo, el supervisor cuenta con una herramienta completa para visualizar y gestionar el estado de todas las visitas.

Archivo: firebase.json

Este archivo contiene la configuración del servicio Firebase Hosting. Incluye directivas que indican:

- Carpeta pública desde donde se sirven los archivos (en este caso, “.”)
- Archivos que deben ignorarse
- Reescrituras específicas, como:

```
"rewrites": [
  { "source": "/monitor", "destination": "/monitor.html" }
]
```

Figura 9 Regla de Redirección en Hosting

Esta escritura permite que el supervisor acceda al panel mediante la ruta: /monitor, Sin necesidad de especificar el archivo manualmente.

Archivo: firestore.rules

Contiene las reglas de acceso para Firestore. Durante la fase de desarrollo se utilizó un conjunto de reglas abiertas para permitir pruebas sin restricciones:

- allow read, write: if true;

Estas reglas son temporales y deben reemplazarse por un conjunto más estricto para producción, a fin de evitar accesos no autorizados.

Despliegue del proyecto

Una vez implementados los archivos principales del sistema y configurado correctamente Firebase Hosting, el siguiente paso consiste en realizar el despliegue (deploy) de la aplicación. Este proceso permite publicar la plataforma en un dominio web proporcionado por Firebase, de modo que tanto el visitante como el supervisor puedan acceder al sistema desde cualquier dispositivo con conexión a internet.

El despliegue se realiza utilizando Firebase CLI, que se encarga de empaquetar los archivos del proyecto, aplicar reglas de Firestore, actualizar la configuración del hosting y cargar los recursos al servidor. A continuación, se describe el procedimiento completo.

Verificar la carpeta del proyecto

Antes de desplegar, es necesario comprobar que la terminal se encuentra ubicada dentro de la carpeta donde se encuentran los archivos del sistema. En este caso:

- C:\Users\Luis\Documents\Tesis

Despliegue de reglas de Firestore

Las reglas de seguridad de Firestore (archivo firestore.rules) deben cargarse antes del funcionamiento completo del sistema, ya que determinan qué operaciones son permitidas en la base de datos. Durante el desarrollo se utilizaron reglas abiertas para facilitar pruebas, y estas se despliegan con el comando: firebase deploy --only firestore

Este comando envía directamente las reglas definidas a la consola de Firebase, habilitando el funcionamiento de lectura y escritura para el sistema durante la fase de pruebas.

En entornos de producción se deberá modificar este archivo para implementar controles de acceso más estrictos.

Despliegue del hosting

Una vez cargadas las reglas y con la estructura del proyecto lista, se procede a desplegar la aplicación web utilizando:

- firebase deploy --only hosting

Este comando:

- Sube todos los archivos del proyecto a Firebase Hosting
- Aplica la configuración del archivo firebase.json
- Habilita las rutas definidas (incluyendo la de /monitor)
- Genera una versión actualizada de la página
- Publica el contenido en un dominio seguro https://…web.app/

Al finalizar, la consola mostrará un mensaje confirmando que el despliegue fue exitoso y proporcionará las URLs del proyecto. En este caso https://sistema-de-monitoreo-8542b.web.app/

Resultado del despliegue

Una vez completado el deploy, el sistema queda disponible en dos rutas principales:

- **Ruta del visitante**
  - https://sistema-de-monitoreo-8542b.web.app/?vid=visita001

El visitante puede:

- Iniciar su visita
- Compartir ubicación en tiempo real
- Ver su recorrido en el mapa



Figura 10 Visualización de interfaz vista desde un dispositivo móvil

- **Ruta del supervisor (monitor)**
  - https://sistema-de-monitoreo-8542b.web.app/monitor.html

El supervisor puede:

- Ver todas las visitas activas
- Consultar la última ubicación del visitante
- Centrar el mapa en cualquier recorrido

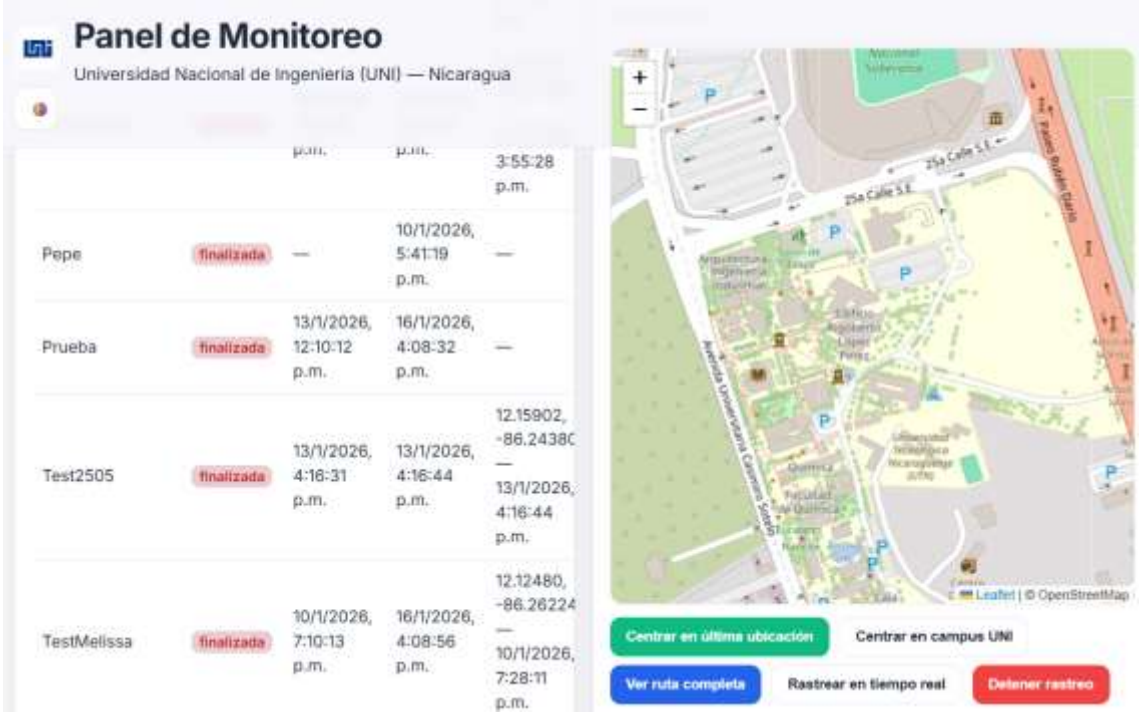


Figura 11 Visualización de Panel de Monitoreo

Importancia del despliegue

El proceso de despliegue representa la transición entre el desarrollo local y la puesta en marcha del sistema a nivel público. A partir de este momento:

- La aplicación puede ser utilizada por visitantes reales
- La plataforma está accesible desde telefonía móvil y escritorio
- El monitoreo se convierte en un servicio continuo
- La base de datos Firestore registra datos en tiempo real

Además, cada vez que el proyecto requiera una actualización (código corregido, nuevas funciones o estilos visuales), bastará con repetir el comando:

- firebase deploy --only hosting

7.2. Generación del código QR para el acceso del visitante

Con el propósito de agilizar el acceso del visitante al sistema y reducir errores durante el proceso de digitación del enlace, se implementó un mecanismo de acceso mediante códigos QR personalizados. Esta estrategia permite que cada visitante ingrese directamente a su interfaz correspondiente simplemente escaneando el código con la cámara de su dispositivo móvil, sin necesidad de teclear direcciones web ni parámetros manuales.

El primer paso consistió en definir la URL base del sistema desplegado en Firebase Hosting. En este caso, la dirección definitiva generada por Firebase fue: https://sistema-de-monitoreo-8542b.web.app

Una vez generados los enlaces personalizados, se utilizó la herramienta en línea:

- **https://www.the-qrcode-generator.com/**



Figura 12 Acceso a panel de visitante mediante QR

Esta plataforma permite convertir fácilmente cualquier URL en un código QR escaneable.

1. Seleccionar la opción para generar un QR de tipo URL.

2. Generar el código automáticamente.
3. Descargar la imagen resultante (generalmente en formato PNG).
4. Imprimir o distribuir el código según fuera necesario.

7.3. Presentación y análisis de resultados

La evaluación comprende tres dimensiones:

1. **Técnica:** latencia extremo a extremo (desde que el visitante envía coordenadas hasta su visualización en el panel), integridad y orden de los datos almacenados, precisión efectiva del GPS en condiciones reales (exteriores e interiores), estabilidad del panel con concurrencia moderada (múltiples visitas activas) y tiempos de carga del frontend bajo HTTPS/CDN.
2. **Operativa:** simplicidad del flujo para el visitante (escaneo QR → permiso → iniciar → finalizar), claridad de la interfaz del CPF (última ubicación, trayectoria, estado de visita) y tiempo de adopción (capacitación mínima).

7.3.1. Preguntas guía

- ¿La **latencia** de actualización en el panel es lo suficientemente baja para apoyar el monitoreo en tiempo real?
- ¿La **precisión** del GPS es adecuada para el contexto del campus, y se maneja correctamente la variabilidad en interiores?
- ¿El sistema se integra sin fricciones a los **procedimientos** y **recursos** disponibles (conectividad, dispositivos, turnos)?

7.3.2. Valores obtenidos

- Latencia (mediana) ≤ 1–2 s; integridad ≥ 98 % de puntos esperados, sin duplicados y ordenados por tiempo; precisión aceptable en exteriores (mayoría de puntos con precesión < 25 m), con registro explícito de precisión en interiores, concurrencia fluida con visitas activas, tiempo de carga del frontend acorde a uso operativo en campus; flujo de uso del visitante ≤ 3 pasos y panel comprensible para el CPF.

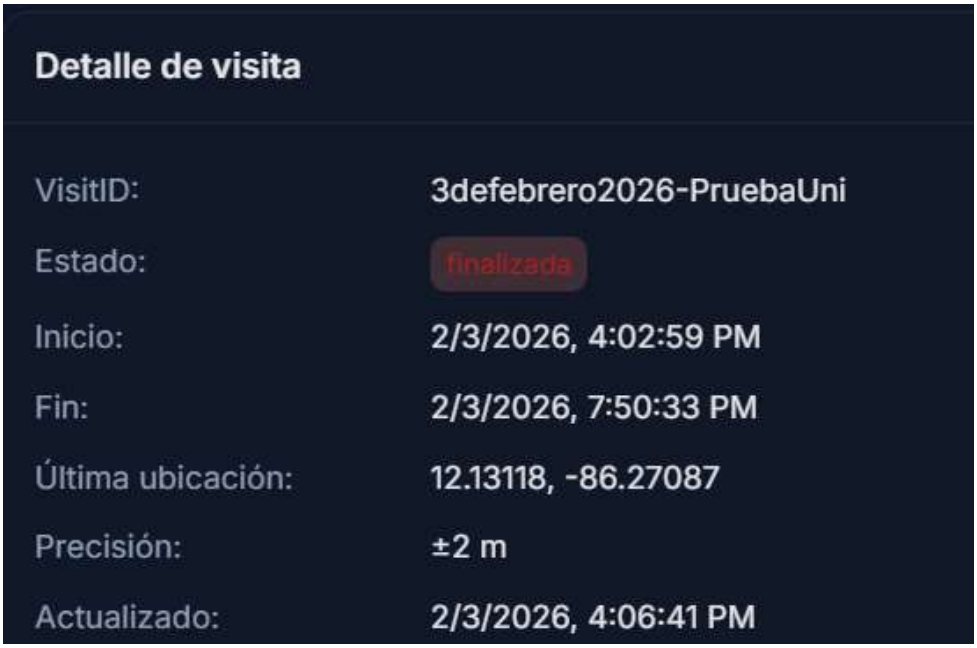


Figura 13 Datos obtenidos en prueba 1

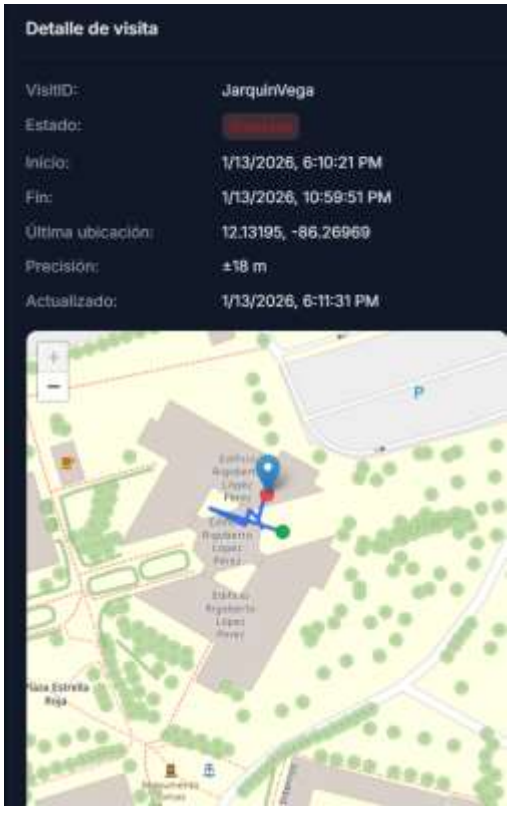


Figura 14 Trayectoria vista en mapa

En base en las pruebas realizadas y las métricas obtenidas durante esta etapa, se concluye que el sistema es viable técnica y operativamente para su implementación en la UNI. El prototipo funciona de manera estable, responde en tiempo real, es fácil de usar tanto para los visitantes como para el CPF y cumple con los requisitos de monitoreo definidos en el proyecto. Por ello, el sistema se considera apto para su adopción institucional.

8. Conclusiones

Se logró identificar de manera clara y estructurada los componentes necesarios para el desarrollo de un sistema de monitoreo de visitantes en el campus UNI-RUSB, integrando tecnologías de geolocalización mediante GPS y herramientas basadas en servicios en la nube. Esta identificación permitió definir una arquitectura funcional capaz de responder a las necesidades de control y seguimiento dentro del entorno universitario, optimizando los procesos de supervisión y reduciendo la dependencia de métodos manuales.

Se configuró e implementó correctamente la plataforma Firebase como solución de almacenamiento y sincronización de datos, garantizando la gestión eficiente de la información generada por el sistema. Gracias a sus capacidades en tiempo real, se logró mantener actualizados los registros de ubicación y actividad de los visitantes, facilitando el acceso inmediato a la información y mejorando la toma de decisiones por parte del personal encargado.

Se evaluó la viabilidad técnica y operativa del sistema propuesto, comprobando que su implementación en el campus UNI-RUSB es factible y adecuada. Los resultados obtenidos demuestran que el sistema no solo funciona correctamente, sino que también aporta beneficios significativos en términos de seguridad, control de acceso y supervisión de visitantes, además de presentar potencial de escalabilidad para futuras mejoras o ampliaciones.

Adicionalmente, el sistema demostró ser una solución eficiente en cuanto a costos, al no requerir infraestructura adicional ni equipos especializados, aprovechando dispositivos móviles existentes y servicios en la nube. Esto lo convierte en una alternativa accesible para su implementación en otras instituciones con características similares.

En conjunto, el proyecto cumple con los objetivos planteados, validando el uso de tecnologías modernas para la implementación de sistemas de monitoreo en tiempo real, y contribuye al fortalecimiento de la seguridad institucional mediante una solución innovadora, escalable y de fácil adopción.

9. Recomendaciones

- **Implementación de Autenticación Multifactor (MFA).**

Se recomienda incorporar un mecanismo de Autenticación Multifactor (MFA) para todo acceso administrativo dentro del sistema de monitoreo. Esta medida refuerza la seguridad del panel del supervisor al añadir una segunda capa de verificación, reduciendo la posibilidad de accesos no autorizados. El MFA puede implementarse mediante el envío de códigos de verificación al correo institucional del personal autorizado o mediante servicios de autenticación externos compatibles con Firebase. Debido a que el sistema gestiona información sensible relacionada con la ubicación de visitantes, este control adicional resulta fundamental para garantizar la integridad y la confidencialidad de los datos.

- **Activación de alertas automáticas por salida del perímetro**

Se recomienda anexar a código, un límite que incluya el perímetro puntualmente de la UNI, esto podrá generar alertas instantáneas cuando un visitante salga del área definida como perímetro de supervisión. Esta funcionalidad fortalecería la capacidad del personal del CPF para identificar desplazamientos no autorizados, incidentes, extravíos o situaciones de riesgo. Al detectarse la salida del perímetro, el sistema debería emitir una notificación automática en el panel de monitoreo, permitiendo una respuesta más rápida y oportuna ante cualquier eventualidad.

- **Integración de un botón de pánico para emergencias.**

Se recomienda incluir en la interfaz móvil del visitante un botón de pánico o emergencia, accesible de manera inmediata. Esta herramienta permitirá al visitante notificar situaciones críticas como accidentes, problemas de salud, amenazas o cualquier evento que requiera asistencia urgente. Al activarse el botón, el sistema debe enviar una notificación directa al panel del supervisor, acompañada de la última ubicación GPS registrada, permitiendo al Cuerpo de Protección Física coordinar una respuesta rápida y adecuada.

10. Referencias Bibliográficas

Darwin Tapia, R. A. (2016). *Repositorio UNAN.* Recuperado el 24 de abril de 2020, de https://repositorio.unan.edu.ni/3298/12/93806.pdf

docs, G. (2026). *Google Docs.* Obtenido de https://firebase.google.com/docs/

Garcia Carballo, J., & Malespín Cabrera, J. (2015). *SIBIUNI.* Obtenido de https://ribuni.uni.edu.ni/2668/1/93274.pdf

GPS.GOV. (16 de marzo de 2021). *GPS.GOV*. Recuperado el 12 de abril de 2021, de https://www.gps.gov/systems/gps/space/

Henao Melo, L. G. (2014). *Core.* Obtenido de Henao Melo, Luis Guillermo

Julio García, J. M. (2015). *https://ribuni.uni.edu.ni/2668/1/93274.pdf.* Recuperado el 4 de agosto de 2021, de https://ribuni.uni.edu.ni/2668/1/93274.pdf

Raldúa Veuthey, J. (2012). *Institutional Repository of Carlos III University of Madrid.* Obtenido de https://e-archivo.uc3m.es/rest/api/core/bitstreams/f216f761-aa6a-40b9-ba21-7be9e204892c/content

REAL ACADEMIA ESPAÑOLA. (2021). *RAE*. Recuperado el 07 de marzo de 2021

Tapia Moraga, D., & Aguilar Díaz, R. (2016). *Repositorio Institucional de la UNAN-Managua.* Obtenido de https://repositorio.unan.edu.ni/id/eprint/3298/12/93806.pdf

W3Schools. (2026). Obtenido de W3Schools: https://www.w3schools.com/html/

11. Anexos

Enlace a repositório código:

https://mngenvmcap133575-my.sharepoint.com/my?id=%2Fpersonal%2Fluis%5Ftechy%5Fcloudns%5Fch%2FDocuments%2FDocuments%2FTesis%2Fpublic&viewid=dbf565ad%2D349d%2D452f%2D80c1%2D82b797aeba21

Creación del Proyecto en Firebase

La creación del proyecto en Firebase constituye el primer paso fundamental para el desarrollo del sistema, ya que este servicio provee la infraestructura necesaria para alojar la aplicación web, gestionar la base de datos en tiempo real y habilitar la comunicación entre los distintos módulos. A continuación, se describe el proceso de forma detallada:

- Ir a: https://console.firebase.google.com
- Clic en Agregar proyecto
- Dar nombre al proyecto (ej.: “sistema-de-monitoreo”)
- Desactivar Google Analytics (opcional)
- Crear el proyecto

Activar Firestore

- En el panel izquierdo → Firestore Database
- Clic en Crear base de datos
- Modo de seguridad: Inicio en modo prueba
- Ubicación: predeterminada
- Crear

Crear Aplicación Web en Firebase

La creación de la aplicación web dentro del proyecto Firebase es un paso esencial para permitir la integración entre el sistema desarrollado y los servicios del ecosistema Firebase. Esta configuración proporciona las credenciales necesarias para conectar el

código JavaScript con la base de datos Firestore, así como con cualquier otro servicio habilitado, como Hosting o Authentication. A continuación, se detalla el proceso completo.

- En Firebase, panel lateral → Project Overview
- Clic en icono Web (</>)
- Nombre de app: “web-visitante”
- Clic en Registrar App

Obtener la configuración de Firebase (firebaseConfig)

Después del registro, Firebase mostrará un bloque de código con las credenciales y puntos de acceso necesarios para que la aplicación web pueda interactuar con el backend del proyecto. Este fragmento contiene información como la clave pública del servicio, el dominio del proyecto, el ID del proyecto y el identificador de la aplicación.

El bloque generado es de la siguiente forma:

```
const firebaseConfig = {
  apiKey: "…",
  authDomain: "…",
  projectId: "…",
  storageBucket: "…",
  messagingSenderId: "…",
  appId: "…"
};

```

Figura 15 Fragmento de código para obtención de ubicación

Este conjunto de claves cumple varias funciones esenciales:

- **Permite inicializar la app de Firebase** dentro del navegador.
- **Establece conexión con Firestore**, la base de datos del sistema.
- **Habilita el uso de Firebase Hosting**, Authentication y otros módulos.
- **Identifica nuestro proyecto** de forma ún ica a nivel global.

Estas credenciales no representan un riesgo de seguridad por sí mismas, ya que Firebase las considera información pública necesaria para la inicialización. La seguridad del sistema depende de las reglas de Firestore, no de este bloque.

Ubicación del firebaseConfig en los archivos del proyecto

Para que la aplicación pueda funcionar correctamente, este bloque debe ser incluido dentro de los archivos JavaScript encargados de manejar la lógica tanto del visitante como del monitor.

Obtención de la configuración (firebaseConfig)

Después de registrar la aplicación, Firebase mostrará un bloque de configuración conocido como firebaseConfig, el cual contiene información clave como:

• apiKey

• authDomain

• projectId

• storageBucket

• messagingSenderId

• appId

Este bloque es necesario para inicializar Firebase dentro del código JavaScript.

Funciones principales del firebaseConfig:

• Permite inicializar la aplicación en el navegador

• Establece la conexión con Firestore

• Habilita servicios como Hosting y Authentication

• Identifica el proyecto de manera única

Es importante destacar que estas credenciales son públicas y no representan un riesgo directo, ya que la seguridad se gestiona mediante las reglas de Firestore.

5. Integración de Firebase en el proyecto web

Una vez obtenido el firebaseConfig, este debe integrarse dentro del archivo JavaScript principal (por ejemplo, app.js y monitor.js).

Pasos:

• Importar los SDK de Firebase en el HTML

• Inicializar la aplicación utilizando firebaseConfig

• Establecer conexión con Firestore

Esto permite que la aplicación pueda enviar y recibir datos en tiempo real.

Imágenes asociadas al funcionamiento en pruebas de campo.



Figura 16 Escaneo de código QR



Figura 17 Prueba de registro desde dispositivo móvil



Figura 18 Registro de visitante en dispositivo IOS



Figura 19 Trayectoria finalizada de visitante

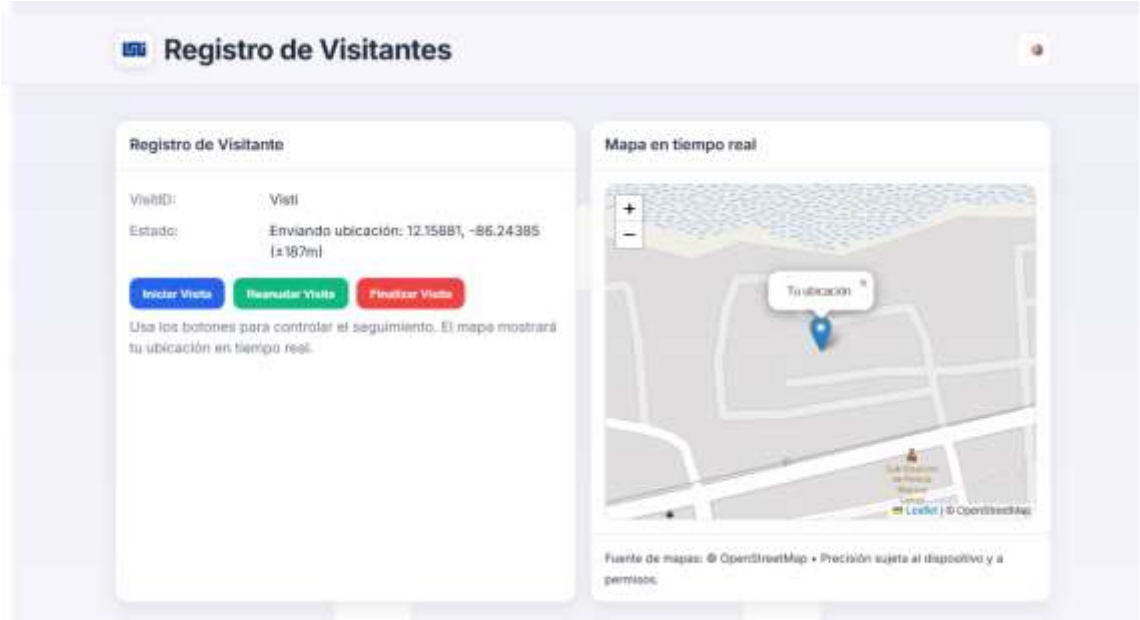


Figura 20 Vista completa de panel de visitante

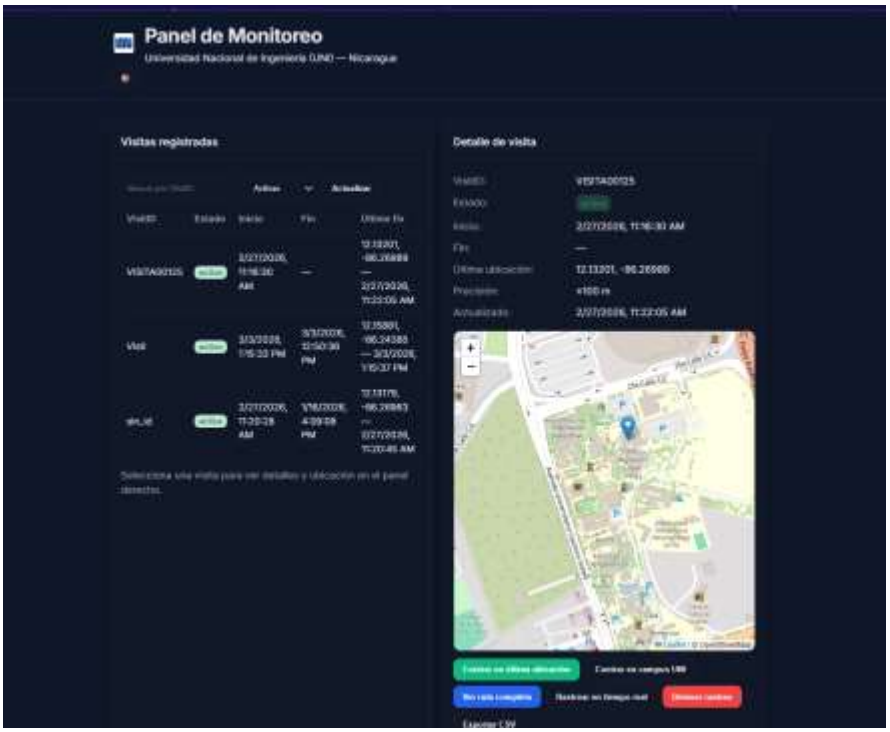


Figura 21 Panel de monitoreo

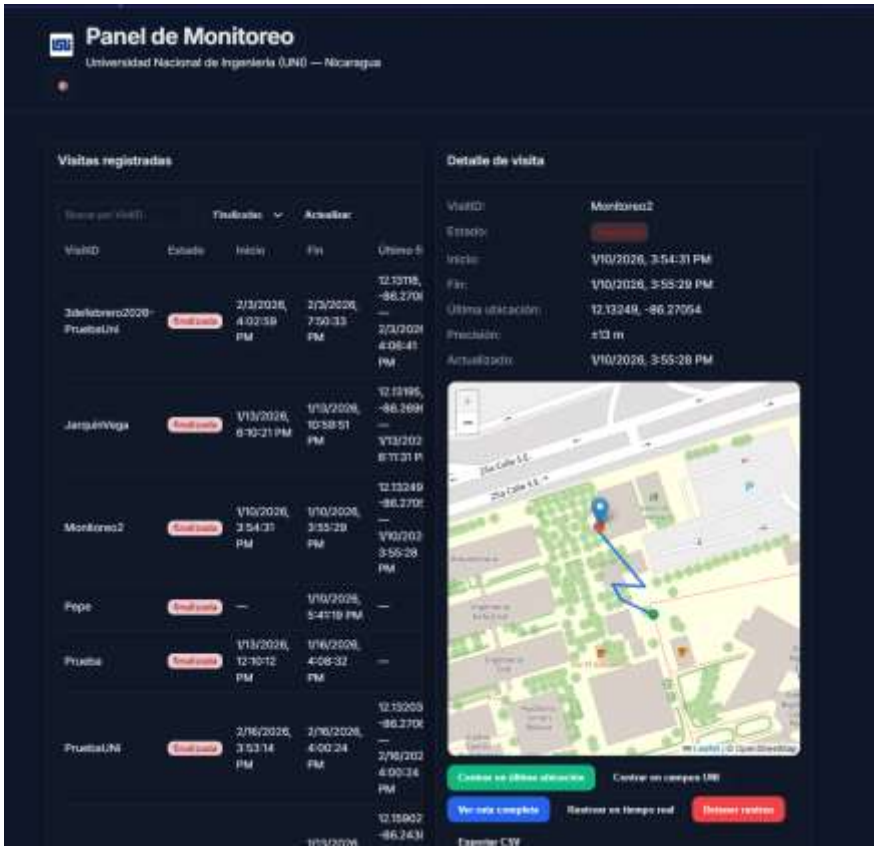


Figura 22 Trayectoria vista en tiempo real de visitante