# Diseño de un Sistema de Alarma con Botón de Pánico para los Corredores Seguros en Situaciones de Robo

Daniel Niño Villamil

Katerin Pinzón Nieto

Universitaria Agustiniana - Uniagustiniana

Facultad de Ingeniería

Ingeniería de Telecomunicaciones

Bogotá D.C.

2025

# Diseño de un Sistema de Alarma con Botón de Pánico para los Corredores seguros en Situaciones de Robo

Daniel Niño Villamil

Katerin Pinzón Nieto

Tutor:

Eliana Umbacia Betancourt

Trabajo para optar al título de Ingeniero de Telecomunicaciones

Universitaria Agustiniana - Uniagustiniana

Facultad de Ingeniería

Ingeniería de Telecomunicaciones

Bogotá D.C.

2025

# Tabla de contenido

## Lista tablas

## Lista de Figuras

## Resumen (Abstract)

El presente proyecto tiene como propósito diseñar, ensamblar y validar un prototipo de alarma portátil con botón de pánico orientado a fortalecer la seguridad en los corredores seguros universitarios y disminuyendo los tiempos de respuesta ante situaciones de robo, teniendo en cuenta que el sistema integra módulos de comunicación inalámbrica (XBee PRO S3B), geolocalización GPS y una interfaz de monitoreo implementada en Node-RED, complementada con un Chat Bot de Telegram para la notificación automática de alertas en tiempo real.

Durante este periodo de desarrollo, se llevó a cabo el diseño y ensamblaje del prototipo, seleccionando componentes electrónicos de bajo consumo energético y alta compatibilidad entre sí. Así mismo, se realizaron pruebas experimentales con antenas de diferentes ganancias entre ellas una antena omnidireccional de 5 dB y un direccional tipo Yagi de 15 dB con el fin de evaluar el alcance, la estabilidad del enlace y la precisión del localizador GPS.

Con el prototipo desarrollado se desea demostrar la factibilidad técnica y operativa de un sistema de alerta portátil basado en tecnologías IoT, diseñado para mejorar la seguridad de la comunidad universitaria al ofrecer un mecanismo de respuesta rápida, seguimiento en tiempo real y fácil acceso para los usuarios.

*Palabra clave:* Sistemas de alarma, botón de pánico, corredores seguros, geolocalización, comunicación inalámbrica, IoT (Internet de las Cosas)

**Abstract (resumen)**

The purpose of this project is to design, assemble, and validate a prototype portable alarm with a panic button aimed at strengthening security in university safe corridors and reducing response times to theft situations, taking into account that the system integrates wireless communication modules (XBee PRO S3B), GPS geolocation, and a monitoring interface implemented in Node-RED, complemented by a Telegram Chat Bot for automatic notification of alerts in real time.

During this development period, the prototype was designed and assembled, selecting low-energy consumption electronic components that are highly compatible with each other. Experimental tests were also carried out with antennas of different gains, including a 5 dB omnidirectional antenna and a 15 dB Yagi directional antenna, in order to evaluate the range, link stability, and accuracy of the GPS locator.

The prototype was developed to demonstrate the technical and operational feasibility of a portable alert system based on IoT technologies, designed to improve the safety of the university community by offering a rapid response mechanism, real-time tracking, and easy access for users.

Keyword: Alarm systems, panic button, safe corridors, geolocation, wireless communication, IoT (Internet of Things)

## Introducción

Hoy en día la inseguridad ha ido creciendo de forma alarmante ya que afecta a jóvenes estudiantes, docentes, y personal administrativo, ya que al salir de las instalaciones universitarias se ven expuestos especialmente en horas de la tarde y de la noche en sitios más concurridos por los estudiantes como lo son los paraderos del transporte público y los parques cercanos, es por eso que el proyecto propone un diseño y prototipo de un sistema de alarma la cual está equipada con un botón de pánico, en donde la integración de tecnologías de geolocalización (GPS) y comunicación inalámbrica (XBee) con el fin de ofrecer una herramienta eficiente para la activación de alertas en situaciones de robo en donde se busca reducir la vulnerabilidad de los estudiantes y además poder llegar a tener una red de apoyo institucional.

La investigación se basa en el análisis de las necesidades en términos de seguridad, evidencias por el incremento de incidentes delictivos en entornos universitarios, y en la revisión del arte de dispositivos de alerta personal. Así, el sistema planteado no solo representa una innovación tecnológica, sino que también se alinea con las demandas de protección y la búsqueda de soluciones eficientes en el ámbito de la seguridad universitaria.

## Problema

La inseguridad en Bogotá D.C se ha consolidado como una de las principales preocupaciones ciudadanas, especialmente en lo relacionado con los robos en donde el hurto a personas es el delito más recurrente en la capital, con cifras que reflejan una situación alarmante, según él Informa Anual de Seguridad 2024 elaborado por el concejal Espinosa, se registraron 130.504 casos de hurto a personas. Lo que equivale a una tasa de 1.645 por cada 100.000 habitantes, posicionando a Bogotá como la ciudad con mayor incidencia de este delito en Colombia (Espinosa, 2024).

Aunque se ha reportado una leve disminución respecto al año anterior cuando se registraron 158.747 casos la magnitud sigue siendo preocupante, además, se reportaron 9.311 hurtos de automotores, 5.241 hurtos de motocicletas y 6.036 robos a residencias (Espinosa, 2024). Estos datos evidencian que, a pesar de los esfuerzos institucionales, los mecanismos de seguridad no han logrado contener el fenómeno de manera efectiva.

El análisis por localidades muestra que el centro de Bogotá concentra los mayores índices de criminalidad, los Mártires, la Candelaria y Santa Fe lideran las tasas de hurto a personas, con 14.000, 9.569 y 8.882 casos por cada 100.000 habitantes respectivamente (Espinosa, 2024). En Chapinero se reportaron 10.924 casos, mientras que en Teusaquillo la cifra fue de 7.502, estas zonas no solo presentan una alta densidad poblacional y actividad comercial, sino también la presencia de estructuras criminales organizadas.

En contraste, localidades como suba (974), Usme (578) y Ciudad Bolívar (716) presentan tasas más bajas, aunque no están exentas de riesgos (Espinosa, 2024). La criminalidad en Bogotá no se limita a los espacios públicos, sino que también afecta entornos educativos, especialmente universidades ubicadas en zonas urbanas densamente pobladas.

Según el Informe del Departamento Nacional de Planeación (DNP, 2025), en el primer semestre del 2024 se registraron 168.553 hurtos a personas en Bogotá, lo que equivale a 931 robos diarios, lo cual ha generado una creciente percepción de inseguridad, particularmente en zonas cercanas a instituciones educativas. Los momentos del día con mayor incidencia de robos entre las 12 p.m. y la 1 p.m. y entre las 6p.m. y las 8p.m. siendo los celulares el objeto más robado (DNP, 2025).

Los estudiantes y el personal académico enfrentan riesgos diarios, especialmente en horarios nocturnos y en zonas con escasa iluminación y vigilancia. En 2023 se reportaron más de 2.100 robos a personas cerca de universidades públicas y privadas, además de 127 casos de lesiones personales (Aldana, 2024).

La Encuesta del Multipropósito del DANE también revelo que la percepción de inseguridad en el país aumento el 44% al 52.9% entre el 2022 y el 2023 y que los municipios con mayor victimización en la región metropolitana fueron Mosquera (20%), Bogotá (17%) y Soacha (14%) (Aldana, 2024).

La falta de información contundente sobre las características de estos robos, junto con la limitada disponibilidad de herramientas para reaccionar ante emergencias, aumenta el problema ya que, a pesar de implementar medidas como patrullajes esporádicos, cámaras de seguridad y rutas de atención, estos mecanismos no han sido suficientes para contener el fenómeno.

La inseguridad en Bogotá, especialmente en los entornos universitarios, representa un desafío urgente que requiere soluciones innovadoras, en donde es necesario fortalecer las estrategias tecnológicas, mejorar la infraestructura urbana y garantizar una presencia institucional efectiva que responda a las demandas de protección y tranquilidad de la población

afectada.

## Justificación

El sistema de alarma con botón de pánico para los corredores seguros surge como una solución tecnológica urgente ante la creciente inseguridad en entornos universitarios colombianos. Las estadísticas revelan que el país registra aproximadamente 931 robos diarios (Comunicados CEJ, 2024), concentrados especialmente en los horarios de salida de clases, cuando estudiantes y docentes son más vulnerables.

La efectividad del sistema radica en su capacidad para enviar automáticamente alertas de seguridad con la ubicación exacta del usuario en tiempo real, combinando tecnologías de geolocalización con comunicación inalámbrica, teniendo en cuenta un diseño versátil el cual permite implementarlo tanto en dispositivos portátiles como en puntos estratégicos de cualquier campus universitario, adaptándose a las necesidades específicas de diferentes usuarios y espacios.

Más allá de su función protectora inmediata, este sistema representa un avance en seguridad comunitaria al generar datos valiosos sobre incidentes y zonas de riesgo, información crucial para mejorar las políticas de prevención. Al reducir significativamente los tiempos de respuesta no solo protege bienes y vidas, sino que devuelve a la comunidad académica la tranquilidad y libertad de movimiento que la inseguridad ha erosionado, esta solución tecnológica se convierte así en un puente hacia entornos universitarios más seguros e inclusivos.

# Objetivos

## Objetivo General

- Desarrollar un prototipo funcional de sistema de alarma para corredores seguros capaz de transmitir en tiempo real la ubicación del usuario durante situaciones de robo, con el fin de fortalecer la seguridad y respuesta inmediata

## Objetivos Específicos

- Diseñar un prototipo de alarma para los corredores seguros que permita ver las alertas en situaciones de robo
- Ensamblar el prototipo de alarma diseñado previamente
- Probar la funcionabilidad de prototipo, mediante pruebas de validación del sistema analizando el tiempo de respuesta y la precisión del localizador

# Marco Referencial

## Estado del Arte

El desarrollo tecnológico en el campo de las telecomunicaciones ha impulsado la creación de soluciones innovadoras orientadas a mejorar la seguridad ciudadana, con un enfoque en los entornos universitarios, ya que concentran una amplia comunidad de estudiantes, docentes y personal administrativo, estas instituciones requieren sistemas que brinden protección efectiva ante situaciones de riesgo como el hurto, en este contexto, los corredores seguros universitarios hoy en día tienen mayor importancia en sistemas de alerta temprana que integren tecnologías de localización.

Dentro de este orden de ideas, Benavides & Cárdenas (2021), desarrollaron en Ecuador una investigación titulada **"**Implementación de un dispositivo IoT, basado en tecnología LoRa para la geolocalización y monitoreo fisiológico de personas en lugares turísticos"(p. 1), cuyo objetivo fue implementar un dispositivo IoT para geolocalizar y monitorear los signos vitales de los turistas en la Reserva de Producción de Fauna Chimborazo, el trabajo fue implementado teniendo en cuenta la revisión bibliográfica, el diseño e ingeniería del dispositivo, pruebas de laboratorio y validación en campo en donde los resultados demostraron que el sistema permitió geolocalizar con un margen de error de 2,5 m y medir variables fisiológicas con alta precisión, además de integrar un botón de auxilio que envía alertas en la plataforma Ubidots sin embargo se recomendó ampliar pruebas en distintos escenarios, optimizar la autonomía y estudiar la escalabilidad de la red.

Cabe resaltar que en Colombia, Quijano (2021), llevo a cabo una investigación titulada "Diseño e implementación de monitoreo de heart rate, temperatura y movimiento en población adulto mayor con sistema de alerta" (p. 1), cuyo objetivo fue diseñar un wearable de bajo costo

denominado I♥Care para monitorear signos vitales de adultos mayores y generar alertas en situaciones críticas, en este proyecto se probaron diferentes sensores y comunicación vía Sigfox en donde los resultados evidenciaron que el monitoreo era cada 20 minutos y las aletas se enviaban en menos de 2 segundos sin embargo, se recomendó mejorar la precisión en usuarios con patologías específicas, optimizar el diseño ergonómico y gestionar las certificaciones médicas.

Además, Ardila (2023), desarrollo la investigación "Estudio exploratorio para la validación del modelo startup Biky que integra tecnologías sostenibles e informáticas en beneficio de la seguridad del transporte no motorizado en la ciudad de Bogotá" (p. 1) en donde su objetivo principal fue validar el modelo tecnológico Biky, conformado por un botón de pánico portátil y una aplicación móvil para mejorar la seguridad de ciclistas y peatones, para este proyecto se basaron en una metodología basada en el método Lean Startup, con desarrollo de prototipo y encuestas de percepción, aunque los resultados mostraron la alerta en tiempo real a autoridades, contactos y usuarios cercanos en un radio de 200 metros, reduciendo los tiempos de respuesta en emergencias se recomendó masificar la herramienta, fortalecer alianzas institucionales y optimizar la integración con tecnologías complementarias como IoT y GPS.

De igual manera Cifuentes & Silva ( 2021), realizaron la investigación titulada "Desarrollo de una aplicación Android para anunciar la presencia de sobrevivientes mediante un botón de pánico" (p. 1), cuyo objetivo fue desarrollar una aplicación para apoyar la localización y rescate de personas en catástrofes naturales basando su proyecto en una metodología ágil Scrum en tres sprints, donde realizaron un diseño de una base de datos en Firebase, diagrama UML y programación de módulos de registro y envió de alertas vía Bluetooth, en donde los resultados mostraron que la aplicación permitió enviar señales sin depender de redes móviles

colapsadas, analizando distintas distancias de hasta 100 metros, teniendo en cuenta esto se recomendó realizar más pruebas de validación en diferentes escenarios y optimizar la compatibilidad con nuevas versiones Android.

Por el contrario en México, Espinosa & Retana (2022), desarrollaron una investigación titulada " Prototipo de geolocalización para personas vulnerables: botón de pánico SOS" (p. 1), en el cual se enfocaron en diseñar un sistema de geoposición de bajo costo integrado en una pulsera para víctimas de secuestros o desapariciones, utilizando una metodología experimental con Arduino UNO, modulo GPS y GSM, realizando pruebas de laboratorio y de campo en la Ciudad de México, visualizando que el prototipo envía mensajes SOS con enlace a Google Maps para poder visualizar la ubicación de la persona con una precisión de ± 2,5 m, aunque presento limitaciones por interferencias arquitectónicas, es por esta razón que se recomendó mejorar el módulo GPS, optimizar el rendimiento en áreas urbanas y explorar mecanismos de implementación comercias.

Asimismo Añezco & Sanchez (2023), en Ecuador, realizaron el trabajo titulado "Diseño e implementación de un sistema de alerta de emergencias con ubicación GPS" (p. 1), en donde se enfocaron en diseñar una mochila de emergencia que integra un módulo GPS, un módulo GPRS para el envío de alertas y un registro en Firebase, en donde se enfocaron primero en el desarrollo del hardware, después la aplicación móvil y por último la base de datos sin embargo aunque comprobaron la operatividad del sistema para enviar mensajes con ubicación en tiempo real se recomendó ampliar el alcance del bluetooth que es el que transmite los datos a la aplicación móvil y fortalecer la integración de organismos de respuesta

Sin embargo, Salazar (2011), desarrollo un proyecto titulado "Diseño de un sistema de localización y seguimiento de personas" (p. 1), en donde se centraron en crear un sistema

integrado capaz de localizar y hacer seguimiento a las personas mediante un módulo GPS, una aplicación móvil y un servidor externo permitiendo el envío y registro periódico de posiciones pero la duración de la batería es muy corto por el consumo/ gestión de la batería es por eso que se recomienda realizar más pruebas teniendo en cuenta la duración de la batería en distintos escenarios.

En cambio Vélez (2023), desarrollo un proyecto titulado “Ataraxia” (p. 1), cuyo objetivo principal es diseñar una solución integral que combine una pieza de joyería vestible y una plataforma digital para generar una comunidad y así prevenir y alertar oportunamente sobre situaciones de violencia y además poder educar y poder visualizar la problemática desde cada punto de vista como resultado se creó una joya con un botón en donde solo las personas que lo usen sepan que es un botón, se recomienda continuar con las pruebas de campo, mejorar la ergonomía y establecer alianzas con entidades para asegurar la efectividad del servicio.

De igual manera Gutarra & Chunga (2022), en Perú, propusieron una solución tecnológica HelPoint, creo una pulsera con botón de pánico vinculada a una aplicación móvil que envía alertas y geolocalización en situaciones de robo en donde tuvieron un enfoque aplicado teniendo en cuenta las entrevistas y el análisis contextual de la inseguridad en Lima, se concluyó que este tipo de dispositivos pueden compensar limitaciones de la respuesta policial, siendo aceptados por su discreción y potencial de salvar vidas en escenarios de peligro.

De la misma forma Acero, et al., (2024), Desarrolló una investigación titulada “Pulseras Securelet” (p. 1), en donde diseñaron un plan de negocio para una pulsera con GPS y botón de pánico vinculada a una aplicación móvil, a través de un análisis de mercado en donde identificaron un alto interés por el producto por parte de padres de familia y adultos preocupados por su seguridad.

En México Altamirano, Rafael, et. al (2016), en el artículo “Prototipo de dispositivo de alerta “Ay Ta” (p. 1), presentaron un gadget localizador adaptable en forma de pulsera o collar, dirigido a la protección de infantes, mascotas y objetos personales, el dispositivo está basado en la comunicación WiFi y Bluetooth, emitiendo alertas inmediatas y permitiendo el registro de incidentes en una plataforma web con mapas interactivos, como resultado se comprobó la eficiencia del dispositivo para reducir riesgos de robo, recomendando optimizar la autonomía del dispositivo y ampliar pruebas de campo

Por otra parte, Montesdeoca (2022), desarrollo un proyecto de software titulada “Desarrollo de aplicación móvil de geolocalización para alarma y notificación de pánico en plataforma Android” (p. 1), la cual se basó en desarrollar una aplicación para Android que permita activar una alarma sonora y al mismo tiempo enviar la ubicación GPS a contactos de confianza para ello utilizo una metodología ágil en donde uso herramientas como Node.js, MongoDB y Twilio, al final demostró la funcionalidad de la app pero se recomienda optimizar la interfaz, mejorar la gestión de mensajes y vincular la aplicación con servicios de emergencia.

Así mismo Parra( 2023), en su tesis “Desarrollo de una aplicación móvil tipo botón de pánico para la comunidad de Ainche”(p. 1), diseño e implemento una aplicación móvil con geolocalización y alertas de emergencia con el fin de mejorar la seguridad de comunidades rurales, en donde mediante un enfoque aplicado, integrando Android Studio, Firebase y APIs de geolocalización, se validó la utilidad de la herramienta con alta satisfacción de los usuarios, quienes destacaron la facilidad de uso y rapidez de respuesta pero se recomendó su difusión comunitaria, actualizaciones periódicas, alianzas con autoridades locales y posible conexión con otras comunidades.

Por el contrario Gordon & Martínez (2019), llevaron a cabo un estudio “Diseño de red LoRaWAN para el servicio de un botón de pánico en Barranquilla” (p. 1), cuyo objetivo fue desarrollar un sistema de comunicación de bajo consumo que transmitiera la ubicación desde un botón de pánico. Mediante simulaciones de cobertura, desarrollo de prototipo y pruebas de campo, se evidencio la factibilidad técnica del sistema, aunque se presentaron limitaciones de autonomía energética por esta razón se recomendó integrar técnicas de cosecha de energía y optimizar el microcontrolador para mejorar la escalabilidad y sostenibilidad del servicio.

Teniendo en cuenta esto es importante resaltar otros escenarios en donde han explorado aplicaciones específicas como en Ecuador donde Romero & Torres (2022), desarrollaron e implementaron un sistema de seguridad para camiones transportistas basado en un botón de pánico y un módulo GPS, logando un rastreo en tiempo real y emisión de alertas. Asimismo, Saraguro (2020), en su trabajo diseño un brazalete de alerta temprana contra ataques de feminicidio mediante GPS, logrando el envío de mensajes SOS, además Guaman& Orquera (2020), en Ecuador, diseñaron una aplicación móvil de botón de socorro, mientras que Olmedo (2023), implemento un tótem de botón de pánico para campus universitarios con alimentación autónoma y Garzón (2024), aplico un sistema IoT de seguridad con botón de pánico para locales comerciantes.

Los corredores seguros han sido un tema que aún no es muy popular pero poco a poco han ido tomando fuerza como en México en donde el proyecto de Macías (2020), titulada “Espacios para la no violencia: el caso del Corredor Seguro para Mujeres en Ciudad Juárez en 2020” (p. 1), tuvo una metodología cualitativa sustentada en entrevistas, análisis documental y revisión de políticas públicas en donde sus hallazgos mostraron que las mujeres perciben el espacio público como inseguro debido a factores estructurales (iluminación deficiente, ausencia

de vigilancia activa, aceras en mal estado) y que las intervenciones físicas (iluminación, cámaras, rutas seguras) deben complementarse con políticas de prevención centradas en género, participación comunitaria y presupuesto municipal dedicado sin embargo se recomendó incorporar enfoques CPTED con perspectiva de género, la educación comunitaria y la coordinación interinstitucional para garantizar sostenibilidad y legitimidad.

Por parte de Passarelli (2015), en la tesis "Corredor seguro: Uso, disputas y apropiación del espacio público en la ciudad de La Plata" (p. 1), aportó evidencia empírica sobre los efectos sociales de la implementación de los corredores seguros, empleando entrevistas semiestructuradas a vecinos y comerciantes, observación no participante y análisis de documentos municipales, la autora describió como las medidas (instalación de cámaras DOMO, incremento de alumbrado, poda de árboles, botones antipánico) transformaron el espacio público y a la vez que redujeron ciertos riesgos percibidos, generaron procesos de "resignificación" del barrio y disputas sobre la privatización parcial del espacio público, en este proyecto se recomendó evaluar el impacto social de estas intervenciones, mantener la participación vecinal y evitar que la seguridad situacional sustituya políticas sociales más amplias.

Además, en Colombia Suarez (2021), desarrollo un proyecto de grado titulado "Privatización de la seguridad en Bogotá: el caso de los corredores universitarios seguros y el rol de las empresas de seguridad privada" (p. 1), cuyo objetivo fue analizar el papel de las empresas de seguridad privada dentro del programa de corredores seguros universitarios en el centro de Bogotá D.C., la investigación empleo entrevistas semiestructuradas con jefes de seguridad de universidades y representantes de empresas privadas, aplicando un análisis coaxial con codificación abierta y teórica sin embargo en los resultados se evidencio que los barrios circundantes, los vendedores ambulantes y la vulnerabilidad estudiantil son los principales focos

de riesgo además la ejecución de las estrategias depende de la operación entre la policía y los manejadores caninos de las empresas privadas ya que los corredores seguros incrementaron la percepción de seguridad aunque desplazaron el delito a otras zonas teniendo en cuenta esto se concluyó que la seguridad privada cumple un rol complementario en la provisión de seguridad pública, siendo fundamental la coordinación interinstitucional, la definición de alineamientos claros sostenibilidad del programa.

En conjunto estas investigaciones muestran una tendencia regional e internacional hacia la integración de dispositivos portátiles, aplicaciones móviles, sistemas IoT y soluciones hibridas que combinan geolocalización, comunicación inalámbrica y monitoreo en tiempo real, en donde los hallazgos coinciden en que los botones de pánico y dispositivos de alerta representan herramientas eficaces de prevención y respuestas especialmente en contextos urbanos, rurales y universitarios, asimismo, se destaca la necesidad de optimizar la autonomía energética, robustecer las comunicaciones, validar la usabilidad en distintos grupos de usuarios y fortalecer los vínculos de seguridad para garantizar su efectividad y sostenibilidad en la práctica.

## Marco Teórico

Para comprender el funcionamiento del sistema en esta investigación, es necesario definir conceptos claves que sustentan su desarrollo y aplicación, con el fin de explicar no solo el propósito del sistema, sino que también contextualizan la tecnología utilizada.

### Corredores Seguros

La estrategia de corredores seguros consiste en la delimitación de corredores geográficos generalmente cercanos a instituciones educativas o zonas de alto transito donde se refuerzan medidas de vigilancia, control y prevención del delito, con el fin de reducir la incidencia delictiva y mejorar la percepción de seguridad de la comunidad, por ejemplo se han

implementado patrullajes coordinados, cámaras de vigilancia, mejoras en la iluminación para disminuir los robos y agresiones (Secretaria de Seguridad, Convivencia y Justicia de Colombia, 2024), es por eso que en este proyecto se entiende “Corredores seguros” como el entorno físico en donde se desplegara el sistema de alarma, el cual tiene como objetivo fortalecer la seguridad en las rutas de acceso, espacios exteriores y paraderos utilizados por estudiantes y la comunidad académica.

### Botón de pánico

El botón de pánico es un dispositivo de seguridad diseñado para activar alertas de emergencia de manera rápida y discreta, su principal función es evitar una seña de auxilio generalmente acompañada de información sobre la ubicación de usuario, a contactos de confianza o a las autoridades competentes (Acecho Seguridad, 2024).

Este tipo de dispositivos pueden ser físicos los cuales vemos en relojes llaveros o digitales con el fin de poder tenerlos en el celular, relojes inteligentes, entre otros; esto depende mucho de la situación en la que se requiera implementar este dispositivo teniendo en cuenta los requisitos que se deban cumplir.

Los botones de pánico han demostrado ser herramientas efectivas en la prevención y atención de emergencias, especialmente en casos de acoso, violencia, robos y secuestros, es decir, que estos dispositivos son muy versátiles ya que se han adaptado ampliamente en sistemas de seguridad ciudadana y corporativa, siendo implementados en distintos sectores como por ejemplo en la protección de trabajadores solidarios, seguridad escolar, asistencia a adultos mayores.

### Geolocalización

La geolocalización es el proceso de determinar la ubicación exacta de un objeto o persona en tiempo real mediante el uso de tecnologías como el GPS (Sistema de Posicionamiento Global), redes celulares GSM (Sistema Global para comunicaciones Móviles) o protocolos de comunicación de baja potencia como LoRa.

Por lo general el GPS es la tecnología más utilizada en sistemas de rastre, ya que permite obtener coordenadas precisas a través de señales satelitales, sin embargo, su efectividad puede verse afectada en entornos urbanos densos, donde edificios y otras estructuras pueden inferir por la señal; por otro lado, el uso de redes móviles GSM permite la transmisión de la ubicación en tiempo real a servidores remotos o dispositivos móviles lo que facilita la rápida asistencia a la persona en peligro (Coder, 2023).

En este proyecto la geolocalización juega un papel fundamental, ya que permite que el sistema envié la ubicación del usuario a contactos de emergencia cada cierto intervalo de tiempo asegurando que puedan rastrear su movimiento en caso de peligro

### Internet de las Cosas (IoT)

El IoT (Internet de las Cosas) se refiere a la interconexión de dispositivos electrónicos a través de internet o redes privadas, permitiendo la recopilación y el intercambio de datos en tiempo real, hoy en día la aplicación del IoT ha ido en aumento más que todo en temas de seguridad personal en donde ha revolucionado el desarrollo de dispositivos inteligentes capaces de detectar situaciones de robo y actuar de manera autónoma (Red Hat Inc, 2023).

En el proyecto la integración de IoT permitirá que el sistema de alarma se comunique con el servidor central (Raspberry Pi) almacenando la información refleja en el dashboard y notificando a contactos de emergencia de manera automática, además la implementación de esta

tecnología posibilita la actualización del sistema en el futuro mejorando su capacidad de respuesta y logrando mayores alcances en la cobertura.

### Dispositivos vestibles (Wearables)

Los dispositivos vestibles son tecnologías diseñadas para ser utilizadas como accesorios o prendas de vestir, proporcionando funciones adicionales como monitoreo de la salud, comunicación o seguridad con el fin de que las personas se sientas cómodas al momento de cargar el dispositivo ya que muchas buscan siempre estar a la moda, pero usando tecnología que los ayuden en sus tereas del día a día. (Banco Santander S.A, 2022)

En este proyecto el dispositivo con botón de pánico, es ergonómico fácil de cargar diseñado para ser discreto, accesible y cómodo para el usuario con el fin que siempre lo lleve con el sin importar la ocasión

### Seguridad y Prevención del delito

La seguridad personal se ha convertido en una prioridad social, especialmente en entornos donde la incidencia delictiva es alta, a lo largo de los años se han desarrollado diferentes estrategias con el fin de reducir el riesgo de ser víctima de algún crimen, por ejemplo, la implementación de corredores seguros, vigilancia comunitaria y el uso de tecnologías para la protección individual (Robayo, 2023).

### Chat Bot

Un chat Bot es un programa automatizado que interactúa con usuarios a través de una interfaz de mensajería, la cual funciona como una extensión del sistema de comunicación, integrándose con interfaces personalizadas para poder recibir ubicaciones, stickers, archivos, mensajes de voz entre otros. (Telegram, 2025).

Este mecanismo permite que el sistema reaccione en tiempo real, gestionando múltiples formatos de datos (texto, ubicación, entre otros) y ofreciendo una interfaz de usuario simple y fácil de usar, es por esta razón que se ha ido implementando en sistemas de emergencias, mejorando el tiempo de respuesta del sistema y facilitando el uso en entornos de alta vulnerabilidad. (Telegram, 2025)

## Marco Legal

El diseño y prototipo de un sistema de alarma con botón de pánico en los corredores seguros universitarios deben desarrollarse bajo el marco normativo colombiano, considerando las disposiciones relacionadas con la seguridad ciudadana, la responsabilidad institucional y el uso de tecnologías de información y comunicación (TIC)

### Artículo 2 de la Constitución Política

El artículo 2 de la constitución política dice que “proteger a todas las personas residentes en Colombia en su vida, honra, bienes, creencias y demás derechos y libertades” (Escobar, 1991), este artículo establece el fundamento legal para promover e implementar sistemas tecnológicos de prevención y respuesta ante situaciones que amenacen la seguridad individual y colectiva, como lo son los robos.

### Artículo 15 de la Constitución Política

Asimismo, el artículo 15 reconoce el derecho a la intimidad y a la protección de datos personales, lo que implica que cualquier sistema que recopile o transmita información por ejemplo ubicación o identidad del usuario debe garantizar la confidencialidad, seguridad y tratamiento adecuado de los datos. (Escobar, 1991)

### Artículo 15 de la Constitución Política

Asimismo, el artículo 15 reconoce el derecho a la intimidad y a la protección de datos personales, lo que implica que cualquier sistema que recopile o transmita información por ejemplo ubicación o identidad del usuario debe garantizar la confidencialidad, seguridad y tratamiento adecuado de los datos. (Escobar, 1991)

### Ley 1801 de 2016 del Código Nacional de Seguridad y Convivencia Ciudadana

La ley 1801 de 2016 busca asegurar la convivencia pacífica y la protección de la vida mediante mecanismos preventivos y el uso de tecnologías, en el artículo 8 se establecen los deberes de las autoridades para promover entornos seguros y prevenir la comisión de delitos, es decir, que la norma faculta el uso de herramientas tecnológicas que faciliten la reacción inmediata ante situaciones de emergencia (Congreso de la Republica de Colombia, 2016).

### Ley 1621 de 2013 del Código Nacional de Seguridad y Convivencia Ciudadana

La ley 1621 de 2013 regula las actividades de vigilancia y monitoreo de información en donde su aplicación se orienta a organismos de inteligencia y establece los principios de legalidad, necesidad y proporcionalidad en el uso de tecnologías que recopiles datos por lo tanto los sistemas de alarma con geolocalización deben respetar la privacidad y solo utilizar la información con fines de seguridad (Congreso de la Republica de Colombia, 2013)

### Normas sobre Telecomunicaciones y TIC

El ministerio de Tecnologías de la información y las Comunicaciones (MinTIC) mediante la ley 1341 de 2009, promueve el uso eficiente de las TIC para el desarrollo social y la seguridad ciudadana, es decir, que la ley impulsa la innovación tecnología con responsabilidad social, asegurando que las soluciones digitales se implementen respetando los principios de inclusión transparencia y protección del usuario (Congreso de la Republica de Colombia, 2009)

Asimismo, el Decreto 1078 de 2015 copila las normas del sector TIC y regula la infraestructura y servicios de telecomunicaciones, permitiendo la transmisión de datos a través de redes inalámbricas y sistemas IoT bajo parámetros de seguridad y confiabilidad (Ministerio de Tecnologias de la Infotmacion y las Comunicaciones, 2015)

# Metodología

## Análisis de necesidades

La situación actual en los alrededores de las universidades presenta un aumento alarmante en los incidentes de robo y diversos actos delictivos, lo que ha desencadenado un ambiente de constante preocupación e inseguridad entre los estudiantes y otros miembros de las comunidades universitarias,  ya que esta problemática no solo afecta la percepción de seguridad en los espacios cercanos a las instituciones, sino que también tiene un impacto negativo en el bienestar emocional y psicológico de los estudiantes, quienes enfrentan el desafío diario de convivir con la posibilidad de ser víctimas de estos eventos

La sensación de vulnerabilidad va en aumento por la carencia de mecanismos de alerta inmediata y dispositivos de emergencia que sean accesibles y eficaces ante situaciones de robo en entornos donde los incidentes delictivos puedan ocurrir de forma repentina como en los parques cerca a las instituciones o los paraderos de buses de transporte público, teniendo en cuenta esto si no se cuenta con un sistema que permita la comunicación inmediata y precisa en momentos críticos, la respuesta ante estos incidentes se vuelve lenta o simplemente no existe, lo que puede derivar consecuencias más graves o peligros prolongados

Es evidente que la combinación de estos factores ha creado un entorno en el que la percepción de inseguridad se encuentra en aumento, afectando no solo la vida de los estudiantes, sino también la confianza general en el entorno que debería ser propicio para el aprendizaje y el desarrollo personal, por ello se hace imprescindible adoptar medidas que fortalezcan la protección de la comunidad, promoviendo soluciones integrales que reduzcan esta creciente sensación de inseguridad.

**Requisitos**

Para garantizar la efectividad del sistema de alarma basado en un botón de pánico y GPS, se establecen los siguientes requisitos:

- Alcance de comunicación de 200 hasta 300 metros.
- Tiempo de respuesta menor a 10 segundos.
- Funcionamiento en interiores y exteriores.
- Alta precisión en la ubicación GPS.
- Dispositivo que sea fácil de usar y cómodo de llevar
- Capacidad de 30 usuarios

**Diseño de la Arquitectura**

Se realizaron los respetivos diagramas para poder visualizar de manera clara y ordenada el funcionamiento del sistema teniendo en cuenta los requisitos con el fin de representar la interacción de los distintos módulos del hardware y software que conforman el sistema de alarma, facilitando la compresión del flujo de información, las interconexiones físicas y la secuencia de eventos que ocurren desde la generación de la alerta hasta su recepción y notificación en el servidor central y el Chat Bot en Telegram

Durante esta etapa, se elaboraron distintos diagramas los cuales fueron fundamentales para definir con precisión las relaciones entre los componentes electrónicos (microcontrolador ESP32 C3 Super mini, modulo GPS, sensor INA3221, módulos XBee PRO S3B, el módulo de carga, los LEDs indicadores y la batería) así como los módulos de software involucrados con el servidor central (Node-Red) y el Chat Bot de Telegram con el fin de garantizar la integración entre las diferentes capas del sistema, optimizando la comunicación, la gestión energética y la transmisión de datos.

**Diseño de la Detallado**

En esta etapa se definieron los componentes específicos y las conexiones eléctricas necesarias para el funcionamiento del sistema en donde se realizaron diagramas de las interconexiones del hardware asegurando que cada subsistema desempeñara adecuadamente las funciones asignadas según los requerimientos establecidos en las fases previas del proyecto

En el diseño detallado se tuvieron en cuenta criterios técnicos como el consumo energético, los niveles de voltaje, la capacidad de comunicación, la precisión de los sensores y la facilidad dentro del dispositivo portátil, configurando cada módulo por separado por medio del microcontrolador para al final integrar todos los códigos en uno solo con el fin de minimizar los errores en la configuración del sistema.

**Implementación**

Se construyó el prototipo físico de un sistema de alarma, integrando los módulos de hardware seleccionados en las etapas previas de diseño teniendo en cuenta la estructura del dispositivo para poder garantizar la portabilidad y facilidad de uso para los usuarios, además se estableció la conexión con el servidor central (Raspberry pi) mediante radiofrecuencias para el envío y procesamiento de alertas en tiempo real priorizando la comunicación inalámbrica entre el dispositivo portátil y el servidor central.

En el servidor central se configuro la plataforma Node-RED con el fin de crear un entorno de programación visual para la gestión integral de alertas en donde se diseñó un flujo principal compuesto por nodos que permiten la recepción de los datos desde el módulo XBee Maestro

**Pruebas Unitarias**

En esta etapa se verifico el funcionamiento individual de cada módulo y subsistema del proyecto, asegurando su correcto desempeño conforme a los parámetros técnicos definidos con el fin de minimizar los errores por parte de los dispositivos; se realizaron las pruebas de la siguiente manera:

- **Microcontrolador ESP32 C3 Super mini:** Se evaluó la correcta ejecución del código, la gestión eficiente de energía, la comunicación serial y el envío de datos hacia el módulo XBee
- **Modulo GPS:** Se comprobó la precisión de la ubicación (±2.5 metros) y el tiempo promedio de adquisición de coordenadas (< 3 segundos)
- **Modulo XBee PRO S3B:** se probó la transmisión y recepción de datos con un alcance entre los 5m y 30m
- **Modulo INA3221:** medición del voltaje y corriente de la batería, comparando los valores obtenidos con instrumentos de referencia (multímetro)
- **Módulo de carga TP4056:** verifico que el módulo cargue la batería solo cuando el dispositivo este apagado
- **LEDs indicadores**
  - Encendido: confirma el suministro de energía
  - Activación de la Alarma: se enciende al presionar el botón
  - Nivel de la batería: se enciende al INA3221 reportar que la batería esta descargada
- **Chat Bot de Telegram:** recepción inmediata de la alerta con ubicación GPS

- **Servidor Central:** gestión de los datos para la visualización de las alertas en el dashboard
- **Botón de pánico:** se verifico la sensibilidad y tiempo de respuesta inmediato

**Pruebas de Integración del Sistema**

En un entorno controlado de laboratorio, se validó la interoperabilidad de todos los módulos, verificando que la comunicación y transmisión de la información se efectúa sin errores.

- **Comunicación dispositivo- servidor:** Transmisión exitosa de alertas mediante los módulos XBee PRO S3B
- **Servidor- Chat Bot:** recepción de datos y envío de alertas con la ubicación
- **Dashboard Node-Red:** visualización en tiempo real de la ubicación de la alerta
- **Tiempo de respuesta total:** Menor a 10 segundos

**Pruebas de Validación**

Se ejecutaron pruebas de campo en los alrededores de la Uniagustiniana, evaluando el sistema con 1 dispositivo en condiciones reales como las siguientes:

- Alcance de comunicación: pruebas con distancias de 10m, 20m, 30m y 50m con antenas de distintas ganancias
- Precisión del GPS: Comparación de coordenadas reales del dispositivo
- Estabilidad del enlace: Observación del desempeño ante obstáculos físicos

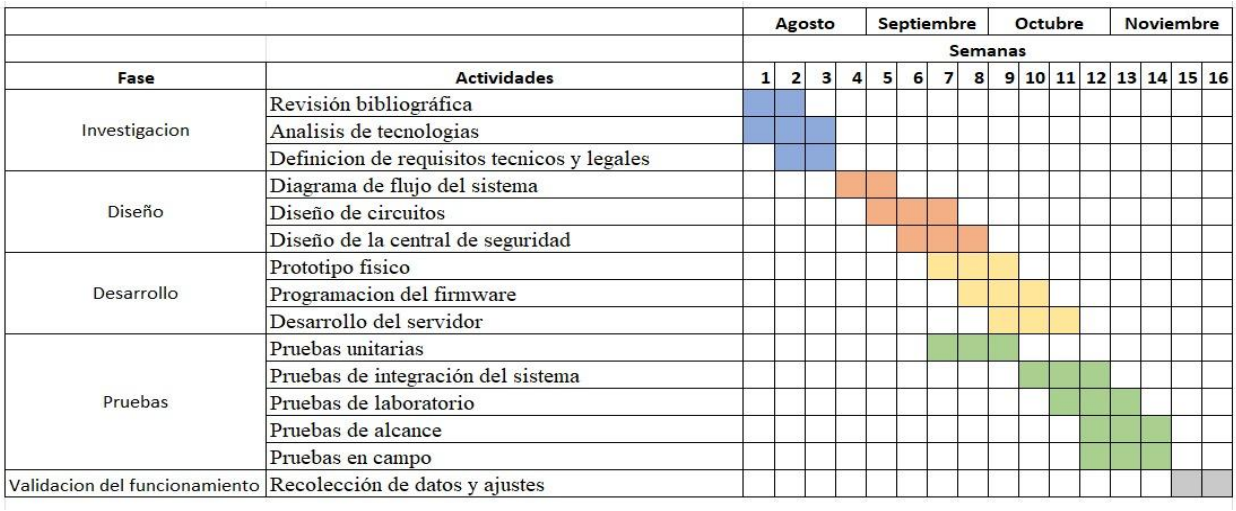
## Administración del proyecto

### Cronograma

En la figura 1 se planteó el cronograma el cual permite visualizar la duración y orden de las actividades, así como la interdependencia entre ella, de igual manera posibilita un seguimiento continuo del avance del proyecto, lo que contribuye a la identificación oportuna de retrasos y estableciendo acciones correctivas que aseguren el cumplimiento de los objetivos

**Figura 1**

*Diagrama de Gantt*



| | | Agosto | | | | Septiembre | | | | Octubre | | | | Noviembre | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Semanas | | | | | | | | | | | | | | | |
| **Fase** | **Actividades** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| Investigacion | Revisión bibliográfica | | | | | | | | | | | | | | | | |
| | Analisis de tecnologias | | | | | | | | | | | | | | | | |
| | Definicion de requisitos tecnicos y legales | | | | | | | | | | | | | | | | |
| Diseño | Diagrama de flujo del sistema | | | | | | | | | | | | | | | | |
| | Diseño de circuitos | | | | | | | | | | | | | | | | |
| | Diseño de la central de seguridad | | | | | | | | | | | | | | | | |
| Desarrollo | Prototipo fisico | | | | | | | | | | | | | | | | |
| | Programacion del firmware | | | | | | | | | | | | | | | | |
| | Desarrollo del servidor | | | | | | | | | | | | | | | | |
| Pruebas | Pruebas unitarias | | | | | | | | | | | | | | | | |
| | Pruebas de integración del sistema | | | | | | | | | | | | | | | | |
| | Pruebas de laboratorio | | | | | | | | | | | | | | | | |
| | Pruebas de alcance | | | | | | | | | | | | | | | | |
| | Pruebas en campo | | | | | | | | | | | | | | | | |
| Validacion del funcionamiento | Recolección de datos y ajustes | | | | | | | | | | | | | | | | |

*Nota:* Autoridad propia (2025)

### Presupuesto

El presupuesto se elaboró con base en un análisis detallado de los requerimientos técnicos, priorizando la calidad y funcionabilidad del dispositivo a un bajo costo, de esta manera se busca garantizar el equilibrio entre la inversión realizada y los resultados obtenidos.

**Tabla 1**

*Presupuesto*

| Componente | Cantidad | Precio Unitario (COP) | Precio Total (COP) |
|---|---|---|---|
| Microcontrolador ESP32 C3 Super Mini | 1 | 25.000 | 25.000 |
| Botón de Pánico | 1 | 500 | 500 |
| Modulo GPS Neo 8m | 1 | 62.900 | 62.900 |
| INA 3221 | 1 | 16.900 | 16.900 |
| Módulo de carga TP4056 | 1 | 5.000 | 5.000 |
| Módulo XBee PRO S3B | 2 | 455.008 | 910.016 |
| Resistencias de 220 Ω | 3 | 100 | 300 |
| Resistencias de 10 kΩ | 1 | 100 | 100 |
| LED | 3 | 200 | 900 |
| Interruptor | 1 | 700 | 700 |
| Regleta hembra | 2 | 1.700 | 3.200 |
| Regleta Macho | 3 | 1.000 | 3.000 |
| Cables Jumpers | 7 | 300 | 2.100 |
| Placa PCB | 1 | 38.000 | 38.000 |
| Impresión 3D (Carcasa) | 1 | 160.000 | 160.000 |
| Raspberry Pi | 1 | 446.000 | 446.000 |
| Micro SD | 1 | 16.000 | 16.000 |
| Batería Lipo | 1 | 25.000 | 25.000 |
| Total | | | 1.698.716 |

*Nota:* Precios tomados de Mercado Libre y Electronilab (2025)

# Desarrollo

## Objetivo 1. Diseño del prototipo

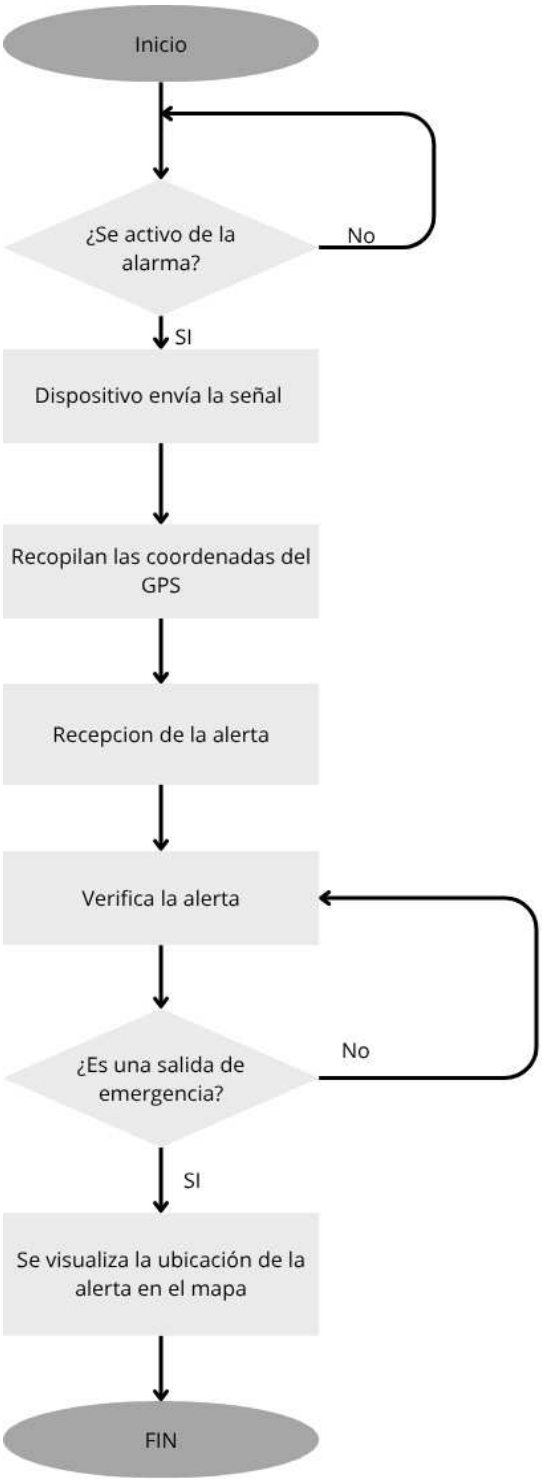
El sistema propuesto se basa en una arquitectura cliente-servidor diseñado para la gestión de alertas de situaciones de robo en tiempo real, optimizando la comunicación entre el dispositivo portátil del usuario y el servidor central encargado del procesamiento de alertas, con el fin de garantizar la transmisión confiable de información y la integración segura de los diferentes componentes tecnológicos que conforman el sistema

La arquitectura se compone de tres niveles principales:

- Dispositivo portátil del usuario el cual es el encargado de la captura de datos, la activación del botón de pánico y el envío de la información
- Servidor central es el que actúa como nodo maestro y es el que se encarga de recibir, procesar y almacenar las alertas
- Interfaz de usuario y sistema de notificación, implementada en Node-RED y un Chat Bot de Telegram, que visualiza la información y envía alertas automáticas a los contactos de seguridad

***Figura 2***

*Diagrama de flujo del sistema*




*Nota:* autoridad propia (2025)

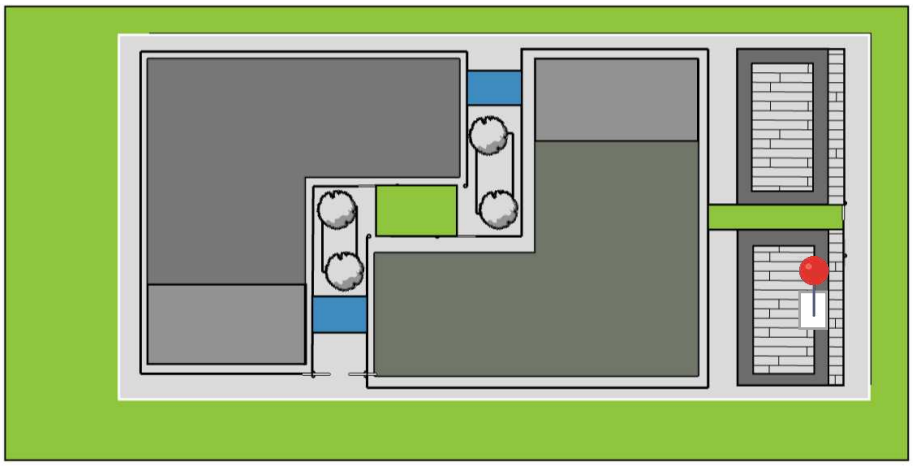
El diagrama de flujo del sistema (figura 2) ilustra el proceso paso a paso desde la activación de la alerta hasta la respuesta de emergencia, desde la activación de la alarma hasta la visualización en el servidor central, el proceso inicia cuando el usuario presiona el botón físico del dispositivo, momento el cual el sistema verifica si la alarma ha sido activada correctamente, una vez confirmada se hace la transmisión al servidor central para su procesamiento

Posteriormente, el sistema recopila las coordenadas GPS en tiempo real, asociándolas con los datos del dispositivo para asegurar la precisión de la ubicación, después una vez que la alerta es recibida por el servidor central, esta se procesa y se visualiza en el mapa y en la base de datos del servidor central.

Teniendo esto en cuenta esto se elaboró un mapa para poder visualizar el posicionamiento del servidor central (figura 3) que muestra la ubicación estratégica del servidor central dentro de los corredores seguros universitarios, el punto de instalación fue definido considerando la proximidad de las principales entradas y zonas con mayor flujo estudiantil, optimizando la recepción de las señales y visibilidad de las aletas

***Figura 3***

*Universidad Ficticia*

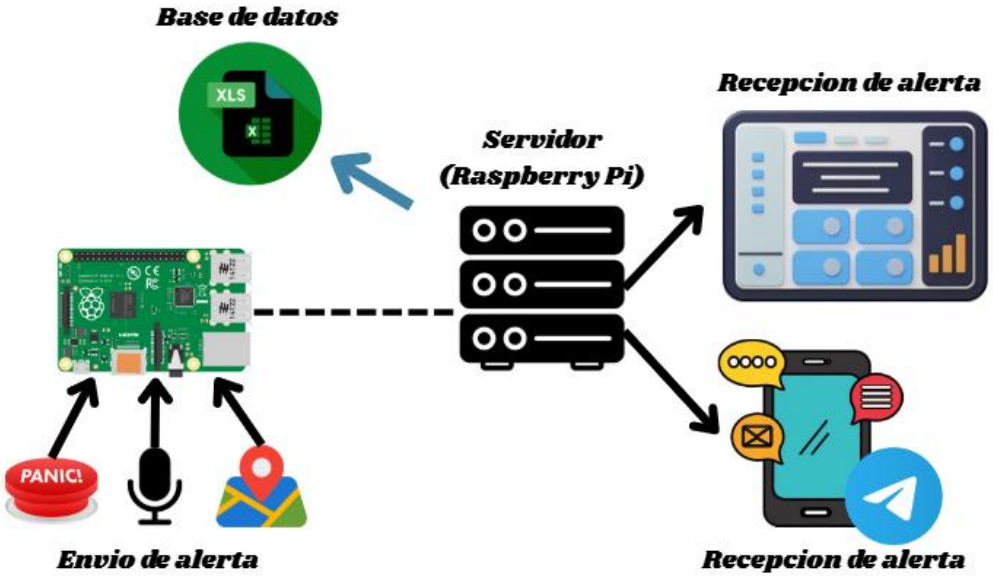


*Nota:* autoridad propia (2025)

En el mapa permite una visualización geográfica del área de influencia del sistema, facilitando la planeación de futuras expansiones hacia otros sectores del campus o en diferentes instituciones universitarias.

**Figura 4**

*Diseño de la Arquitectura*




*Nota*: autoridad propia (2025)

El sistema de emergencia universitario representado en la figura 4 funciona como un circuito cerrado de protección que conecta a los estudiantes con el servidor central en cuestión de segundos sin embargo cabe resaltar que todo comienza con un sistema de alarma en donde primero el usuario debe presionar el botón, estos dispositivos no solo envían la señal de alarma, si no que transmiten la ubicación exacta gracias a un sistema GPS el cual funciona tanto en exteriores como en interiores.

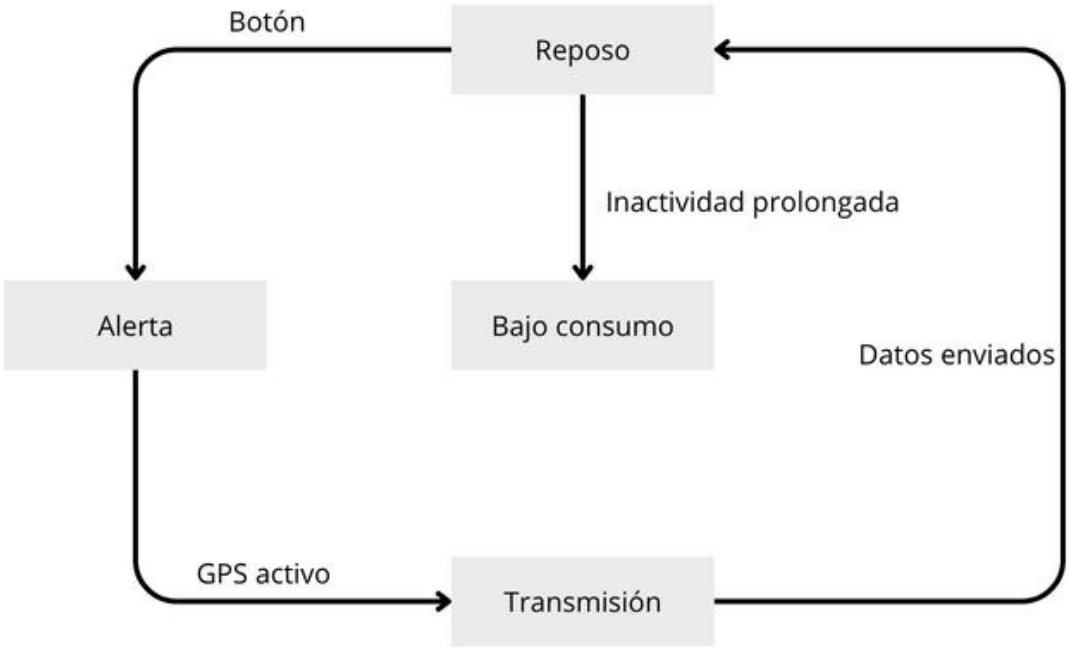
La información viaja a través de una red inteligente que selecciona automáticamente la mejor conexión disponible ya sea la red celular convencional, la red LoRa o un módulo XBee el cual transmite por medio de radiofrecuencias para zonas con poca cobertura, así garantiza que la

alerta llegue siempre a su destino, teniendo en cuenta esto al recibir la alerta el servidor central analiza la información y prioriza las situaciones dependiendo el orden de llegada.

Finalmente, la alerta llega al servidor central con el fin de poder visualizar la ubicación de la alerta e información relevante como los accesos más cercanos a la ubicación o las situaciones de robo en la zona, todo este proceso ocurre en menos tiempo del que toma leer esta descripción, creando un escudo de protección digital para las comunidades universitarias.

**Figura 5**

*Diagrama de estados del dispositivo portátil*



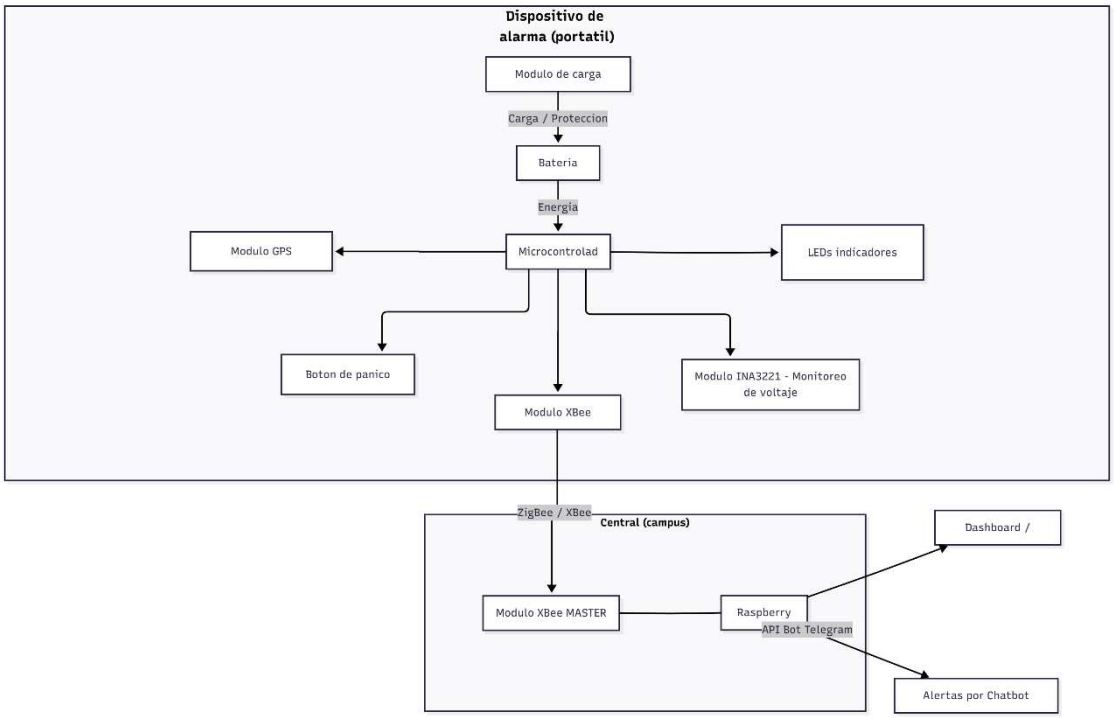

*Nota:* autoridad propia (2025)

En la figura 5 el diagrama presenta un flujo de funcionamiento que involucra diferentes estados y acciones, en primer lugar se inicia con un botón que genera una alerta, esta alerta puede llevar al dispositivo a un estado de reposo, en el cual se contemplan dos opciones: la inactividad prolongada o el modo de bajo consumo, después cuando el dispositivo se encuentra en reposo, puede continuar enviando datos si se activa el GPS, manteniendo así una conexión activa para la transmisión de información

Si el dispositivo experimenta una inactividad prolongada, también puede seguir funcionando en un modo de bajo consumo, garantizando que el consumo de energía no sea tan alto, esto permite que el sistema esté disponible para enviar datos sin consumir demasiada batería, lo que es esencial para la duración de la operación, es por esta razón que a través del diagrama se visualiza como el dispositivo optimiza su funcionabilidad y consumo energético en diferentes situaciones

**Figura 6**

*Diagrama de componentes*




*Nota:* autoridad propia (2025)

Teniendo en cuenta la anterior información se realizó un diagrama (figura 6) para poder ver la arquitectura del sistema en donde se visualiza de forma detallada el sistema de alarma portátil conectado a el servidor central, en donde la parte de arriba está dedicada al dispositivo portátil, cuyo núcleo es un microcontrolador que coordina la comunicación y el control de los sensores, al cual se conectan un módulo GPS para obtener la ubicación en tiempo real, un botón

de pánico que permite al usuario activarla alerta manualmente, además incorpora un módulo XBee en modo esclavo, encargado de la transmisión inalámbrica hacia la central.

En la parte derecha se encuentra el servidor central, compuesta por una Raspberry Pi y un módulo XBee configurado como maestro, el cual recibe las señales enviadas desde el dispositivo portátil en donde la Raspberry Pi procesa los datos, se conecta a un servidor o panel de control mediante protocolos HTTP permitiendo así la visualización, gestión y almacenamiento de la información de forma remota

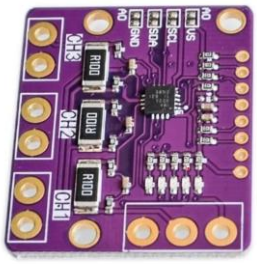
La comunicación entre el módulo XBee esclavo del dispositivo y el maestro del servidor central se realiza mediante tecnología ZigBee/XBee, asegurando un enlace inalámbrico de bajo consumo, después internamente la Raspberry Pi se comunica con su modulo maestro atreves de una conexión USB lo que garantiza compatibilidad y velocidad en la transferencia de datos

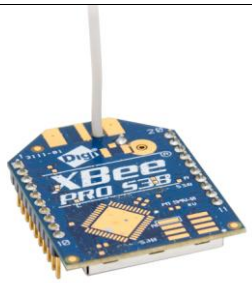
Según lo anterior esta arquitectura asegura que ante una emergencia la información viaja de forma rápida y confiable desde el dispositivo portátil hasta el servidor central y el envío de alertas a través del Chat Bot de Telegram

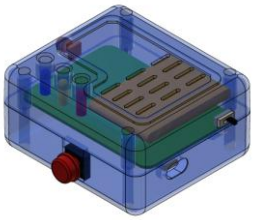
Con base en los esquemas se procedió a la elección de los componentes electrónicos que conforman el sistema de alarma, estos componentes fueron elegidos con base en su compatibilidad, eficiencia energética, disponibilidad en el mercado y bajo costo, para asegurar el funcionamiento del dispositivo y la integración efectiva entre el hardware y el software

**Tabla 2**

*Componentes*

| Componente | Imagen | Descripción |
|---|---|---|
| Microcontrolador ESP 32 C3 Super Mini |  | Es el núcleo principal del sistema ya que procesa la información de los diferentes módulos, además controla los Leds indicadores además tiene un tamaño relativamente pequeño sin perder todas las funciones de un esp 32 convencional |
| Botón de Pánico |  | Es un dispositivo físico permite al usuario generar la alerta |
| Modulo GPS Neo 8m |  | Modulo encargado de determinar la ubicación geográfica del dispositivo, brindando dos variables (Longitud y Latitud) |
| INA 3221 |  | Es un sensor de monitoreo de voltaje y corriente que permite supervisar el nivel de la batería |

| | | |
|---|---|---|
| Módulo de carga TP4056 |  | Modulo encargado de cargar la batería el cual incluye protección contra la sobre carga, asegurando la durabilidad de la batería |
| Módulo XBee PRO S3B |  | Módulos de comunicación inalámbrica basado en el protocolo ZigBee, utilizado para la transmisión y recepción, estos módulos operan a 900MHz |
| Resistencias de 220 Ω |  | Componentes utilizados para proteger los LEDs |
| Resistencias de 10 kΩ |  | Se emplean para el botón asegurando niveles lógicos estables |
| LED |  | Diodos emisores de luz usados para indicar el nivel de la batería, el funcionamiento del dispositivo y el nivel de la batería |
| Interruptor |  | Elemento de control manual para encender y apagar el dispositivo portátil |
| Cables Jumpers |  | Cables flexibles para garantizar la conexión de botón y el interruptor |

| | | |
|---|---|---|
| Placa PCB |  | Tarjeta de circuito impreso para realizar las conexiones eléctricas entre los distintos componentes |
| Impresión 3D (Carcasa) |  | Estructura fabricada con el fin de proteger los componentes electrónicos |
| Raspberry Pi |  | Servidor central del sistema encargado de recibir y procesar las alertas transmitidas mediante Node-RED |
| Micro SD |  | Tarjeta de memoria utilizada para el almacenamiento del sisma operativo de la Raspberry Pi |
| Batería Lipo |  | Fuente principal de alimentación portátil, la cual proporciona un voltaje de 3.7V y una capacidad de 1000 mAh suficiente para mantener la operación continua del dispositivo |

*Nota:* Autoría propia con imágenes de Mercado libre y Electroniclab (2025)

Con el fin de evaluar el consumo energético del sistema y determina la autonomía de funcionamiento del dispositivo portátil, se realizó una estimación de la corriente y potencia consumida por cada componente electrónico durante las diferentes fases de operación, con el fin de identificar los elementos de mayor demanda energética, optimizar el uso de la batería y garantizar una duración adecuada para las condiciones

En la tabla 3 se presentan los valores típicos de corriente y potencia de cada módulo los cuales fueron calculados teniendo en cuenta una alimentación de 3.3 V, así como una estimación de la autonomía del sistema alimentado con una batería Lipo de 1000 mAh, Estos datos fueron obtenidos a partir de las hojas de los fabricantes

**Tabla 3**

*Tabla del consumo energético del dispositivo portátil*

| **Componente** | **Estado/Modo** | **Corriente típica (mA)** | **Potencia (mW)** |
|---|---|---|---|
| ESP 32 C3 Super Mini | CPU activa (sin Wi-Fi) | 24 | 79 |
| XBee PRO S3B | Transmisión (Tx) | 215 | 710 |
| GPS NEO8m | Tracking continuo | 21 | 69 |
| INA3221 | Medición activa | 0.35 | 1.2 |
| LEDs | Siempre encendidos | 40 | 132 |
| Botón | Presionado/inactivo | 0.001 | 0.003 |
| TOTAL (envió activo) | - | ≈ 300.35 | ≈991 |
| TOTAL, reposo | - | ≈ 85.35 | ≈282 |

*Nota:* autoría propia tomando datos de los datasheets de cada componente tomados de ALLDATASHEET.ES (2025)

A partir de los valores presentados en la tabla, se puede observar que el módulo XBee PRO S3B es el componente con mayor consumo energético debido a si función de transmisión inalámbrica a través de radiofrecuencias en la banda de 900 MHz, aunque por otro lado el módulo GPS NEO-8M mostro su consumo estable y ligeramente inferior concentrado principalmente en el momento de adquisición de las coordenadas una vez establecida la comunicación satelital, el INA3221 y los LEDS registraron consumos mínimos cumpliendo con funciones únicamente de monitoreo y señalización.

**Tabla 4**

*Autonomía de la batería de 1000 mAh (3.3V)*

| Estado | Corriente (mA) | Autonomía Aprox. |
|---|---|---|
| Reposo | 85.35 | 11-12 horas |
| Transmisión Activa | 300.35 | 3.20 horas |

*Nota:* autoridad propia (2025)

En conjunto, los resultados demuestran que el sistema presento un consumo promedio garantizando una autonomía de 4 horas con una batería LiPo de 1000 mAh.

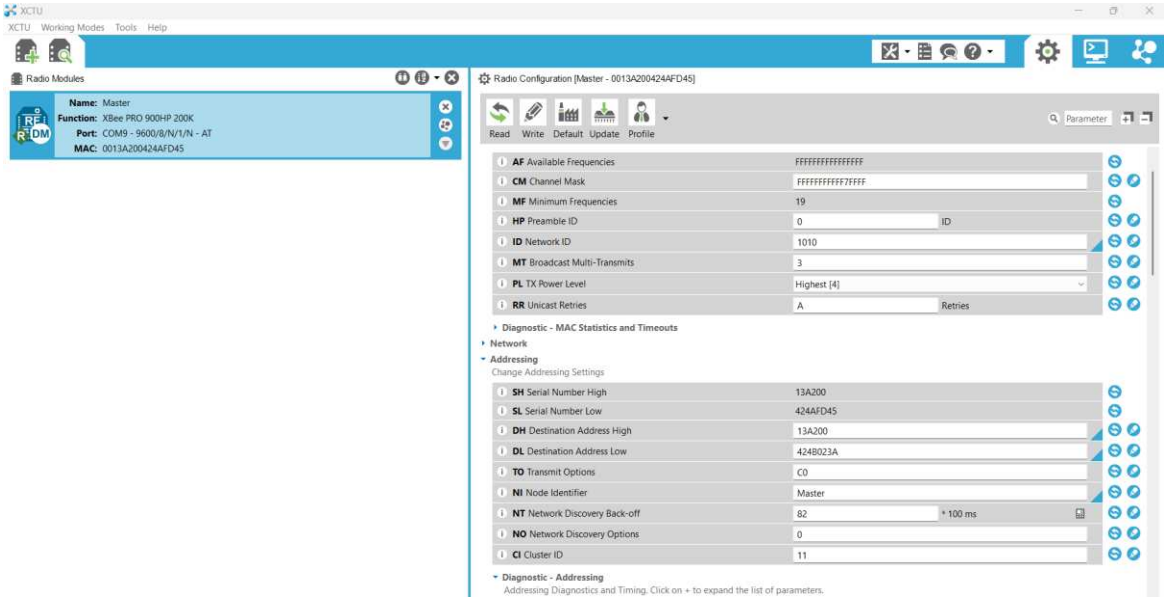
**Objetivo 2- programación del sistema e integración del sistema**

**Módulos XBee PRO S3B**

La siguiente fase fue programar cada componente, primero se configuraron y programaron los módulos XBee PRO S3B los cuales constituye el medio de comunicación inalámbrica ente el dispositivo portátil y el servidor central, para este proceso se utilizó la aplicación XCTU, desarrollada por Digi Internacional la cual permite configurar, probar y monitorear los módulos de forma gráfica y precisa.

**Figura 7**

*Programa XCTU*




*Nota:* autoría propia (2025)

Inicialmente como se muestra en la figura 7 se estableció la red definiendo un ID de red (1010) único, con el fin de garantizar que los módulos se comunicaran exclusivamente entre sí y evitar interferencias con otros dispositivos inalámbricos, posteriormente se designó uno como coordinador o maestro, conectado a la Raspberry Pi y el otro como dispositivo remoto o esclavo, integrando en el sistema portátil con el microcontrolador ESP32 C3 Super mini.

Durante la configuración se ajustaron parámetros como la velocidad de transmisión y las direcciones de 64 bits (SH y SL), necesarias para establecer una comunicación multipunto a punto, una vez completa la configuración, se realizaron pruebas de conectividad mediante el entorno XCTU, verificando el envío y recepción de paquetes de datos sin pérdida de información ni retardos significativos

Después de configurar los módulos XBee PRO S3B en la aplicación XCTU, se procedió a su programación y prueba de comunicación usando el entorno de Arduino IDE con el fin de comprobar la transmisión y recepción a través de un monitor serial

***Figura 8***

*Programa Arduino (Receptor)*

rx.ino

```
#include <HardwareSerial.h>

// Puerto serial para el XBee
HardwareSerial XBeeSerial(1);

// Pines donde conectarás el XBee al ESP32-C3
#define XBEE_TX 21   // ESP32 TX -> DIN (XBee pin 3)
#define XBEE_RX 20   // ESP32 RX -> DOUT (XBee pin 2)

void setup() {
  // Serial para depuración
  Serial.begin(9600);
  delay(1000);
  Serial.println("Receptor ESP32-C3 con XBee Pro S3B...");

  // Inicializamos el puerto serial para el XBee
  XBeeSerial.begin(9600, SERIAL_8N1, XBEE_RX, XBEE_TX);
  delay(1000);
}

void loop() {
  // Si hay datos desde el XBee, los mostramos en Serial
  while (XBeeSerial.available()) {
    char c = XBeeSerial.read();
    Serial.print(c);
  }
}
```

*Nota:* Anexo A *Fuente:* autoría propia (2025)

**Figura 9**

*Programa Arduino (Transmisor)*

Tx.ino

```ino
#include <HardwareSerial.h>

// Creamos un puerto serial para el XBee (usar los pines de tu ESP32-C3)
HardwareSerial XBeeSerial(1);

// Pines donde conectarás el XBee al ESP32-C3
#define XBEE_TX 21   // ESP32 TX -> DIN (XBee pin 3)
#define XBEE_RX 20   // ESP32 RX -> DOUT (XBee pin 2)

void setup() {
  // Serial para depuración en el monitor serie
  Serial.begin(9600);
  delay(1000);
  Serial.println("Iniciando ESP32-C3 con XBee...");

  // Inicializamos el puerto serial del XBee
  XBeeSerial.begin(9600, SERIAL_8N1, XBEE_RX, XBEE_TX);
  delay(1000);

  Serial.println("Listo. Enviando datos al XBee...");
}

void loop() {
  // Enviar un mensaje cada 2 segundos
  XBeeSerial.println("Hola desde ESP32-C3 con XBee Pro S3B!");
  Serial.println("Mensaje enviado...");

  delay(2000);

  // Si llega algo desde el XBee (otro módulo respondió), lo mostramos
  while (XBeeSerial.available()) {
    char c = XBeeSerial.read();
    Serial.print(c);
  }
}
```

*Nota:* Anexo B *Fuente*: autoría propia (2025)

Una vez configurados los módulos de comunicación, se continuo con la programación individual de los componentes electrónicos con el fin de garantizar el correcto funcionamiento de cada componente antes de la integración total del sistema, se comprobó la transmisión (figura 9) y la recepción (figura 8) como demuestra en el anexo A y B en donde para establecer la comunicación serial se empleó la librería HardwareSerial, creando un puerto UART adicional en el microcontrolador. Los pines 21(Tx) y 20(Rx) del ESP32 C3 Super mini se conectaron correctamente a los pines DIN (3) y DOUT (2) del XBee PRO S3B, definiendo un canal de transmisión directa mediante el protocolo SERIAL_8N1 a una velocidad de 9600 baudios, valor suficiente para el intercambio estable entre ambos módulos

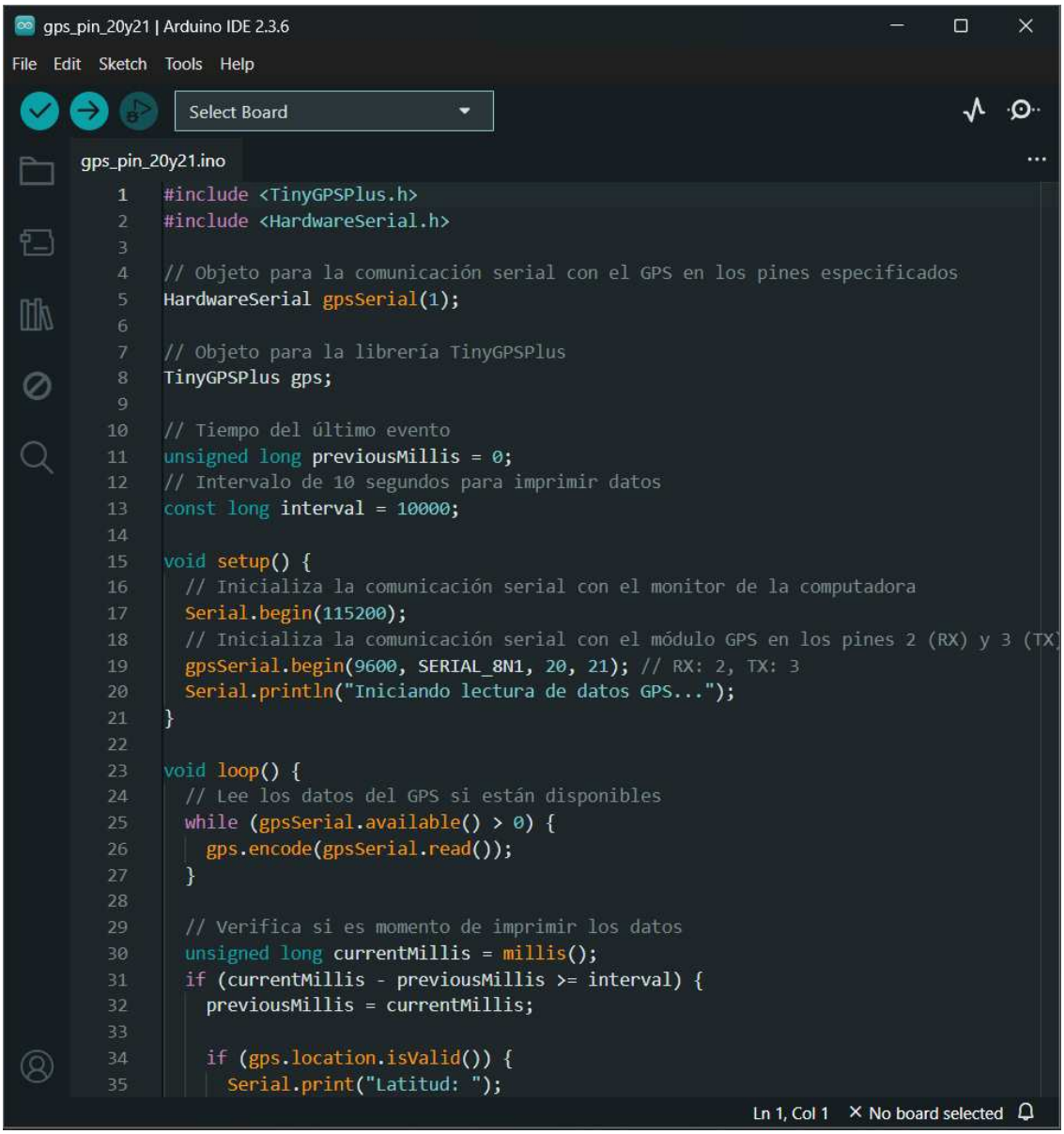
En el dispositivo transmisor (Tx), se programó un ciclo de envío periódico de mensajes a través del puerto serial, verificando la correcta salida de los datos hacia el XBee PRO S3B Master mediante el monitor serial y mostrar en la pantalla cualquier información recibida, permitiendo comprobar la comunicación de la red

Durante estas pruebas, los módulos demostraron una conexión estable, sin pérdida de paquetes ni interferencias, confirmando la correcta configuración de canal de comunicación y validez del ID de la red, en esta etapa cabe resaltar que para que el sistema de alarma pudiera enviar en tiempo real los mensajes, la salida de estos de estar en formato JSON para más adelante poder visualizarlos en el servidor de red.

**GPS NEO-8M**

**Figura 10**

*Programación del GPS*




*Nota:* Anexo C *Fuente:* autoría propia (2025)

En segundo lugar se realizó la programación del módulo GPS NEO-8M con el microcontrolador ESP 32 C3 Super Mini (Anexo C), estableciendo la comunicación por el protocolo UART para la adquisición y lectura de las coordenadas en tiempo real, al principio se

realizó la programación utilizando los pines de recepción y transmisión (20 Y 21) al confirmar el correcto funcionamiento del GPS pero como los módulos XBee PRO S3B necesitan esos pines a través de código se le asigno ESP32 dos pines diferentes como unos segundos TX y RX con el fin que el GPS y los modulo XBee tengan su propio TX y RX para la transmisión y recepción de la información.

**Botón y LEDs**

Posteriormente, se procedió a la programación e integración del botón de pánico y los LEDs indicadores con el microcontrolador ESP32 C3 Super mini gestionando las funciones de activación y retroalimentación visual del sistema.

En el programa del botón (Anexo D) se declaró en el pin GPIO 3 y se definieron 2 variables lógicas (alarma_on y alarma_off) encargadas de registrar el estado del sistema, dentro del ciclo principal el microcontrolador realizo la lectura del pin digital para que cuando se detecte un estado bajo la alarma se activa la variable alarma_on =1 y alarma_off =0, indicando que el botón ha sido presionado de lo contrario se mantiene alarma_on =0 y alarma_off =1, reflejando que el sistema permanece en reposo.

**Figura 11**

*Programación del Botón*

boton.ino

```ino
#include <Arduino.h>

// Pin del botón
const int botonPin = 3;

// Variables
int alarma_off = 0;
int alarma_on = 0;

void setup() {
  Serial.begin(115200);
  pinMode(botonPin, INPUT_PULLUP);  // Botón con resistencia pull-up interna
}

void loop() {
  // Leer el estado del botón
  int estadoBoton = digitalRead(botonPin);

  // Resetear valores
  alarma_off = 0;
  alarma_on = 0;

  // Si está presionado (LOW por pull-up), se activa alarma_on
  if (estadoBoton == LOW) {
    alarma_on = 1;
  } else {
    alarma_off = 1;
  }

  // Crear JSON manualmente
  String jsonString = "{";
  jsonString += "\"alarma_off\":" + String(alarma_off) + ",";
  jsonString += "\"alarma_on\":" + String(alarma_on);
  jsonString += "}";
```

*Nota:* Anexo D *Fuente:* autoría propia (2025)

Para la programación de los LEDs se definieron tres salidas digitales correspondientes a los distintos estados del sistema (LED de encendido, LED de alarma, LED de la batería)

**Figura 12**

Programación de los LEDs

led_pulsador.ino

```
#include <TinyGPSPlus.h>
#include <SoftwareSerial.h>
#include <Arduino.h>

// ----------------- GPS -----------------
SoftwareSerial gpsSerial(4, 5); // RX, TX
TinyGPSPlus gps;

unsigned long previousMillis = 0;
const long interval = 1000; // 1 segundo (puedes cambiar a 10000 ms = 10s)

// ----------------- Botón y LEDs -----------------
const int botonPin = 3;
const int led_encendido = 6;
const int led_alarma = 7;
const int led_bateria = 8;

int alarma_off = 0;
int alarma_on = 0;

void setup() {
  Serial.begin(115200);   // Monitor serial
  gpsSerial.begin(9600);  // GPS NEO-8M

  // Configuración de pines
  pinMode(botonPin, INPUT_PULLUP);
  pinMode(led_encendido, OUTPUT);
  pinMode(led_alarma, OUTPUT);
  pinMode(led_bateria, OUTPUT);

  // LED encendido siempre al iniciar
  digitalWrite(led_encendido, HIGH);
```

*Nota:* Anexo E *Fuente:* autoría propia (2025)

Como se muestra en el anexo E, el funcionamiento se basa en la lectura continua del estado del botón mediante la función digitalRead () ya que si el botón detecta un nivel bajo (LOW) el sistema activa el LED de la alarma manteniendo encendido los demás indicadores, además el código fue optimizado con el fin de evitar interferencias eléctricas y rebotes del botón

permitiendo que cada pulsación sea reconocida de forma precisa y así asegurar una respuesta visual clara y rápida.

### INA 3221

**Figura 13**

*Programación del INA3221*

configuracion_INA3221.ino

```
#include <Wire.h>

// Pines I2C del ESP32-C3
#define SDA_PIN 8
#define SCL_PIN 9

// Dirección detectada
#define INA_ADDR 0x41

// Registros
#define REG_SHUNT_CH1 0x01
#define REG_BUS_CH1   0x02

// Rango batería (0% = 2.0 V, 100% = 3.7 V)
const float V_MIN = 2.0f;
const float V_MAX = 3.7f;

void setup() {
  Serial.begin(115200);
  delay(300);
  Serial.println("\n🔍 Lectura INA3221 corregida (CH1)");

  Wire.begin(SDA_PIN, SCL_PIN);
  Wire.setClock(100000);
  delay(100);
}

bool read16_from(uint8_t addr, uint8_t reg, uint16_t &out) {
  Wire.beginTransmission(addr);
  Wire.write(reg);
  if (Wire.endTransmission(false) != 0) return false;
  Wire.requestFrom((int)addr, (int)2);
```

*Nota:* Anexo F. *Fuente:* autoría propia (2025)

El siguiente que se programo fue el INA3221, para poder monitorear los niveles de voltaje y corriente de la batería que alimenta el dispositivo portátil como se muestra en el anexo F, el código implementado establece la dirección del sensor en el bus I²C y configura el primer canal para la adquisición periódica de los valores de voltaje del bus a partir de estos datos, el microcontrolador calcula el voltaje total de la batería y genera una estimación de porcentaje total de la batería

**Integración de los códigos**

En la última fase de la programación de los componentes, se unifico un solo programa para todas las funciones individuales previamente configuradas: el módulo GPS NEO-8M, el botón de pánico, los LEDs indicadores, el sensor de monitoreo INA 3221 y los módulos de comunicación inalámbrica XBee PRO S3B, esta unificación permitió establecer la interacción coordinada entre los distintos subsistemas logrando una operación estable, sincronizada y eficiente dentro del dispositivo portátil

Como se muestra en el anexo F el código fue implementado en el entorno Arduino IDE, utilizando librerías como TinyGPSPlus para el manejo de datos de geolocalización, la librería Wire para la comunicación I²C con el INA3221, la comunicación con los módulos XBee PRO S3B se configuro mediante HardwareSerial, mientras que el GPS opero a través del SoftwareSerial, reasignando pines alternativos de transmisión y recepción para evitar interferencias entre ambos módulos.

**Figura 14**

*Código Final*

esp_gps_ina_boton_leds.ino

```
#include <HardwareSerial.h>
#include <TinyGPSPlus.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <Arduino.h>

// ----------------- XBee -----------------
HardwareSerial XBeeSerial(1);
#define XBEE_TX 21   // ESP32 TX -> DIN (XBee pin 3)
#define XBEE_RX 20   // ESP32 RX -> DOUT (XBee pin 2)

// ----------------- GPS -----------------
SoftwareSerial gpsSerial(4, 5); // RX=4, TX=5
TinyGPSPlus gps;

// ----------------- Botón y LEDs -----------------
const int botonPin = 3;
const int led_encendido = 0;   // cambiado
const int led_alarma = 1;      // cambiado
const int led_bateria = 2;     // cambiado

int alarma_off = 0;
int alarma_on = 0;

// Control del botón
bool botonAnterior = HIGH;
bool alarmaActiva = false;
unsigned long tiempoAlarma = 0;

// ----------------- INA3221 -----------------
#define SDA_PIN 8
#define SCL_PIN 9
```

*Nota:* Anexo G. *Fuente:* autoría propia (2025)

Durante el funcionamiento, el microcontrolador ESP32 C3 Super mini ejecuta un ciclo continuo de adquisición, procesamiento y transmisión de la información en donde en primer lugar se verifica el estado del botón  es decir si este es presionado cambia el estado de las

variables lógicas alarma_on y alarma_off  y el LED de alarma, iniciando así el envío de mensajes de alerta teniendo en cuenta que paralelamente, el sistema obtiene las coordenadas de GPS y las incorpora al mensaje de salida junto al estado de los LEDs y el nivel de la batería el cual es dado por el sensor INA3221ya que este realiza lecturas periódicas del voltaje de la batería a través del bus I²C calculando así el porcentaje de carga de la batería advirtiendo al usuario sobre el nivel crítico de energía.

Toda la información recolectada se estructura en formato JSON y se envía simultáneamente por los puertos Serial y XBeeSerial hacia el servidor central (Raspberry Pi), ya que este formato se logra una transmisión ordenada, legible y compatible con los flujos de procesamiento de Node-RED, donde se visualiza y gestiona la alerta.

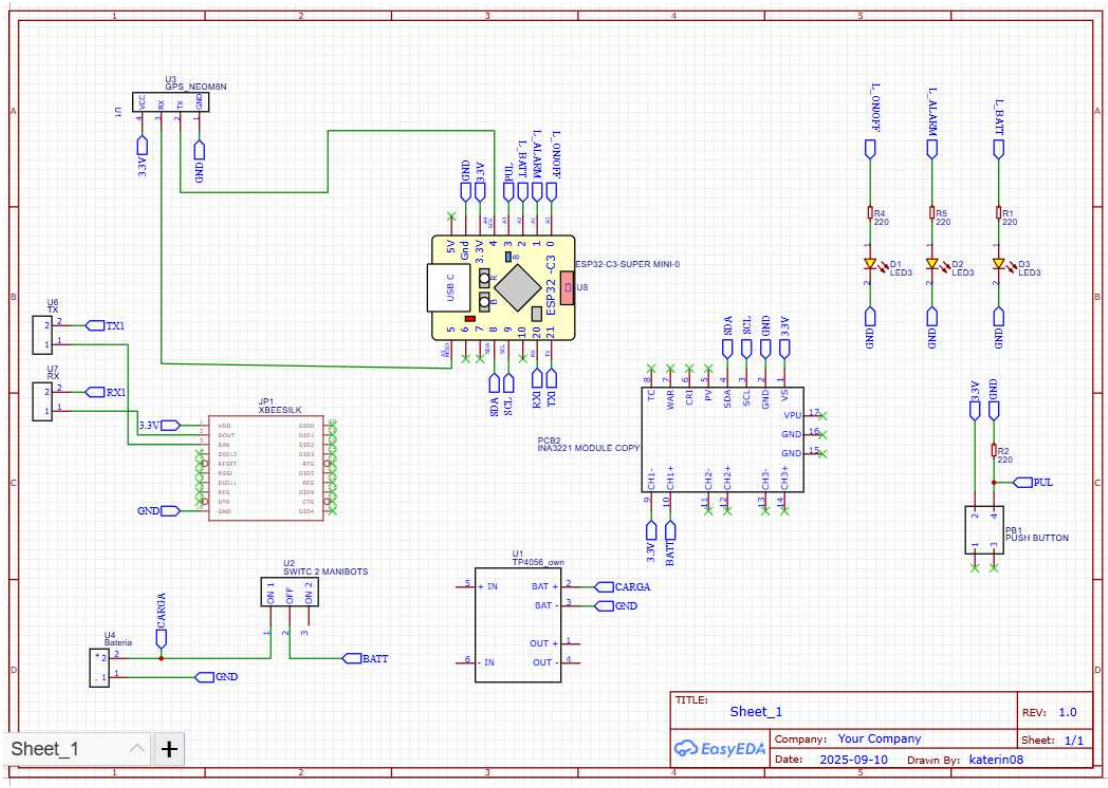
La integración final del código consolido el funcionamiento global del sistema de alarma, logrando una comunicación fluida entre el Hardware y Software, en donde se tuvo en cuenta el consumo energético, se redujo el tiempo de respuesta y logro la transmisión de información formato JSON.

**Ensamblaje del dispositivo final**

Una vez finalizado el diseño electrónico y la programación de los diferentes módulos, se procedió con el ensamblaje físico del sistema sobre una placa PCB personalizada, la cual fue desarrollada con el fin de integrar de manera ordenada y funcional todos los componentes del prototipo

**Figura 15**

*Diseño de la PCB*

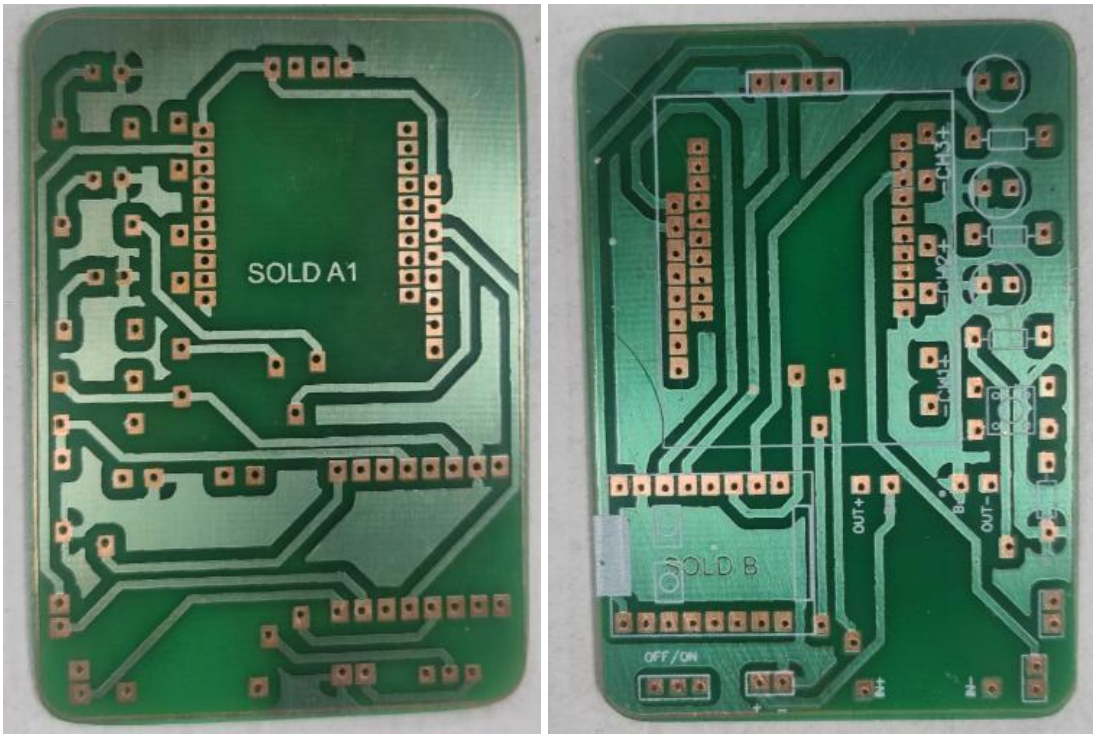



*Nota.* autoría propia (2025)

El proceso inicio con el diseño del circuito impreso en la plataforma EasyEDA (figura 15), en donde de trazaron las rutas de conexión, el posicionamiento de los componentes y las dimensiones generales de la placa, ya que este entorno permitió realizar la simulación eléctrica y la verificación de continuidad de pistas, asegurando que las conexiones entre los módulos fueran correctas antes de la fabricación, posteriormente el diseño fue exportado en formato Gerber y enviado a fabricación, obteniendo así la placa impresa con las pistas de cobre y la serigrafia correspondiente como se ve en la figura 16

**Figura 16**

*Impresión de la placa*




*Nota.* autoría propia (2025)

Para el montaje de los componentes, se emplearon diferentes métodos de conexión según el tipo de modulo y su necesidad de reemplazo o mantenimiento:

- Los módulos ESP32 C3 Super mini, GPS NEO-8N, XBee PRO S3B fueron instalados mediante regletas hembra, lo que permitió insertarlos sin necesidad de soldadura directa. Este método facilita su retiro o sustitución en caso de fallas, además mejora la accesibilidad durante las pruebas.
- Los módulos INA3221 y TP4056 fueron soldados directamente sobre la placa, garantizando una conexión eléctrica estable y reduciendo la resistencia de contacto en líneas críticas de alimentación y medición

- El interruptor, el botón de pánico y la batería LiPo fueron conectados mediante cables ya que estos componentes requieren cierta ubicación especifica fuera del cuerpo principal de la placa, los cables proporcionan la flexibilidad necesaria para adaptarse a la carcasa impresa en 3D y facilitan el mantenimiento del sistema sin comprometer la integridad de las pistas

Durante el ensamblaje se emplearon técnicas de soldadura manual con un cautín de punta fina, utilizando estaño para asegurar que las uniones sean confiables además se realizó una inspección visual de las soldaduras y se midió continuidad en las pistas para poder verificar la conexión entre los terminales de cada componente

Finalmente, para poder garantizar la integridad estructural y estética del sistema, se diseñó una carcasa personalizada en 3D, el plano técnico fue desarrollado en un software de diseño asistido por computadora el cual se llevó a cabo considerando criterios de ergonomía, ventilación, distribución de componentes y facilidad de ensamblaje, de modo que la carcasa no solo cumpliera con su función de resguardo, si no también se tuvo en cuenta su funcionalidad y mantenimiento

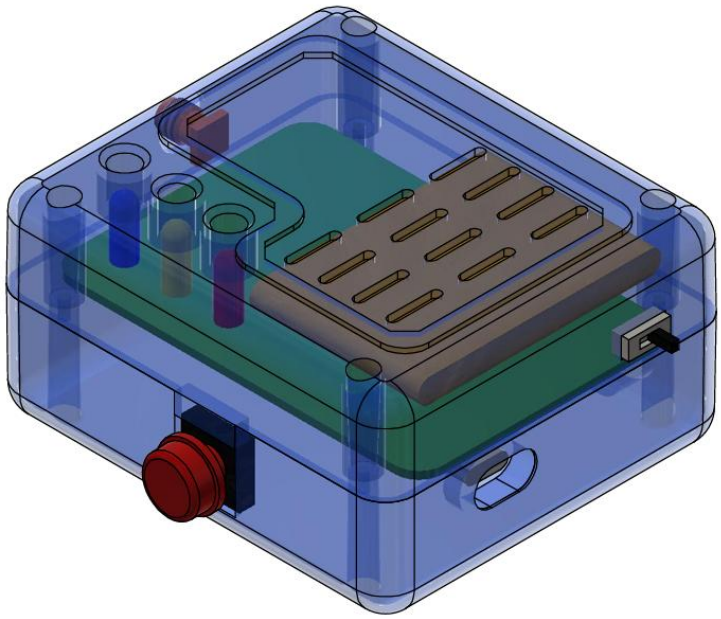
El diseño se realizó en tres etapas:

- Diseño de la base: en el anexo H se visualiza los puntos de anclajes y guías de posicionamiento que facilitan su montaje, además, se añadieron perforaciones para los tornillos de fijación, teniendo en cuenta la salida de cables del interruptor, el botón de pánico y la batería LiPo, su geometría interna permite mantener la placa elevada, evitando contacto directo con superficies que puedan provocar cortocircuitos

- Diseño de la tapa: en el anexo I se complementa la estructura de la base, incluyendo orificios para la visualización de los LEDs indicadores, el acceso al botón de pánico y la correcta ventilación de los módulos internos, también se consideró un espacio destinado a la antena del módulo XBee PRO S3B, con el fin de minimizar la interferencia
- Integración y ensamblaje: en el anexo J se muestra que el proceso de ensamblaje se contempló el acople preciso entre la base y la tapa, garantizando un cierre firme y estable mediante tornillos.

**Figura 17**

*Diseño de la carcasa*



*Nota.* elaborado por Olaechea, Omar (2025)

La impresión final se realizó usando filamento PLA tipo seda debido a su ligereza, rigidez lo que contribuye a que el dispositivo sea fácil de portar además el diseño permite un desmontaje rápido para labores de mantenimiento o actualización de hardware

**Figura 18**

*Carcasa*

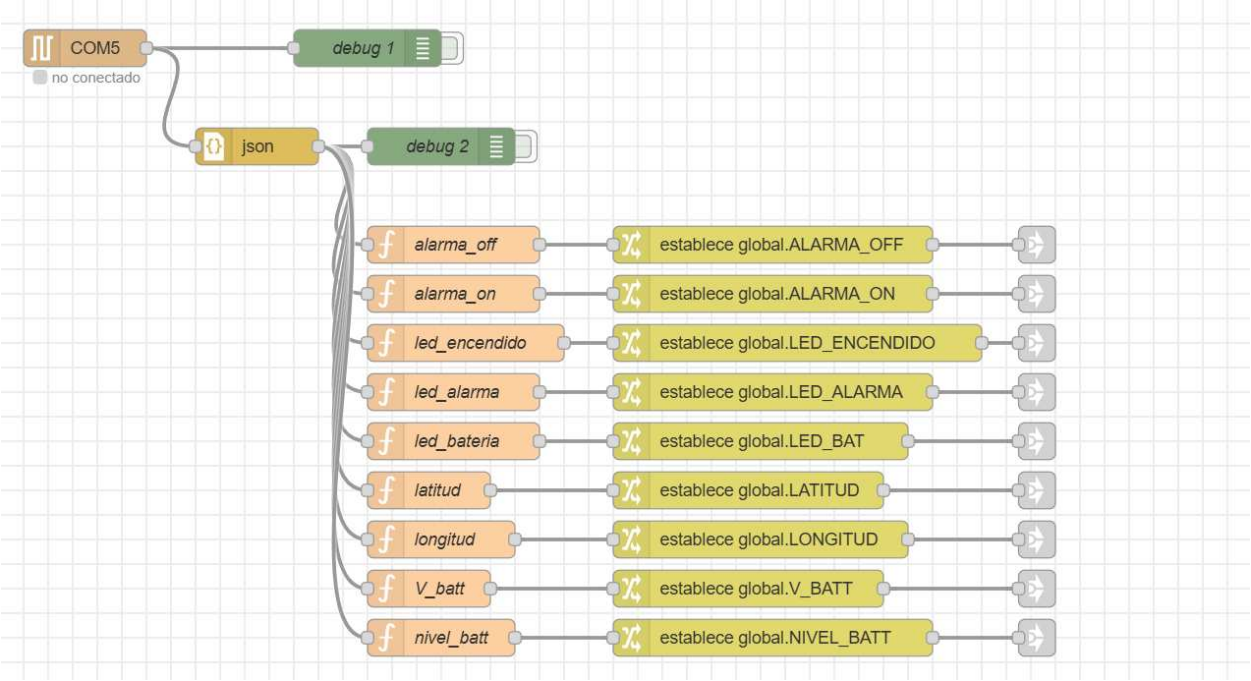
*Nota:* elaborado por Olaechea, Omar (2025)

### Configuración y operación del servidor central con Node-RED

El servidor central fue configurado utilizando Node -RED como plataforma de integración, procesamiento y visualización de los datos recibidos desde el dispositivo portátil de alarma, la herramienta se basa en flujos visuales, actuando como el núcleo de la arquitectura cliente-servidor, recibiendo la información enviada por el microcontrolador ESP32 C3 Super mini mediante el módulo XBee PRO S3B, para luego almacenarlo, presentarla en una interfaz gráfica accesible a través del dashboard.

**Figura 19**

*Flow 1*




*Nota:* autoridad propia tomado de Node-RED (2025)

El primer flujo denominado Flow 1 (figura 19) , se encarga de capturar, decodificar y distribuir la información enviada por el dispositivo, en donde inicia con un nodo Serial In, este nodo está configurado con el puerto en el que se encuentre el módulo XBee PRO S3B Master a una velocidad de 9600 bps, que recibe la trama JSON generada por el microcontrolador, su salida se dirige al nodo debug el cual es opcional ya que solo es para visualizar la información que llega por medio del puerto y así poder verificar los datos y la conexión en tiempo real

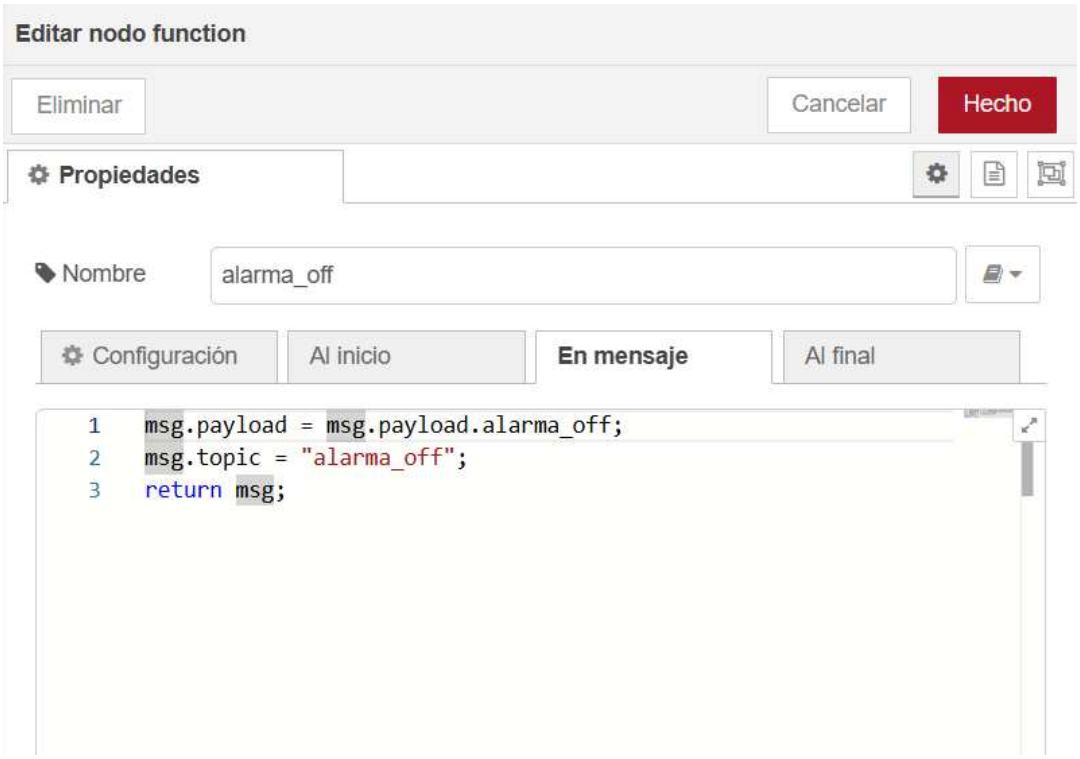
Posteriormente, la trama pasa a través de un nodo JSON el cual convierte la cadena de texto en un objeto estructurado “msg.payload”, permitiendo acceder individualmente a cada campo de información, después el flujo distribuye el objeto hacia múltiples nodos Fuction para extraer cada parámetro.

- Alarma_on y alarma_off: estado del botón

- Led_encendido, led alarma y led batería: indicadores visuales del sistema
- Latitud y longitud: coordenadas obtenidas del GPS
- Voltaje1_bateria y nivel_batería: parámetros medidos por el INA3221

**Figura 20**

*Instrucción del nodo Fuction*

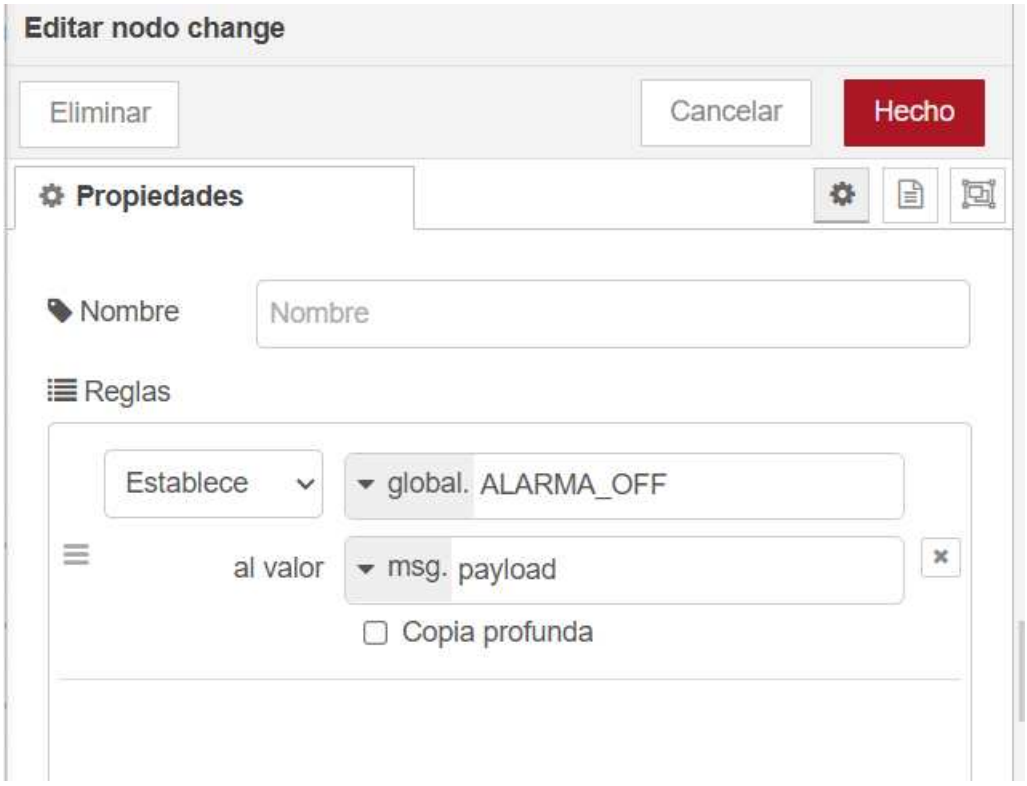



*Nota:* Autoría propia tomado de Node-RED (2025)

En la figura 20 se ve como el nodo Fuction procesa individualmente la información de la variable y la envía a un nodo Change que almacena el valor de la variable global dentro del contexto Node-RED como se ve en la figura 13, permitiendo que otros flujos o pestañas accedan a ellas

**Figura 21**

*Nodo Change*




*Nota:* Autoría propia tomado de Node-RED (2025)

Tras su almacenamiento, cada flujo se conecta a un nodo Link Out, en la cual distribuye las variables hacia otros flujos relacionados, como la interfaz del Dashboard

Como resultado, el Flow 1 toma la trama serial enviada por el microcontrolador, la interpreta, extrae los valores relevantes, los guarda como variables globales y los pone a disposición del resto del sistema para la visualización de la alarma.

**Figura 22**

*INICIO*


*Nota:* autoridad propia (2025)

La pestaña de inicio como se muestra em la figura 22 se constituye la interfaz principal del dashboard, en donde se presenta tanto la información general del proyecto como los modelos tridimensionales del dispositivo mostrando la estructura del dispositivo por medio de diferentes nodos ui_template que contiene código en HTML y CSS, en el primer nodo está configurado para que muestre el nombre del sistema, la universidad y el año en el que se realizó el proyecto, de igual manera el segundo nodo muestra la instrucción y la composición del dispositivo portátil con el fin de mostrarle al público como se creó, en el tercer y último nodo se configuro para que muestre el modelo en 3d del dispositivo creando una escena interactiva que permite rotar, escalar y visualizar la estructura del dispositivo.

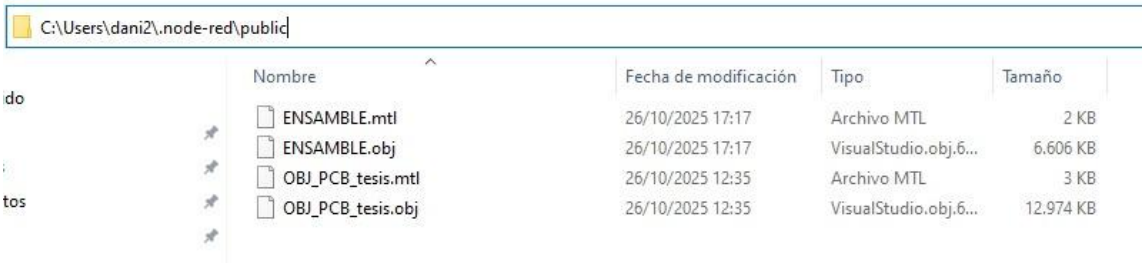
Con el fin de permitir la carga y visualización de los modelos tridimensionales generados, fue necesario habilitar un servidor de archivos estáticos dentro de Node-RED, este proceso permite que los archivos con extensión .obj y. mtl sea accesibles directamente desde un navegador web o desde nodos que requieran su lectura, siguiendo los siguientes pasos

- **Creación de la carpeta publica**, en primer lugar, se creó una carpeta publica denominada Public dentro del directorio principal de Node-RED, dentro de esta

carpeta se almacenaron los modelos 3D como muestra la figura 23, esta carpeta actúa como el directorio estático, es decir, el espacio donde Node-RED servirá archivos al navegador o a otros servicios conectados

**Figura 23**

*Carpeta Public*




*Nota:* autoridad propia (2025)

- **Modificación del archivo de configuración settings.js**: en este paso es necesario acceder al archivo de configuración de Node-RED denominando settings.js, localizado en el mismo directorio de Node-RED, este archivo fue abierto mediante un block de notas y se buscó la línea correspondiente al parámetro “httpStatic”, el cual se encuentra comentado por defecto “//httpStatic: '/home/nol/node-red-static/' “ dicha línea es eliminada y remplazada para poder apuntar a la ruta completa de la carpeta public creada anteriormente en donde en el caso del sistema operativo Windows (figura 24), la configuración quedaría de la siguiente manera, sin embargo al trabajar con otro sistema operativo como Linux en una Raspberry Pi las rutas cambian a httpStatic: "/home/pi/.node-red/public",

**Figura 24**

*Configuracion en la carpeta* settings.js

```
/** When httpAdminRoot is used to move the UI to a different root path, the
 * following property can be used to identify a directory of static content
 * that should be served at http://localhost:1880/.
 * When httpStaticRoot is set differently to httpAdminRoot, there is no need
 * to move httpAdminRoot
 */
httpStatic: 'C:/Users/dani2/.node-red/public', //single static source
/**
 *  OR multiple static sources can be created using an array of objects...
 *  Each object can also contain an options object for further configuration.
 *  See https://expressjs.com/en/api.html#express.static for available options
 *  They can also contain an option `cors` object to set specific Cross-Origin
 *  Resource Sharing rules for the source. `httpStaticCors` can be used to
 *  set a default cors policy across all static routes.
```

*Nota:* autoridad propia (2025)

- **Verificación del directorio estático en Node-RED:** Una vez editado y guardado el archivo settings.js, se reinició Node-RED mediante el comando “node-red start” de esa forma durante el arranque se podrá visualizar el texto mostrado en la figura 25 y eso confirmo que los cambios se guardaron correctamente y que Node-RED esta tiene acceso al contenido desde la carpeta configurada.

**Figura 25**

*Lineas de comando en el arranque del Node-RED*

```
26 Oct 13:37:45 - [info] Settings file  : C:\Users\dani2\.node-red\settings.js
26 Oct 13:37:45 - [info] HTTP Static    : C:\Users\dani2\.node-red\public > /
```

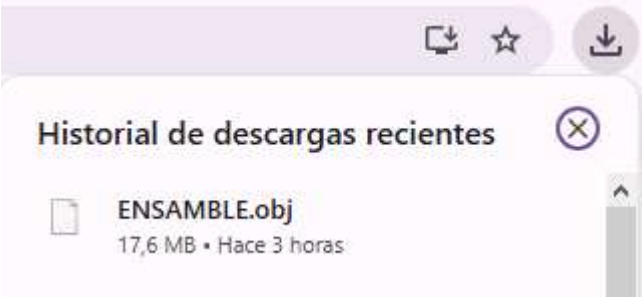
*Nota:* autoridad propia (2025)

- **Prueba de acceso a los archivos:** para comprobar que los archivos se encuentran disponibles, se accedió desde el navegador web utilizando la direccion local del servidor del Node-RED (http://localhost:1880/ENSAMBLE.obj) y si el archivo se descarga o se visualiza correctamente como en la figura 26, significa que la configuración del servidor estatico es funcional y además Node-RED esta

reconociendo los archivos ubicados en la carpeta public, de esta forma cualquier recurso colocado dentro de dicho directorio puede ser utilizado o llamado por los flujos del sistema, lo que resulta especialmente útil para la carga de modelos 3D, imágenes, hojas de estilo o scripts complementarios

***Figura 26***

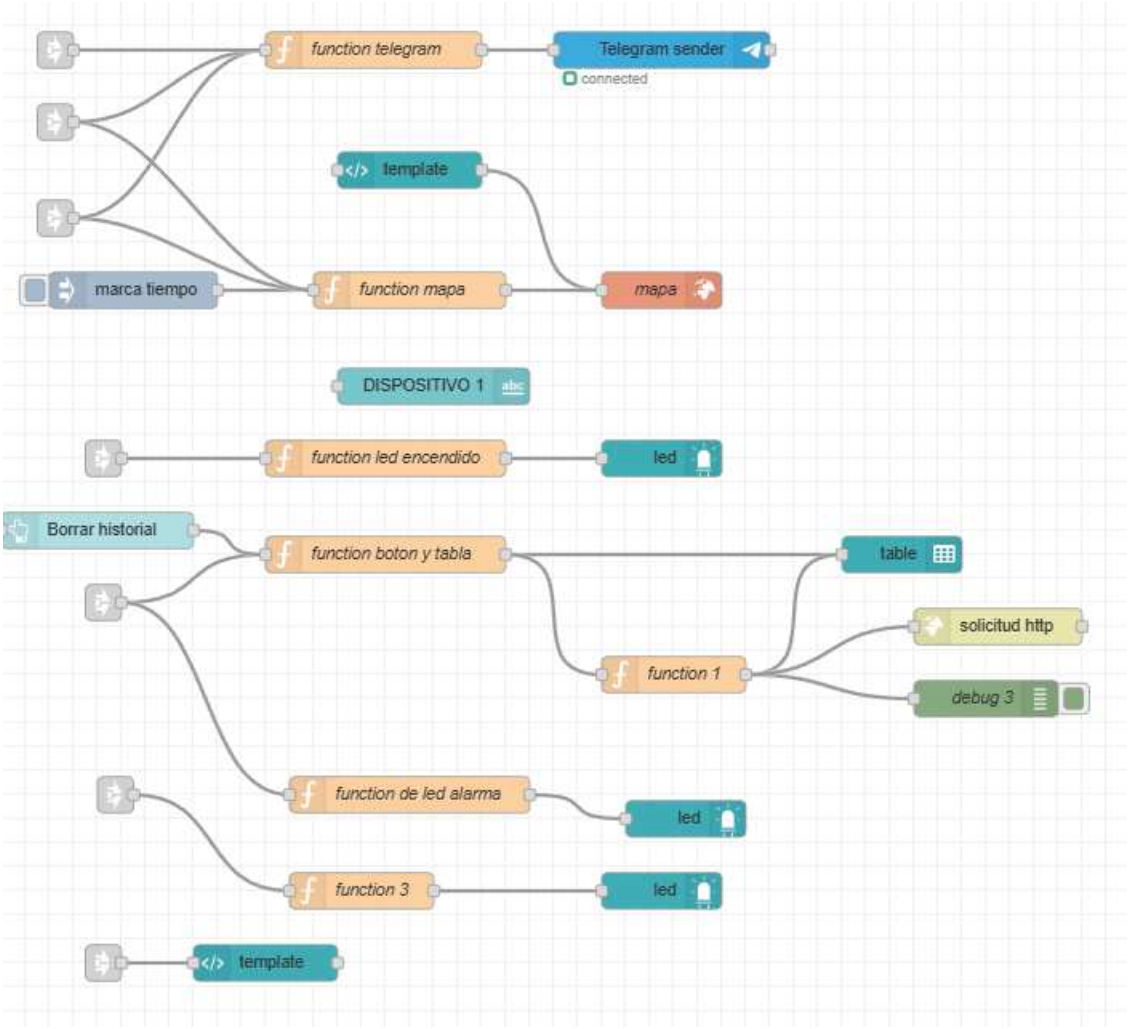
*Prueba de acceso a los archivos de la carpeta public*




*Nota:* autoridad propia (2025)

**Figura 27**

*GEOLOCALIZACION*




*Nota:* autoridad propia (2025)

Para finalizar en la pestaña de geolocalización se visualiza el sistema de alarma como tal ya que cuenta con un mapa interactivo en el cual se puede visualizar la ubicación del dispositivo en tiempo real, gracias al Flow 1 se hace el llamado de las variables hacia la pestaña de geolocalización mediante los nodos link out, estos se enlazan con los nodos link in presentes en la pestaña, teniendo en cuenta eso una vez recibidas las coordenadas, un nodo de tipo fuction

denominado mapa realiza el procesamiento de los datos (figura 28) para construir la estructura que el componente que el nodo worldmap requiere

**Figura 28**

*Fuction Mapa*

```
let lat= global.get("LATITUD");
let lon = global.get("LONGITUD");
let fecha = new Date().toLocaleString();


msg.payload = {
    name: "GPS",
    lat:lat,
    lon:lon,
    icon: "https://cdn-icons-png.flaticon.com/512/252/252025.png",
    iconSize: [60, 60],
    iconColor: "red",
    layer: "GPS",
    label: "<div style='width:200px; white-space:normal; text-align:left;
        + "📍 <b>Coordenadas de dispositivo 1:</b><br>"
        + "Latitud: " + lat + "<br>"
        + "Longitud: " + lon + "<br>"
        + "🕒 Fecha: " + fecha
        + "</div>"
};
return msg;
```

*Nota:* autoridad propia (2025)

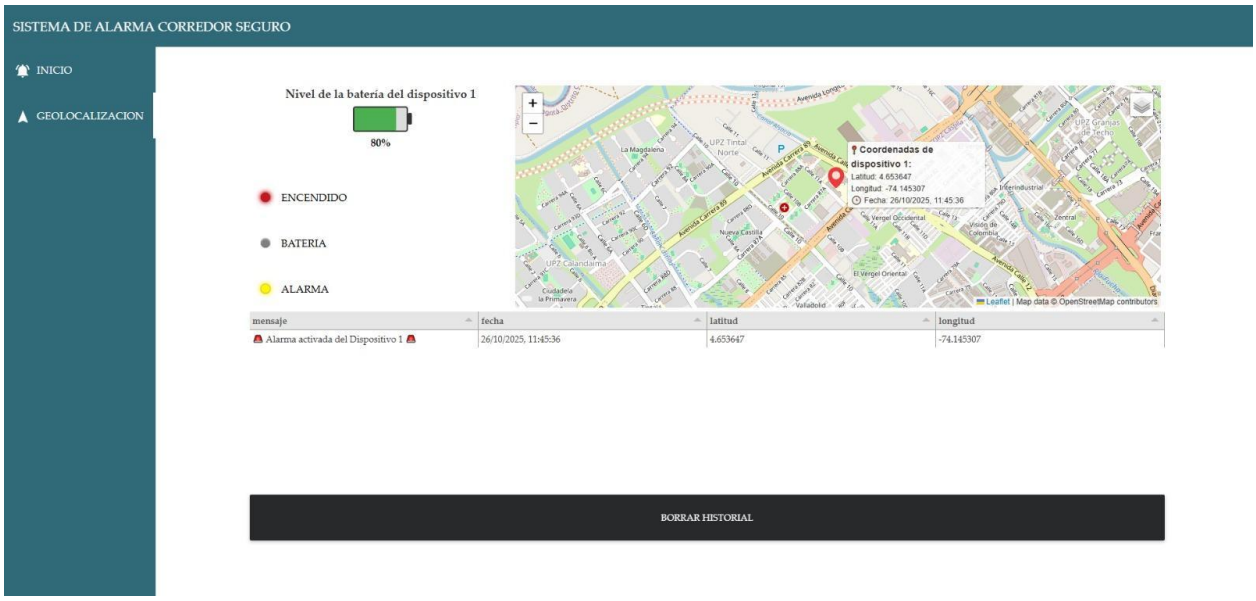
En el nodo WorldMap se recuperan las coordenadas globales (global.get("LATITUD") y global.get("LONGITUD") y se encapsulan en un objeto msg.payload con los campos lat y lon, adicionalmente se agregan propiedades visuales como el nombre del dispositivo , el icono representativo, la etiqueta que muestra la información de la ubicación y el estado dela conexión en tiempo real como se ve en la figura 28

Posteriormente, este mensaje se envía al nodo worldmap el cual se encuentra configurado con la ruta /worldmap y se encarga de renderizar el marcador sobre el mapa interactivo para esto

se una se usa un nodo iu_templete que se conecta con el mapa por medio de un elemento <iframe src="/worldmap"> con el fin de poder observar la posición actualizada del dispositivo directamente desde la pestaña de geolocalización del sistema como se ve en la figura 29.

**Figura 29**

*Pestaña GEOLOCALIZACION*



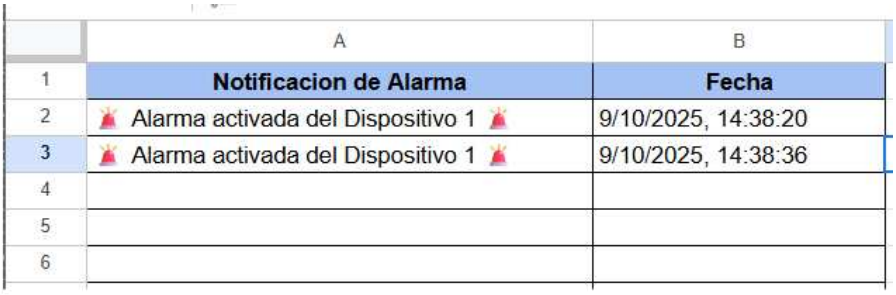

*Nota:* autoridad propia (2025)

El sistema también cuenta con un nodo inject que permite realizar pruebas o actualizaciones manuales del mapa, garantizando la correcta sincronización de los datos, gracias a este diseño cada vez que el microcontrolador envía una nueva trama de datos, las coordenadas se actualizan automáticamente en el dashboard

Además de la visualización, esta pestaña incorpora un nodo adicional de tipo HTTP que permite enviar los datos capturados hacia una hoja de cálculo en línea, este nodo está configurado para ejecutar una petición POST hacia un endpoint público, enviando como contenido un objetivo JSON de esta forma cada actualización del flujo genera una nueva fila en

el documento lo que facilita el registro cronológico de las alertas y la trazabilidad de los eventos como se ve en la figura 30.

**Figura 30**



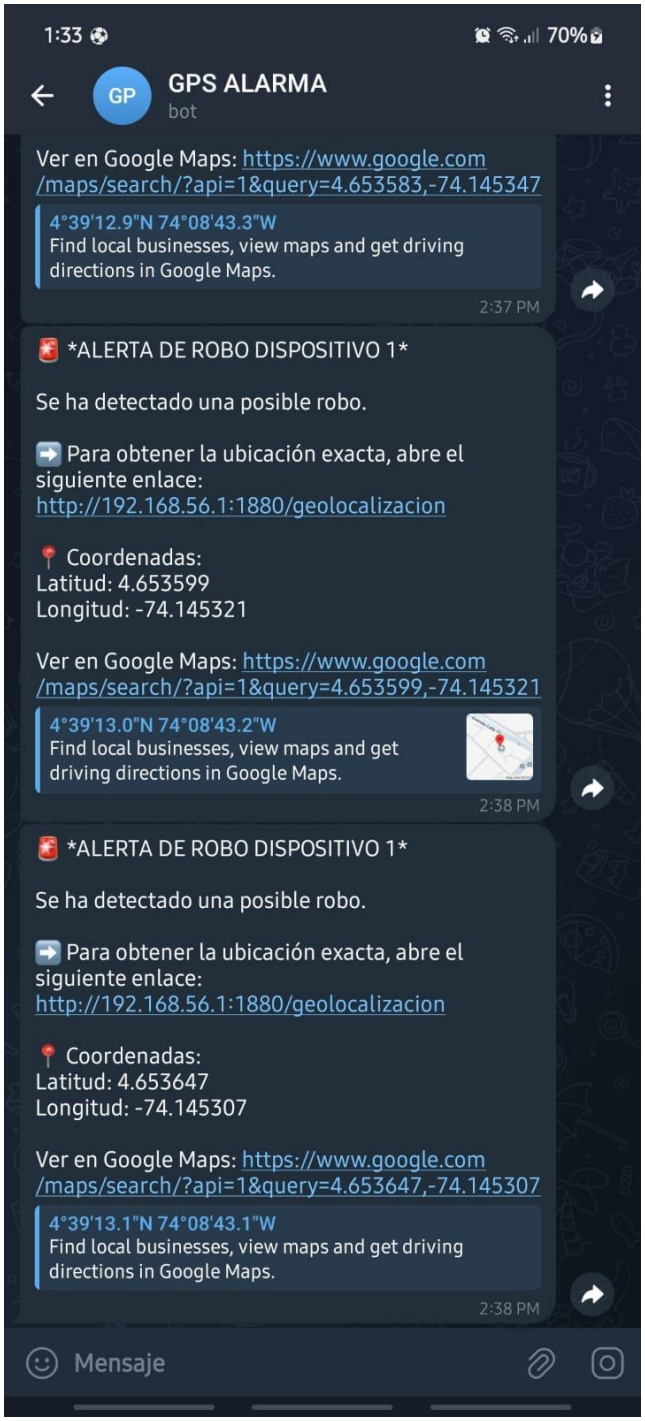
| | A | B |
|---|---|---|
| 1 | **Notificacion de Alarma** | **Fecha** |
| 2 | 🚨 Alarma activada del Dispositivo 1 🚨 | 9/10/2025, 14:38:20 |
| 3 | 🚨 Alarma activada del Dispositivo 1 🚨 | 9/10/2025, 14:38:36 |
| 4 | | |
| 5 | | |
| 6 | | |

*Nota:* autoridad propia tomado de Node-RED (2025)

Finalmente, esta pestaña está conectada con el bot de Telegram, el cual envía notificaciones instantáneas al detectar una alerta de emergencia como se ve en la figura 31 así la pestaña de geolocalización se convierte en un componente integral del servidor central ya que combina la localización, almacenamiento en la nube y comunicación inmediata con los responsables de seguridad consolidando un sistema de respuesta inteligente y automatizado.

**Figura 31**

*Chat Bot Telegram*




*Nota:* autoridad propia (2025)

### Objetivo 3 Pruebas

Una vez finalizadas las fases de diseño electrónico, desarrollo del firmware y ensamblaje físico del prototipo, se procedió a realizar un conjunto estructurado de pruebas orientadas a verificar el correcto funcionamiento de cada módulo, validar la interoperabilidad del sistema completo y evaluar su desempeño en condiciones reales. Las pruebas se desarrollaron en tres etapas: pruebas unitarias, pruebas de laboratorio y pruebas de campo, permitiendo así identificar fallas, documentar el comportamiento operativo y garantizar que el sistema de alarma cumpliera con los requisitos establecidos en el diseño.

#### Pruebas unitarias

En esta etapa se evaluó de forma individual y sistemática cada uno de los componentes que integran el sistema de alarma con botón de pánico. El propósito fue confirmar que cada módulo funcionara correctamente por separado y que existiera compatibilidad con el microcontrolador ESP32-C3 Super Mini antes de realizar la integración total.

El proceso inició con la validación del microcontrolador y del módulo GPS NEO-8N. Se verificó la lectura de pines digitales y análogos, así como la comunicación mediante el protocolo UART. Durante estas pruebas, el GPS registró un margen de error promedio de ±2.5 metros y un tiempo de adquisición inicial menor a 5 segundos. Inicialmente se emplearon los pines UART por defecto del ESP32-C3; sin embargo, al requerirse esos mismos pines para los módulos XBee, fue necesario reasignar pines alternativos para el GPS mediante programación, lo cual permitió mantener una comunicación estable sin interferencias.

Posteriormente, se verificó el funcionamiento del botón de pánico y de los LEDs indicadores configurados para representar diferentes estados del sistema: encendido, activación

de la alarma y nivel bajo de batería. Las pruebas confirmaron una respuesta inmediata de cada LED y la correcta asignación de pines digitales, validando la lógica programada.

Completada esta fase, se evaluaron los módulos de comunicación XBee PRO S3B, configurados a través del software XCTU. Se presentaron dificultades iniciales en el establecimiento del enlace debido a que los módulos no tenían asignados los roles correctos (coordinador y dispositivo final). Adicionalmente, se identificó que, ante configuraciones erróneas, los módulos entraban en modo de protección, requiriendo reinicios dentro de un margen máximo de 10 segundos. Una vez establecidos correctamente los parámetros y roles, se obtuvo una comunicación bidireccional estable y sin pérdidas de datos.

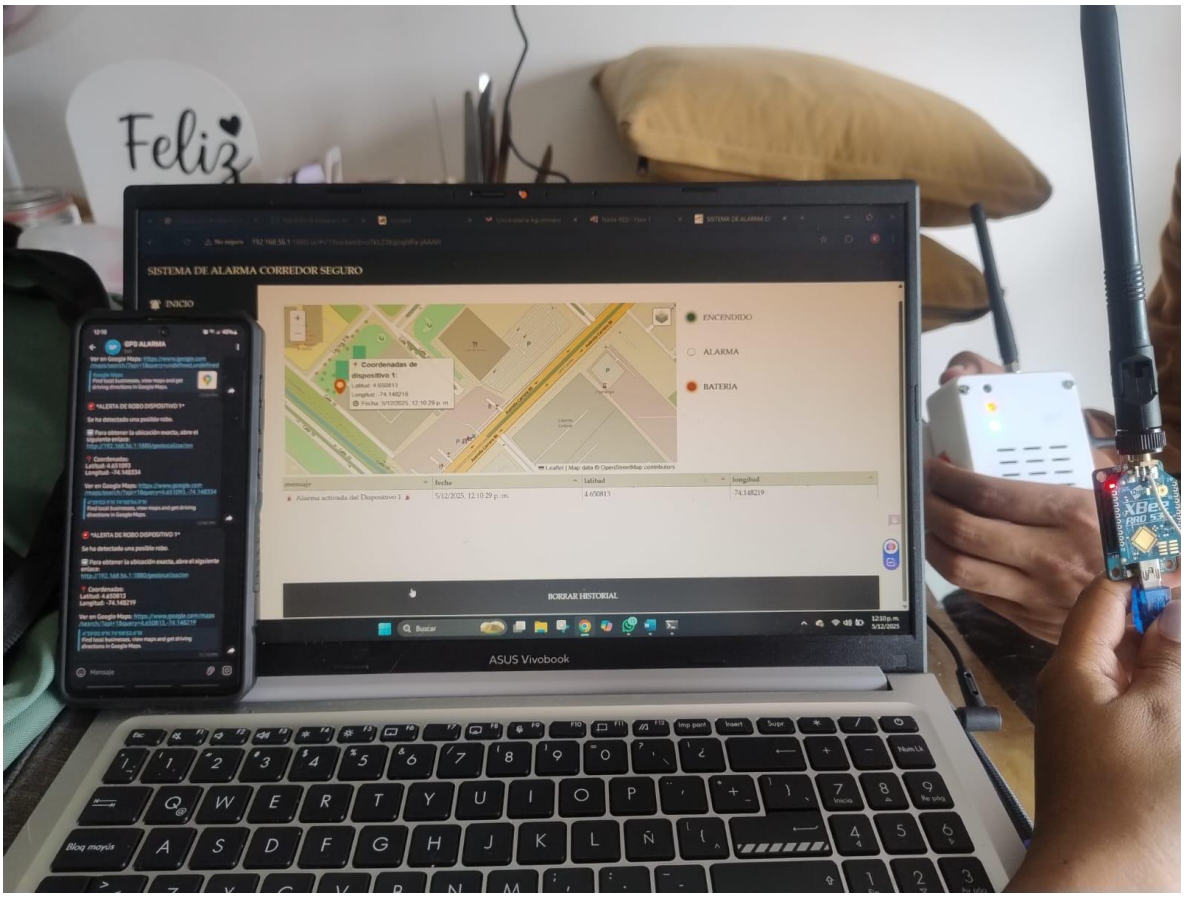
Finalmente, se realizó la configuración y prueba del sensor de monitoreo de voltaje INA3221. Este módulo presentó retos relacionados con la limitada documentación técnica, especialmente para identificar correctamente los pines SDA y SCL y la correspondencia entre los canales de medición. Luego de múltiples pruebas, se determinó que los pines adecuados para la comunicación I2C eran el 8 (SDA) y 9 (SCL). Posterior a su correcta conexión, se validó la lectura de los canales mediante un código de prueba (ver Anexo F).

### Pruebas de laboratorio

Con todos los módulos verificados individualmente, se integraron los diferentes códigos para conformar el firmware final y se realizaron pruebas en un entorno controlado. En esta etapa se evaluó la capacidad del microcontrolador para gestionar procesos simultáneos, incluyendo: lectura del GPS, activación del botón de pánico, monitoreo de voltaje mediante el INA3221 y transmisión inalámbrica de datos.

**Figura 32**

*Sistema completo*



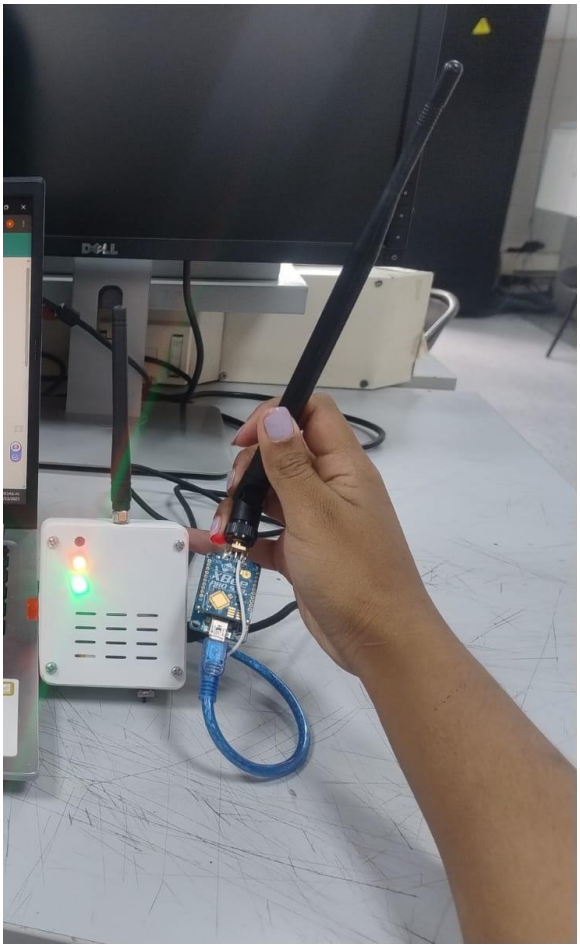

*Nota:* autoridad propia (2025)

La Figura 32 muestra el montaje utilizado durante las pruebas de laboratorio, en el que el prototipo envía información en tiempo real al sistema de monitoreo, sin embargo hay que tener en cuenta que el módulo XBee PRO S3B conectado al computador y enlazado con el dispositivo portátil, mientras la interfaz desarrollada en Node-RED despliega en pantalla el mapa con la ubicación recibida desde el módulo GPS. Asimismo, se aprecia el teléfono móvil con el bot de Telegram abierto, registrando los mensajes de alerta enviados por el sistema al activarse el botón de pánico. Este montaje permitió visualizar simultáneamente la transmisión inalámbrica, la

actualización de coordenadas y la recepción de notificaciones en los distintos componentes del sistema.

**Figura 33**

*antenas*



*Nota:* autoridad propia (2025)

Las pruebas se ejecutaron utilizando antenas omnidireccionales de 5 dB conectadas a los módulos XBee del transmisor y del receptor como se visualiza en la Figura 33, con el objetivo de medir la estabilidad de la comunicación dentro de un rango corto de operación. Se registró una entrega del 100% de las tramas enviadas en distancias entre 10 y 30 metros en interiores, sin pérdidas ni interrupciones. No obstante, al superar los 30 metros, la

comunicación se interrumpió y únicamente se restableció al aproximarse nuevamente al receptor.

El tiempo promedio de respuesta desde la activación del botón de pánico hasta la recepción y visualización del mensaje en el dashboard fue menor a 3 segundos, cumpliendo ampliamente el requisito de tiempo de respuesta inferior a 10 segundos planteado en la fase de diseño. En esta etapa también se confirmó el funcionamiento del INA3221, aunque se identificó un alto consumo energético derivado del envío de actualizaciones cada segundo por parte del ESP32-C3.

### Pruebas de campo

Una vez verificado el correcto funcionamiento del sistema en condiciones controladas, se procedió a realizar las pruebas de campo, cuyo propósito fue evaluar el desempeño real del prototipo en un entorno abierto y bajo condiciones similares a las que tendría durante su uso en los corredores seguros. Esta etapa resultó fundamental para determinar el alcance efectivo de la comunicación inalámbrica, la estabilidad del enlace en presencia de interferencias propias del entorno urbano, la precisión del GPS en movimiento y la capacidad de respuesta del sistema cuando el usuario se desplaza a diferentes distancias del punto de recepción. Las pruebas se desarrollaron en los alrededores de la Universitaria Agustiniana, un espacio que combina zonas abiertas, edificaciones de mediana altura, áreas con árboles y presencia constante de peatones, aspectos que permitieron observar el comportamiento del enlace de comunicación frente a obstáculos, reflexiones, atenuación de señal y variaciones ambientales.

**Figura 34**

*Antena Yagi*

*Nota:* autoridad propia (2025)

Para esta fase se sustituyó la antena omnidireccional de 5 dB utilizada en laboratorio por una antena Yagi de 17 dB, con el objetivo de ampliar el alcance de la comunicación y evaluar la capacidad del sistema bajo condiciones óptimas de recepción. En la figura 34 se observa la antena Yagi, la cual al ser direccional, concentra la energía en un haz estrecho, lo que incrementa la potencia en la dirección hacia la que apunta y favorece mayores distancias de transmisión. Sin embargo, esta misma característica limita su utilidad práctica, ya que únicamente cubre un tramo de la trayectoria y no ofrece cobertura lateral, condición que resulta desfavorable para escenarios donde el usuario puede moverse en diferentes direcciones, como es el caso de los corredores

seguros. Aun así, su uso permitió medir de manera más precisa el alcance máximo teórico del enlace inalámbrico.

**Figura 35**

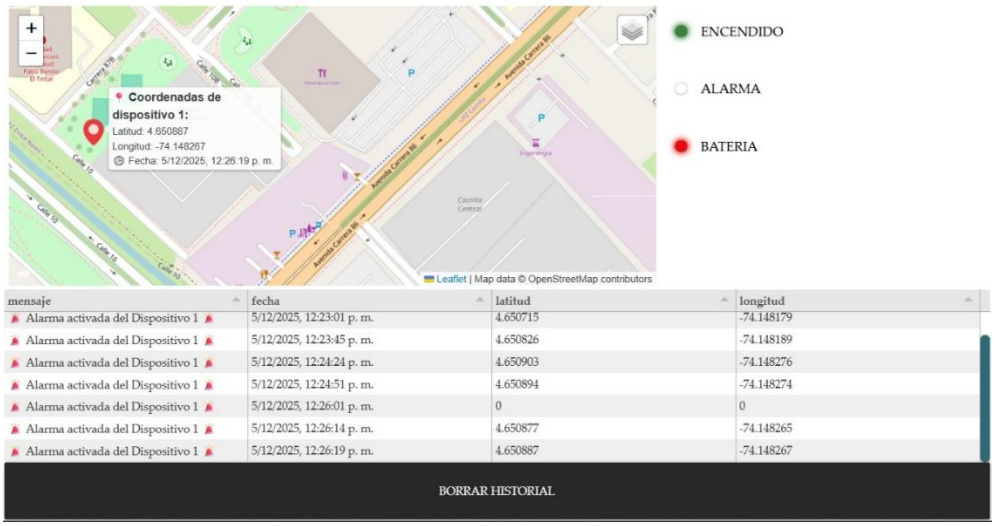
*Pruebas a 20 metros*

*Nota:* autoridad propia (2025)

Para la prueba correspondiente a los 20 metros, ilustrada en la Figura 35, el dispositivo emisor fue trasladado por uno de los integrantes del equipo hasta el punto previamente establecido dentro del corredor de evaluación. Esta acción permitió observar la separación física real entre el prototipo y la central de monitoreo. En esta etapa se procedió a activar el botón de pánico y a transmitir las coordenadas GPS, con el propósito de registrar el comportamiento del enlace inalámbrico en un tramo corto y con línea de vista parcialmente obstruida.

**Figura 36**

*Visualización prueba de 20 metros*




*Nota:* autoridad propia (2025)

La Figura 36 complementa esta prueba al mostrar la visualización obtenida en la central de monitoreo en el momento exacto en que se recibieron los datos enviados por el dispositivo. En esta captura de pantalla se observa la actualización automática del mapa dentro de la interfaz de Node-RED, junto con el registro del evento generado por el sistema, evidenciando la recepción de las coordenadas y el estado del dispositivo durante el ejercicio de campo.

**Figura 37**

*Medición de distancia de la prueba*


*Nota:* imagen tomada de Google Maps (2025)

Finalmente, la Figura 37 presenta la verificación externa realizada mediante Google Maps, donde se confirma la distancia aproximada entre el punto en el que se encontraba el dispositivo y la ubicación de la central de monitoreo. Esta referencia geográfica se empleó para complementar la documentación de la prueba, asegurando la correspondencia entre la distancia medida en campo y la ubicación reportada visualmente en el entorno digital.

En cada punto se efectuó la activación del botón de pánico, el envío automático y manual de coordenadas GPS, la verificación del tiempo de llegada al servidor, el análisis de estabilidad de la señal XBee, la inspección de pérdidas de paquetes y la observación del comportamiento del

sistema ante la presencia de obstáculos naturales o arquitectónicos. El receptor permaneció inmóvil y orientado hacia el punto de mayor visibilidad, permitiendo minimizar interferencias inmediatas y evaluar únicamente los efectos del entorno sobre la comunicación.

Tras finalizar la evaluación a 20 metros, se procedió a extender la distancia del recorrido con el fin de analizar el desempeño del sistema en condiciones de mayor separación entre el dispositivo y la central de monitoreo. Para esta siguiente etapa, correspondiente a la prueba realizada a 100 metros, se aplicó el mismo procedimiento metodológico, manteniendo la orientación fija de la antena receptora y desplazando el emisor a lo largo del corredor hasta alcanzar la distancia establecida. Las figuras que acompañan esta sección permiten observar de manera progresiva la posición física del operador, la respuesta generada en la plataforma de monitoreo y la verificación de la distancia a través de herramientas externas, proporcionando una visión integral del funcionamiento del sistema bajo un escenario de mayor alcance

**Figura 38**

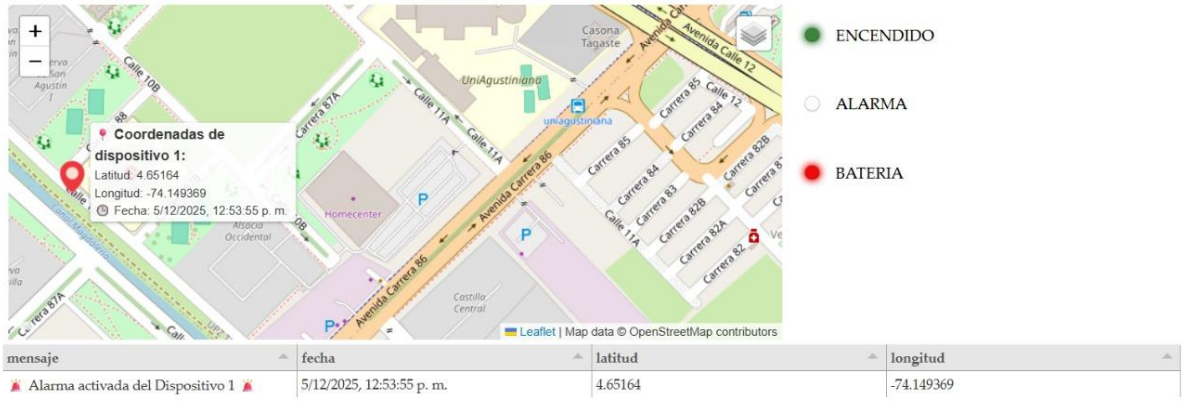
*Pruebas a 100 metros*


*Nota:* autoridad propia (2025)

En la prueba correspondiente a los 100 metros, representada en la Figura 38, el integrante del equipo encargado del traslado del dispositivo se ubicó en el punto predeterminado del recorrido, permitiendo observar la separación real respecto a la central de monitoreo. En este punto se realizó nuevamente la activación del botón de pánico y el envío automático de coordenadas GPS, con el objetivo de registrar el comportamiento del enlace inalámbrico a una distancia considerablemente mayor que en la prueba anterior.

**Figura 39**

*Visualización prueba de 100 metros*




*Nota:* autoridad propia (2025)

La Figura 39 muestra la información visualizada en la central de monitoreo durante esta prueba. En la captura de pantalla pueden apreciarse la actualización del mapa, la recepción de las coordenadas enviadas por el prototipo y el registro del evento en la plataforma de Node-RED. Esta visualización permitió verificar que, a pesar del incremento en la distancia, el sistema continuó procesando y mostrando los datos generados por el dispositivo en tiempo real.

**Figura 40**

*Medición de distancia de la prueba de 100 metros*


*Nota:* imagen tomada de Google Maps (2025)

Finalmente, la Figura 40 presenta la verificación de la distancia mediante Google Maps, utilizada como herramienta de referencia para corroborar el punto exacto en el que se encontraba el dispositivo durante la prueba de los 100 metros. Esta comprobación permitió complementar la documentación del experimento, asegurando que la ubicación registrada en el mapa coincidía con la distancia estimada en campo y validando así la correspondencia espacial del ensayo.

Los resultados mostraron que, en el rango de 10 a 100 metros, la comunicación se mantuvo completamente estable, sin pérdidas de paquetes y con un tiempo de respuesta inferior a tres segundos. Las coordenadas del GPS se recibieron correctamente y no se detectaron

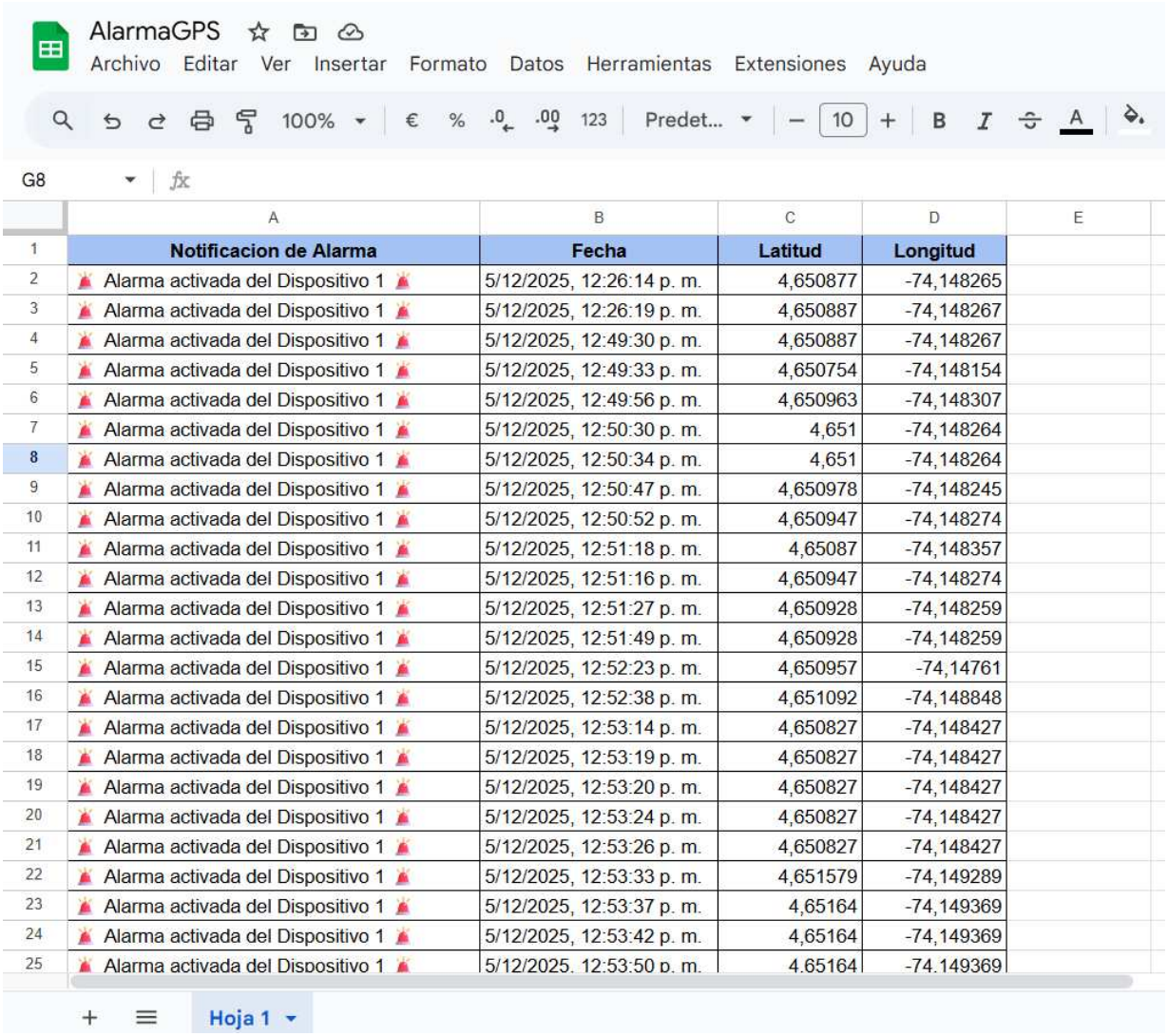
variaciones bruscas ni errores significativos en la lectura. Sin embargo, al superar los 30 metros comenzaron a observarse intermitencias en la señal, principalmente ocasionadas por la presencia de personas en movimiento y elementos estructurales que producían atenuación y dispersión. Aunque en este intervalo todavía se lograban transmitir los datos, el sistema mostró una ligera degradación en la estabilidad del enlace.

Entre los 40 y 50 metros, la degradación fue más notable. La señal presentó fluctuaciones constantes y, en varias ocasiones, el receptor no logró obtener las tramas completas enviadas por el transmisor. En este rango, la comunicación dependía de manera crítica de la alineación del dispositivo con el eje principal de la antena Yagi, lo que evidencia la alta sensibilidad del sistema a la direccionalidad de esta antena. Adicionalmente, el tiempo de respuesta aumentó debido a los reintentos de envío, y la activación del botón de pánico no siempre generaba una transmisión exitosa. Al superar los 50 metros, la comunicación se perdió completamente y no fue posible restablecerla incluso realizando ajustes en el ángulo del dispositivo, confirmando el límite del alcance bajo estas condiciones específicas.

En cuanto al desempeño del GPS durante las pruebas de campo, el módulo mantuvo un margen de error entre ±2 y ±3 metros, con una lectura estable y constante durante los desplazamientos. El tiempo de adquisición de señal satelital se mantuvo dentro de los parámetros observados en las pruebas de laboratorio y la lectura no se vio significativamente afectada por la presencia de edificaciones o movimiento, lo que confirma la confiabilidad del GPS en entornos urbanos abiertos.

**Figura 41**

*Base de datos de las pruebas*



AlarmaGPS

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda

G8

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Notificacion de Alarma | Fecha | Latitud | Longitud | |
| 2 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:26:14 p. m. | 4,650877 | -74,148265 | |
| 3 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:26:19 p. m. | 4,650887 | -74,148267 | |
| 4 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:49:30 p. m. | 4,650887 | -74,148267 | |
| 5 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:49:33 p. m. | 4,650754 | -74,148154 | |
| 6 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:49:56 p. m. | 4,650963 | -74,148307 | |
| 7 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:50:30 p. m. | 4,651 | -74,148264 | |
| 8 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:50:34 p. m. | 4,651 | -74,148264 | |
| 9 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:50:47 p. m. | 4,650978 | -74,148245 | |
| 10 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:50:52 p. m. | 4,650947 | -74,148274 | |
| 11 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:51:18 p. m. | 4,65087 | -74,148357 | |
| 12 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:51:16 p. m. | 4,650947 | -74,148274 | |
| 13 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:51:27 p. m. | 4,650928 | -74,148259 | |
| 14 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:51:49 p. m. | 4,650928 | -74,148259 | |
| 15 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:52:23 p. m. | 4,650957 | -74,14761 | |
| 16 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:52:38 p. m. | 4,651092 | -74,148848 | |
| 17 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:53:14 p. m. | 4,650827 | -74,148427 | |
| 18 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:53:19 p. m. | 4,650827 | -74,148427 | |
| 19 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:53:20 p. m. | 4,650827 | -74,148427 | |
| 20 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:53:24 p. m. | 4,650827 | -74,148427 | |
| 21 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:53:26 p. m. | 4,650827 | -74,148427 | |
| 22 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:53:33 p. m. | 4,651579 | -74,149289 | |
| 23 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:53:37 p. m. | 4,65164 | -74,149369 | |
| 24 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:53:42 p. m. | 4,65164 | -74,149369 | |
| 25 | Alarma activada del Dispositivo 1 | 5/12/2025, 12:53:50 p. m. | 4,65164 | -74,149369 | |

Hoja 1

*Nota:* Autoridad propia (2025)

Cuando el usuario activa la alarma, el sistema registra automáticamente el evento en una hoja de cálculo de Google Sheets almacenada en Google Drive como se visualiza en la figura 41, en esta hoja se guarda de forma inmediata la fecha, la hora exacta de activación y las coordenadas geográficas (latitud y longitud) capturadas por el dispositivo. Esta información se consolida en un historial que permite identificar y analizar los lugares donde se han presentado

mayores índices de robo, facilitando así la detección de patrones y la toma de decisiones basadas en datos.

En conjunto, las pruebas de campo permitieron determinar que el sistema es funcional y estable dentro de rangos prácticos de hasta 30 o 35 metros utilizando antenas estándar. La antena Yagi permitió extender el alcance, pero a costa de sacrificar la cobertura lateral, lo cual no es recomendable para una implementación real en corredores seguros. Los resultados también sugieren la necesidad de evaluar alternativas como antenas omnidireccionales de mayor ganancia, repetidores estratégicamente ubicados o la integración de redes malladas que permitan ampliar el rango sin comprometer la estabilidad de la señal. Estas pruebas proporcionan información crucial para optimizar el diseño del sistema y asegurar un funcionamiento adecuado en escenarios reales de seguridad universitaria

## Conclusiones

- Con las diferentes pruebas realizadas en el entorno universitario en días normales de clases con un gran flujo de estudiantes, permitió establecer una comunicación efectiva entre el dispositivo del usuario y el servidor central mediante la arquitectura cliente-servidor implementada en Node-RED. Asimismo, se evidenció el rendimiento de antenas con ganancias de 5 dB Y 15 dB, alcanzando distancias de comunicación muy cortas y manteniendo una precisión de geolocalización dentro del margen de error esperado (1.5m)
- La revisión del estado del arte confirma la pertinencia de la tecnología RFID**,** destacando su bajo costo, facilidad de integración, amplia documentación y efectividad en aplicaciones de seguridad, seguimiento e identificación. En comparación con alternativas más avanzadas como biometría o visión por computadora, RFID ofrece un equilibrio óptimo entre costo, funcionalidad y simplicidad para el contexto académico.
- La arquitectura seleccionada, basada en microcontroladores como ESP32 o Raspberry Pi y bases de datos locales o remotas, permite un sistema escalable y adaptable, capaz de integrarse con futuras mejoras como módulos más completos de inventario, análisis de datos o integración institucional. La selección cuidadosa de componentes garantiza un diseño robusto, replicable y de baja complejidad de implementación.
- El modelo metodológico en V facilitó un desarrollo claro y estructurado**,** permitiendo relacionar cada fase de diseño con su verificación correspondiente.

Esto aseguró la coherencia entre necesidades, especificaciones y comportamiento del prototipo, contribuyendo a su funcionalidad y estabilidad.

## Recomendaciones

- En primer lugar, se recomienda realizar pruebas adicionales empleando antenas de mayor ganancia, con el propósito de ampliar el rango de comunicación y mejorar la estabilidad del enlace entre el dispositivo portátil y el servidor central, especialmente en entornos con alta interferencia o barreras físicas que puedan afectar la transmisión de datos.
- Asimismo, se sugiere evaluar el rediseño estructural del dispositivo, buscando minimizar su tamaño y optimizar la disposición de los componentes electrónicos. Para ello, se propone el uso de una placa ZX908 o de dimensiones reducidas, lo que permitiría adaptar el sistema como una manilla o llavero más ergonómico, garantizando mayor comodidad, portabilidad y aceptación por parte del usuario.
- De igual manera, se recomienda implementar una etapa de pruebas prolongadas en condiciones reales de uso, con el fin de monitorear la durabilidad del sistema, el rendimiento energético y la precisión del GPS durante diferentes jornadas y escenarios ambientales.
- Por último, sería conveniente ampliar las funcionalidades del sistema integrando tecnologías complementarias, como conectividad Wi-Fi o GSM, lo cual permitiría enviar alertas incluso fuera del rango de radiofrecuencia del XBee, asegurando una cobertura más amplia y una respuesta más oportuna ante emergencias.

## Referencias

Banco Santander S.A. (15 de Marzo de 2022). Wearables: ¿qué son y para qué se utilizan? *Santander*. Obtenido de https://www.santander.com/es/stories/wearables-que-son-y-para-que-se-utilizan

Espinosa Espinosa, M. I., & Retana González , R. A. (2022). Prototipo de geolocalización para personas vulnerables: botón de pánico, SOS. *Ciencia Latina Revista Científica Multidisciplinar*, 6. Obtenido de https://doi.org/10.37811/cl_rcm.v6i6.3818

Acecho Seguridad. (27 de Agosto de 2024). ¿Qué es el botón del pánico? ¿Cómo funciona? *Acecho Seguridad*. Obtenido de https://www.acecho.es/el-boton-del-panico-todo-lo-que-se-necesita-saber-sobre-el/

Acero Soria, J. M., Flores Prieto, A., Herrada Rivera, A. L., & Saccsara Torres, M. L. (2018) Pulseras Securelet. [Trabajo de investigacion. UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS]. Obtenido de https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/624063/FLORES_PA.pdf?sequence=14&isAllowed=y

Aldana, J. (2024). *Comportamiento de la criminalidad en la region Bogota-Cundinamarca.* Bogota D.C. Obtenido de https://www.sdp.gov.co/sites/default/files/comportamiento_criminalidad_region_bogota_cundinamarca_1.pdf

Altamirano Cabrera, M., Rafael Perez, E., Morales Hernandez, M., Benites Quecha, C., & Sanchez Mendez, J. E. (2016). Prototipo de dispositivo de alerta “Ay Tá” . *Revista de Tecnología e Innovación*, 13-21. Obtenido de

https://www.ecorfan.org/bolivia/researchjournals/Tecnologia_e_innovacion/vol4num13/Revista_de_Tecnologia_e_Innovacion_V4_N13_2.pdf

Añezco Wilches, S. F., & Sanchez Merchan, M. S. (2023). *Diseño e Implementacion de un sistema de alerta de emergencia con ubicación GPS.* Cuenca: [Trabajo de grado Universidad de Azuay facultad de ciencia y tecnología] Obtenido de https://dspace.uazuay.edu.ec/bitstream/datos/13428/1/18953.pdf

Ardila Rojas, C. E. (2023). *Estudio exploratorio para la validación del modelo startup "Biky" que integra tecnologías sostenibles e informáticas en beneficio de la seguridad del transporte no motorizado en la ciudad de Bogotá.* Bogotá: [Trabajo de maestría Universidad Nacional Abierta y a Distancia UNAD]. Obtenido de https://repository.unad.edu.co/bitstream/handle/10596/55069/ceardilar%20.pdf?sequence=3&isAllowed=y

Benavides Segura, L. I., & Cárdenas Espinoza, B. D. (2021). *Implementación de un dispositivo IoT en tecnología LoRa para la geolocalización y monitoreo fisiológico de personas en lugares Turísticos.* Riobamba: [Trabajo de grado ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO]. Obtenido de https://dspace.espoch.edu.ec:8080/server/api/core/bitstreams/9ff54ede-7313-4c96-9c55-9db3f2975320/content

Cifuentes Mendieta, A. F., & Silva Arias, S. G. (2021). *Desarrollo de una aplicación Android para anunciar la presencia de sobrevivientes mediante un botón de pánico.* Bogotá, Colombia. [Trabajo de grado. UNIVERSIDAD ANTONIO NARIÑO]. Obtenido de https://repositorio.uan.edu.co/server/api/core/bitstreams/8bf954f0-736c-4432-ab91-d4a310e507df/content

Coder y Chill (15 de Julio de 2023). ¿Qué es el GPS? *Medium*. Obtenido de https://medium.com/@diego.coder/qu%C3%A9-es-el-gps-b2f0b2299972

Congreso de la Republica de Colombia. (30 de Julio de 2009). Ley 1341 de 2009 *Alcaldía Mayor de Bogotá D.C.* Obtenido de https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=36913

Congreso de la Republica de Colombia. (17 de abril de 2013). LEY ESTATUTARIA 1621 DE 2013 *secretaria del senado*. Obtenido de http://www.secretariasenado.gov.co/senado/basedoc/ley_1621_2013.html

Congreso de la Republica de Colombia. (2016). Ley 1801 de 2016 *Código Nacional de Seguridad y Convivencia Ciudadana.* Bogotá D.C.: Función Publica. Obtenido de https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=80538

DNP. (2025). *Informe municipal de Seguridad y Convivencia Ciudadana.* Bogotá D.C. Obtenido de https://concejodebogota.gov.co/cbogota/site/artic/20250203/asocfile/20250203082639/informe_anual_de_seguridad_bogot___colombia_concejal_espinosa_1_compressed.pdf

Escobar, J. P. (6 de Julio de 1991). *Función Publica*. Obtenido de https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=4125

Espinosa. (2024). *Informe anual de la seguridad en Bogotá y Colombia.* Bogotá. Obtenido de https://concejodebogota.gov.co/cbogota/site/artic/20250203/asocfile/20250203082639/informe_anual_de_seguridad_bogot___colombia_concejal_espinosa_1_compressed.pdf

Garzón Olivo, F. L. (2024). *Implementación de un sistema de seguridad con IoT para el Asadero Gilgal basado en un botón de pánico y alarma con módulo GPS.* Santa Elena: [Trabajo de grado UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA]. Obtenido de

https://repositorio.upse.edu.ec/server/api/core/bitstreams/debdec10-9fe1-4db1-8aa0-5606b7862f45/content

Gordon Franco, O., & Martínez Herrera, J. (2019). *Diseño de red LoRaWAN™ para el servicio de un botón de pánico en Barranquilla.* Barranquilla Colombia. [Trabajo de grado UNIVERSIDAD DEL NORTE] Obtenido de https://www.academia.edu/74470570/Dise%C3%B1o_de_red_LoRaWAN_para_el_servicio_de_un_bot%C3%B3n_de_p%C3%A1nico_en_Barranquilla

Macías Rojas, M. E. (2020). *Espacios para la no violencia: el caso del Corredor Seguro para Mujeres en Ciudad Juárez en 2020.* Cd. Juárez, Chihuahua: [Trabajo de grado Universidad Autónoma de Ciudad Juárez]. Obtenido de https://www.researchgate.net/profile/Elena-Macias/publication/349043775_Universidad_Autonoma_de_Ciudad_Juarez/links/601c78b0299bf1cc26a2df90/Universidad-Autonoma-de-Ciudad-Juarez.pdf

Ministerio de Tecnologías de la Información y las Comunicaciones. (26 de mayo de 2015). *MINTIC Colombia*. DECRETO 1078 DE 2015 Obtenido de https://normograma.mintic.gov.co/mintic/compilacion/docs/decreto_1078_2015.htm

Montesdeoca García, C. A. (2022). *Desarrollo de aplicación móvil de geolocalización para alarma y notificación de pánico en plataforma Android.* Bogotá D.C.: Escuela Politécnica Nacional. Obtenido de https://bibdigital.epn.edu.ec/bitstream/15000/22680/1/CD%2012163.pdf

Parra Logroño, R. A. (2023). *Desarrollo de una aplicación tipo botón de pánico para la comunidad de Ainche.* Riobamba: [Trabajo de tecnólogo Instituto superior tecnológico Dr Misael Acosta S]. Obtenido de https://dspace-

api.istmas.edu.ec/server/api/core/bitstreams/d364349b-5163-4dd6-9f80-850697be3b2f/content

Passarelli, A. M. (2015). *Corredor seguro: Uso, disputas y apropiación del espacio público en la ciudad de La Plata.* Buenos Aires: [Trabajo de grado UNIVERSIDAD NACIONAL DE LA PLATA]. Obtenido de https://www.memoria.fahce.unlp.edu.ar/tesis/te.1133/te.1133.pdf

Quijano Cárdenas, A. P. (2021). *Diseño e implementación de monitoreo de Heart rate, temperatura y movimiento en población Adulto Mayor con sistema de alerta.* Bogotá D.C.: [Trabajo de grado LA UNIVERSIDAD DE LOS ANDES]. Obtenido de https://repositorio.uniandes.edu.co/server/api/core/bitstreams/9230e887-0319-4a5d-a22d-8fae9fcb1f58/content

Red Hat, Inc. (20 de enero de 2023). *Red Hat*. ¿Qué es el Internet de las cosas (IoT)? Obtenido de https://www.redhat.com/es/topics/internet-of-things/what-is-iot

Robayo, M. A. (28 de diciembre de 2023). *Legis Ámbito jurídico*. Seguridad ciudadana y seguridad pública: la prevención del crimen frente a la atomización del delito Obtenido de https://www.ambitojuridico.com/noticias/analisis/penal/seguridad-ciudadana-y-seguridad-publica-la-prevencion-del-crimen-frente-la

Romero Mosquera, R. A., & Torres Ramos, J. F. (2022). *Desarrollo e implementación de un sistema de seguridad para camiones transportistas basado en un botón de pánico y alarma usando módulos GPS.* Guayaquil: [Trabajo de grado ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL]. Obtenido de https://www.dspace.espol.edu.ec/bitstream/123456789/57108/1/T-113042%20ROMERO%20-%20TORRES.pdf

Salazar Tobalina, L. (2011). *Diseño de un sistema de localización y seguimiento de personas.* Madrid: [Trabajo de grado Universidad Carlos III de Madrid]. Obtenido de https://files.core.ac.uk/download/pdf/30044905.pdf

Saraguro Eras, A. L. (2020). *Prototipo Electrónico para prevenir ataques de feminicidio mediante una alerta inmediata de GPS.* Quito: [Trabajo de grado UNIVERSIDAD TECNOLÓGICA ISRAEL]. Obtenido de http://repositorio.uisrael.edu.ec/bitstream/47000/2608/1/UISRAEL-EC-ELTD-378.242-037.pdf

Secretaria de Seguridad, Convivencia y Justicia de Colombia. (28 de noviembre de 2024). *scj.gov*. ¿Corredores Universitarios Seguros? la nueva estrategia para prevenir delitos en 13 zonas universitarias Obtenido de https://scj.gov.co/prensa/noticias/corredores-universitarios-seguros-la-nueva-estrategia-para-prevenir-delitos-en-13?utm_source=chatgpt.com

Suarez, J. D. (2021). *Privatización de la seguridad en Bogotá: el caso de los corredores universitarios seguros y el rol de las empresas de seguridad privada.* Bogotá D.C.: [Trabajo de grado Universidad de los Andes] Obtenido de https://repositorio.uniandes.edu.co/server/api/core/bitstreams/a496d791-c032-4384-b0c9-04e249bcc1af/content

Telegram. (25 de octubre de 2025). Bot API. *core.telegram.org*. Obtenido de https://telegram.org/faq

Vélez Flórez, M. (2023). *Atraxia.* Bogotá Colombia. [Trabajo de grado Universidad de los Andes]. Obtenido de

https://repositorio.uniandes.edu.co/server/api/core/bitstreams/8e629765-2860-4d9d-8aac-96099a675053/content

# Anexos

## Anexo A

Código de la recepción de datos del XBee PRO S3B

```
#include <HardwareSerial.h>

// Puerto serial para el XBee
HardwareSerial XBeeSerial(1);

// Pines donde conectarás el XBee al ESP32-C3
#define XBEE_TX 21   // ESP32 TX -> DIN (XBee pin 3)
#define XBEE_RX 20   // ESP32 RX -> DOUT (XBee pin 2)

void setup() {
  // Serial para depuración
  Serial.begin(9600);
  delay(1000);
  Serial.println("Receptor ESP32-C3 con XBee Pro S3B...");

  // Inicializamos el puerto serial para el XBee
  XBeeSerial.begin(9600, SERIAL_8N1, XBEE_RX, XBEE_TX);
  delay(1000);
}

void loop() {
  // Si hay datos desde el XBee, los mostramos en Serial
  while (XBeeSerial.available()) {
    char c = XBeeSerial.read();
    Serial.print(c);
  }
}
```

*Nota.* Fuente: autoría propia (2025)

## Anexo B

Código de la transmisión de datos del XBee PRO S3B

```
#include <HardwareSerial.h>

// Creamos un puerto serial para el XBee (usar los pines de tu
ESP32-C3)
HardwareSerial XBeeSerial(1);

// Pines donde conectarás el XBee al ESP32-C3
#define XBEE_TX 21   // ESP32 TX -> DIN (XBee pin 3)
#define XBEE_RX 20   // ESP32 RX -> DOUT (XBee pin 2)
```

```
void setup() {
  // Serial para depuración en el monitor serie
  Serial.begin(9600);
  delay(1000);
  Serial.println("Iniciando ESP32-C3 con XBee...");

  // Inicializamos el puerto serial del XBee
  XBeeSerial.begin(9600, SERIAL_8N1, XBEE_RX, XBEE_TX);
  delay(1000);

  Serial.println("Listo. Enviando datos al XBee...");
}

void loop() {
  // Enviar un mensaje cada 2 segundos
  XBeeSerial.println("Hola desde ESP32-C3 con XBee Pro S3B!");
  Serial.println("Mensaje enviado...");

  delay(2000);

  // Si llega algo desde el XBee (otro módulo respondió), lo
mostramos
  while (XBeeSerial.available()) {
    char c = XBeeSerial.read();
    Serial.print(c);
  }
}
```

*Nota.* Fuente: autoría propia (2025)

**Anexo C**

Código de la programación del GPS NEO-8M

```
#include <TinyGPSPlus.h>
#include <HardwareSerial.h>

// Objeto para la comunicación serial con el GPS en los pines
especificados
HardwareSerial gpsSerial(1);

// Objeto para la librería TinyGPSPlus
TinyGPSPlus gps;
// Tiempo del último evento
unsigned long previousMillis = 0;
// Intervalo de 10 segundos para imprimir datos
const long interval = 10000;

void setup() {
  // Inicializa la comunicación serial con el monitor de la
computadora
```

```
  Serial.begin(115200);
  // Inicializa la comunicación serial con el módulo GPS en los
pines 2 (RX) y 3 (TX)
  gpsSerial.begin(9600, SERIAL_8N1, 20, 21); // RX: 2, TX: 3

  Serial.println("Iniciando lectura de datos GPS...");
}

void loop() {
  // Lee los datos del GPS si están disponibles
  while (gpsSerial.available() > 0) {
    gps.encode(gpsSerial.read());
  }

  // Verifica si es momento de imprimir los datos
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    if (gps.location.isValid()) {
      Serial.print("Latitud: ");
      Serial.println(gps.location.lat(), 6); // Seis decimales de
precisión
      Serial.print("Longitud: ");
      Serial.println(gps.location.lng(), 6);
      Serial.print("Altitud: ");
      Serial.println(gps.altitude.meters(), 2); // Dos decimales
      Serial.print("Satélites: ");
      Serial.println(gps.satellites.value());
      Serial.println("--------------------");
    } else {
      Serial.println("Buscando señal GPS... ");
    }
  }
}
```

*Nota.* autoría propia (2025)

**Anexo D**

Código del Botón

```
#include <Arduino.h>

// Pin del botón
const int botonPin = 3;

// Variables
int alarma_off = 0;
int alarma_on = 0;
```

```
     void setup() {
       Serial.begin(115200);
       pinMode(botonPin, INPUT_PULLUP);  // Botón con
resistencia pull-up interna
     }

     void loop() {
       // Leer el estado del botón
       int estadoBoton = digitalRead(botonPin);

       // Resetear valores
       alarma_off = 0;
       alarma_on = 0;

       // Si está presionado (LOW por pull-up), se activa
alarma_on
       if (estadoBoton == LOW) {
         alarma_on = 1;
       } else {
         alarma_off = 1;
       }

       // Crear JSON manualmente
       String jsonString = "{";
       jsonString += "\"alarma_off\":" + String(alarma_off) +
",";
       jsonString += "\"alarma_on\":" + String(alarma_on);
       jsonString += "}";

       // Enviar por serial
       Serial.println(jsonString);

       delay(500);  // Medio segundo entre envíos
     }
```

*Nota.* autoría propia (2025)

**Anexo E**

Código del Botón y los LEDs

```
#include <TinyGPSPlus.h>
#include <SoftwareSerial.h>
#include <Arduino.h>

// ----------------- GPS -----------------
SoftwareSerial gpsSerial(4, 5); // RX, TX
TinyGPSPlus gps;
```

```
      unsigned long previousMillis = 0;
      const long interval = 1000; // 1 segundo (puedes cambiar a 10000
ms = 10s)

      // ---------------- Botón y LEDs ----------------
      const int botonPin = 3;
      const int led_encendido = 6;
      const int led_alarma = 7;
      const int led_bateria = 8;

      int alarma_off = 0;
      int alarma_on = 0;

      void setup() {
        Serial.begin(115200);   // Monitor serial
        gpsSerial.begin(9600);  // GPS NEO-8M

        // Configuración de pines
        pinMode(botonPin, INPUT_PULLUP);
        pinMode(led_encendido, OUTPUT);
        pinMode(led_alarma, OUTPUT);
        pinMode(led_bateria, OUTPUT);

        // LED encendido siempre al iniciar
        digitalWrite(led_encendido, HIGH);

        // LED batería fijo en HIGH (puedes luego adaptar la lógica)
        digitalWrite(led_bateria, HIGH);

        Serial.println("{\"estado\":\"Iniciando sistema y GPS...\"}");
      }

      void loop() {
        // ----------- Lectura GPS -----------
        while (gpsSerial.available() > 0) {
          gps.encode(gpsSerial.read());
        }

        // ----------- Lectura Botón -----------
        int estadoBoton = digitalRead(botonPin);

        if (estadoBoton == LOW) {
          // Botón presionado
          alarma_on = 1;
          alarma_off = 0;
          digitalWrite(led_alarma, HIGH);
        } else {
          // Botón no presionado
          alarma_on = 0;
          alarma_off = 1;
          digitalWrite(led_alarma, LOW);
        }
```

```
        // ----------- Intervalo de envío JSON -----------
        unsigned long currentMillis = millis();
        if (currentMillis - previousMillis >= interval) {
          previousMillis = currentMillis;

          // ---- Construcción del JSON ----
          String jsonString = "{";
          jsonString += "\"alarma_off\":" + String(alarma_off) + ",";
          jsonString += "\"alarma_on\":" + String(alarma_on) + ",";
          jsonString += "\"led_encendido\":" +
String(digitalRead(led_encendido)) + ",";
          jsonString += "\"led_alarma\":" +
String(digitalRead(led_alarma)) + ",";
          jsonString += "\"led_bateria\":" +
String(digitalRead(led_bateria));

          if (gps.location.isValid()) {
            jsonString += ",\"latitud\":" + String(gps.location.lat(),
6);
            jsonString += ",\"longitud\":" + String(gps.location.lng(),
6);
          } else {
            jsonString += ",\"estado\":\"Buscando señal GPS... 🛰\"";
          }

          jsonString += "}";

          // ---- Enviar JSON ----
          Serial.println(jsonString);
        }
      }
```

*Nota.* Fuente: autoría propia (2025)

**Anexo F**

Código del INA 3221

```
#include <Wire.h>

// Pines I2C del ESP32-C3
#define SDA_PIN 8
#define SCL_PIN 9

// Dirección detectada
#define INA_ADDR 0x41

// Registros
#define REG_SHUNT_CH1 0x01
#define REG_BUS_CH1   0x02
```

```
      // Rango batería (0% = 2.0 V, 100% = 3.7 V)
      const float V_MIN = 2.0f;
      const float V_MAX = 3.7f;

      void setup() {
        Serial.begin(115200);
        delay(300);
        Serial.println("\n🔍 Lectura INA3221 corregida (CH1)");

        Wire.begin(SDA_PIN, SCL_PIN);
        Wire.setClock(100000);
        delay(100);
      }

      bool read16_from(uint8_t addr, uint8_t reg, uint16_t &out) {
        Wire.beginTransmission(addr);
        Wire.write(reg);
        if (Wire.endTransmission(false) != 0) return false;
        Wire.requestFrom((int)addr, (int)2);
        if (Wire.available() < 2) return false;
        out = (Wire.read() << 8) | Wire.read();
        return true;
      }

      void loop() {
        uint16_t rawBus_u = 0;
        uint16_t rawShunt_u = 0;

        bool okBus = read16_from(INA_ADDR, REG_BUS_CH1, rawBus_u);
        bool okShunt = read16_from(INA_ADDR, REG_SHUNT_CH1,
rawShunt_u);

        if (!okBus && !okShunt) {
          Serial.println("⚠️ Sin respuesta (NACK) al leer registros.
Revisar VPU->3.3V, pull-ups y cableado.");
          delay(1500);
          return;
        }

        // Interpreta bus con la corrección de escala del módulo:
        // tu módulo tiene un divisor ~8x, por eso usamos: bus (V) =
rawBus * 0.001 (1 mV/bit)
        float busVoltage = rawBus_u * 0.001f; // en voltios

        // Interpreta shunt como signed 16-bit (two's complement)
        int16_t rawShunt_s = (int16_t)rawShunt_u;
        float shunt_mV = rawShunt_s * 0.04f; // cada bit = 40 uV ->
0.04 mV

        float batteryVoltage = busVoltage + (shunt_mV / 1000.0f);
```

```
        // Porcentaje entre V_MIN y V_MAX
        float pct = ((batteryVoltage - V_MIN) / (V_MAX - V_MIN)) *
100.0f;
        if (pct < 0.0f) pct = 0.0f;
        if (pct > 100.0f) pct = 100.0f;

        // Imprime resultados
        Serial.println("---- Lectura CH1 corregida ----");
        Serial.print("Voltaje bus (corr): "); Serial.print(busVoltage,
4); Serial.println(" V");
        Serial.print("Voltaje batería (total): ");
Serial.print(batteryVoltage, 4); Serial.println(" V");
        Serial.print("Nivel estimado: "); Serial.print(pct, 1);
Serial.println(" %");
        if (pct <= 20.0f) Serial.println("⚠️ Batería baja");
        Serial.println("------------------------------\n");

        delay(1500);
      }
```

*Nota.* Fuente: autoría propia (2025)

**Anexo G**

Código Final

```
#include <HardwareSerial.h>
#include <TinyGPSPlus.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <Arduino.h>

// ----------------- XBee -----------------
HardwareSerial XBeeSerial(1);
#define XBEE_TX 21   // ESP32 TX -> DIN (XBee pin 3)
#define XBEE_RX 20   // ESP32 RX -> DOUT (XBee pin 2)

// ----------------- GPS -----------------
SoftwareSerial gpsSerial(4, 5); // RX=4, TX=5
TinyGPSPlus gps;

// ----------------- Botón y LEDs -----------------
const int botonPin = 3;
const int led_encendido = 0;   // cambiado
const int led_alarma = 1;      // cambiado
const int led_bateria = 2;     // cambiado

int alarma_off = 0;
int alarma_on = 0;
```

```
      // Control del botón
      bool botonAnterior = HIGH;
      bool alarmaActiva = false;
      unsigned long tiempoAlarma = 0;

      // ----------------- INA3221 -----------------
      #define SDA_PIN 8
      #define SCL_PIN 9
      #define INA_ADDR 0x41
      #define REG_SHUNT_CH1 0x01
      #define REG_BUS_CH1   0x02

      // Rango batería (0% = 2.0 V, 100% = 3.7 V)
      const float V_MIN = 2.0f;
      const float V_MAX = 3.7f;

      // ----------------- Tiempos -----------------
      unsigned long previousMillis = 0;
      const long interval = 1000;
      const long duracionAlarma = 2000; // ms

      // ----------- Función lectura INA3221 -----------
      bool read16_from(uint8_t addr, uint8_t reg, uint16_t &out) {
        Wire.beginTransmission(addr);
        Wire.write(reg);
        if (Wire.endTransmission(false) != 0) return false;
        Wire.requestFrom((int)addr, (int)2);
        if (Wire.available() < 2) return false;
        out = (Wire.read() << 8) | Wire.read();
        return true;
      }

      // =====================================================
      // ==================== SETUP ==========================
      // =====================================================
      void setup() {
        Serial.begin(115200);
        delay(1000);
        Serial.println("🚀 Iniciando ESP32-C3 con XBee, GPS e
INA3221...");

        // Inicializamos XBee
        XBeeSerial.begin(9600, SERIAL_8N1, XBEE_RX, XBEE_TX);

        // Inicializamos GPS
        gpsSerial.begin(9600);

        // Inicializamos I2C
        Wire.begin(SDA_PIN, SCL_PIN);
        Wire.setClock(100000);
```

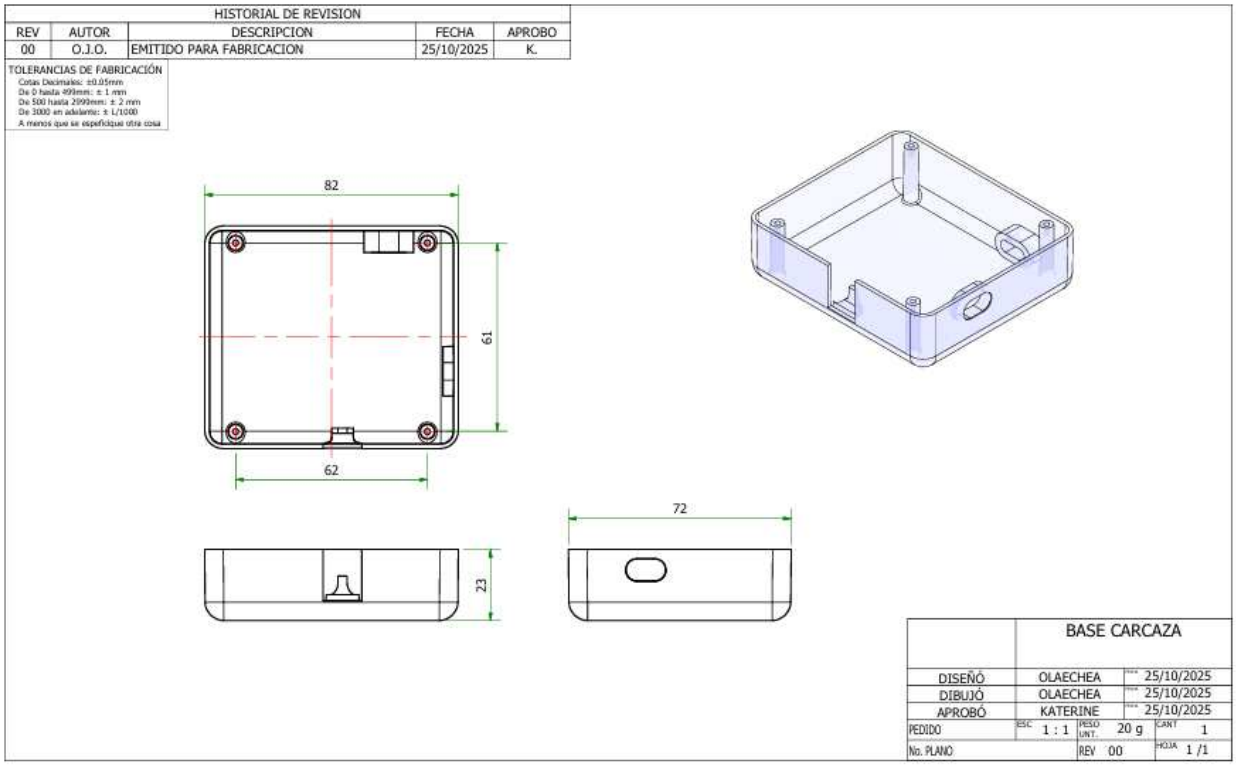
```
        // Configuración de pines
        pinMode(botonPin, INPUT_PULLUP);
        pinMode(led_encendido, OUTPUT);
        pinMode(led_alarma, OUTPUT);
        pinMode(led_bateria, OUTPUT);

        digitalWrite(led_encendido, HIGH);
        digitalWrite(led_bateria, HIGH);

        Serial.println("{\"estado\":\"Sistema iniciado
correctamente\"}");
      }

      // ====================================================
      // ====================== LOOP ========================
      // ====================================================
      void loop() {
        // ----------- Lectura GPS -----------
        while (gpsSerial.available() > 0) {
          gps.encode(gpsSerial.read());
        }

        // ----------- Lectura botón -----------
        int estadoBoton = digitalRead(botonPin);
        if (botonAnterior == HIGH && estadoBoton == LOW) {
          alarmaActiva = true;
          tiempoAlarma = millis();
        }
        botonAnterior = estadoBoton;

        if (alarmaActiva) {
          alarma_on = 0;
          alarma_off = 1;
          digitalWrite(led_alarma, HIGH);
          if (millis() - tiempoAlarma >= duracionAlarma) {
            alarmaActiva = false;
            alarma_on = 1;
            alarma_off = 0;
            digitalWrite(led_alarma, LOW);
          }
        }

        // ----------- Envío periódico JSON -----------
        unsigned long currentMillis = millis();
        if (currentMillis - previousMillis >= interval) {
          previousMillis = currentMillis;

          // Lectura INA3221
          uint16_t rawBus_u = 0, rawShunt_u = 0;
          bool okBus = read16_from(INA_ADDR, REG_BUS_CH1, rawBus_u);
          bool okShunt = read16_from(INA_ADDR, REG_SHUNT_CH1,
rawShunt_u);
```

```
          float busVoltage = 0.0f, batteryVoltage = 0.0f, pct = 0.0f;

          if (okBus && okShunt) {
            busVoltage = rawBus_u * 0.001f; // en voltios
            int16_t rawShunt_s = (int16_t)rawShunt_u;
            float shunt_mV = rawShunt_s * 0.04f;
            batteryVoltage = busVoltage + (shunt_mV / 1000.0f);
            pct = ((batteryVoltage - V_MIN) / (V_MAX - V_MIN)) *
100.0f;
            if (pct < 0.0f) pct = 0.0f;
            if (pct > 100.0f) pct = 100.0f;
          }

          // Control visual de batería baja (<20%)
          if (pct <= 20.0f) {
            digitalWrite(led_bateria, (millis() / 300) % 2); //
parpadea rápido
          } else {
            digitalWrite(led_bateria, HIGH);
          }
          // Construir JSON
          String jsonString = "{";
          jsonString += "\"alarma_off\":" + String(alarma_off) + ",";
          jsonString += "\"alarma_on\":" + String(alarma_on) + ",";
          jsonString += "\"led_encendido\":" +
String(digitalRead(led_encendido)) + ",";
          jsonString += "\"led_alarma\":" +
String(digitalRead(led_alarma)) + ",";
          jsonString += "\"led_bateria\":" +
String(digitalRead(led_bateria)) + ",";
          jsonString += "\"voltaje_bateria\":" + String(batteryVoltage,
3) + ",";
          jsonString += "\"nivel_bateria\":" + String(pct, 1);

          if (gps.location.isValid()) {
            jsonString += ",\"latitud\":" + String(gps.location.lat(),
6);
            jsonString += ",\"longitud\":" + String(gps.location.lng(),
6);
          } else {
            jsonString += ",\"gps\":\"Sin señal\"";
          }

          jsonString += "}";

          Serial.println(jsonString);
          XBeeSerial.println(jsonString);
        }
      }
```

*Nota.* Fuente: autoría propia (2025)

**Anexo H**

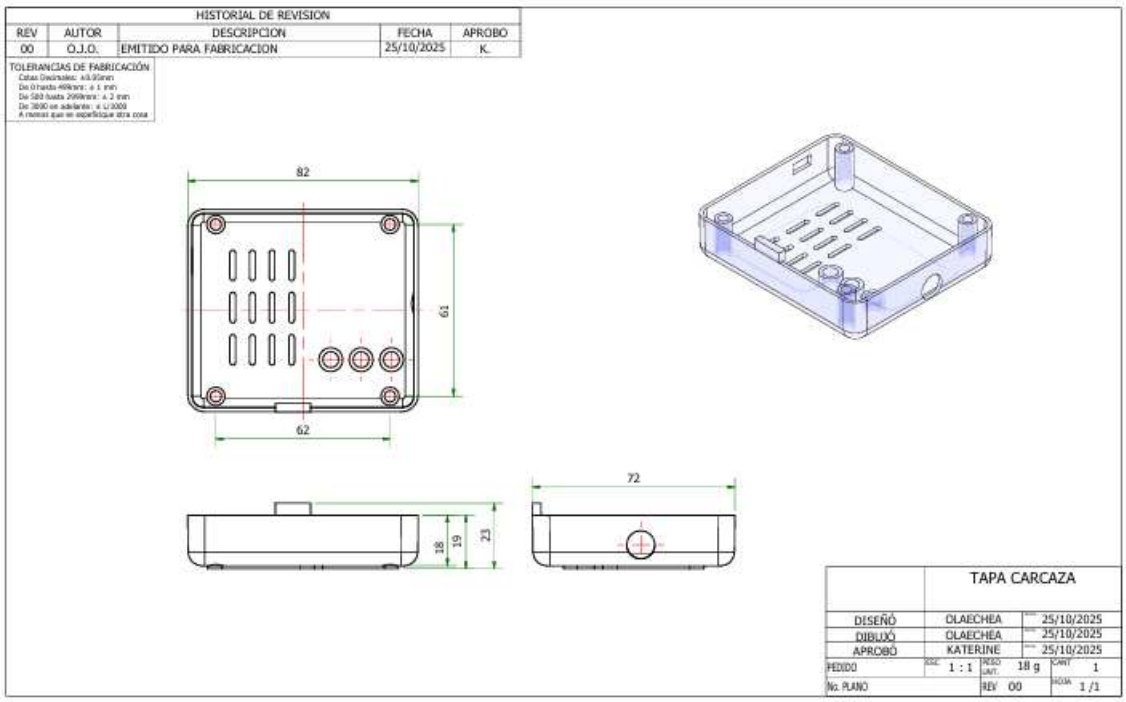
Diseño de la base de la carcasa




*Nota.* Fuente: elaborado por Olaechea, Omar (2025)

## Anexo I

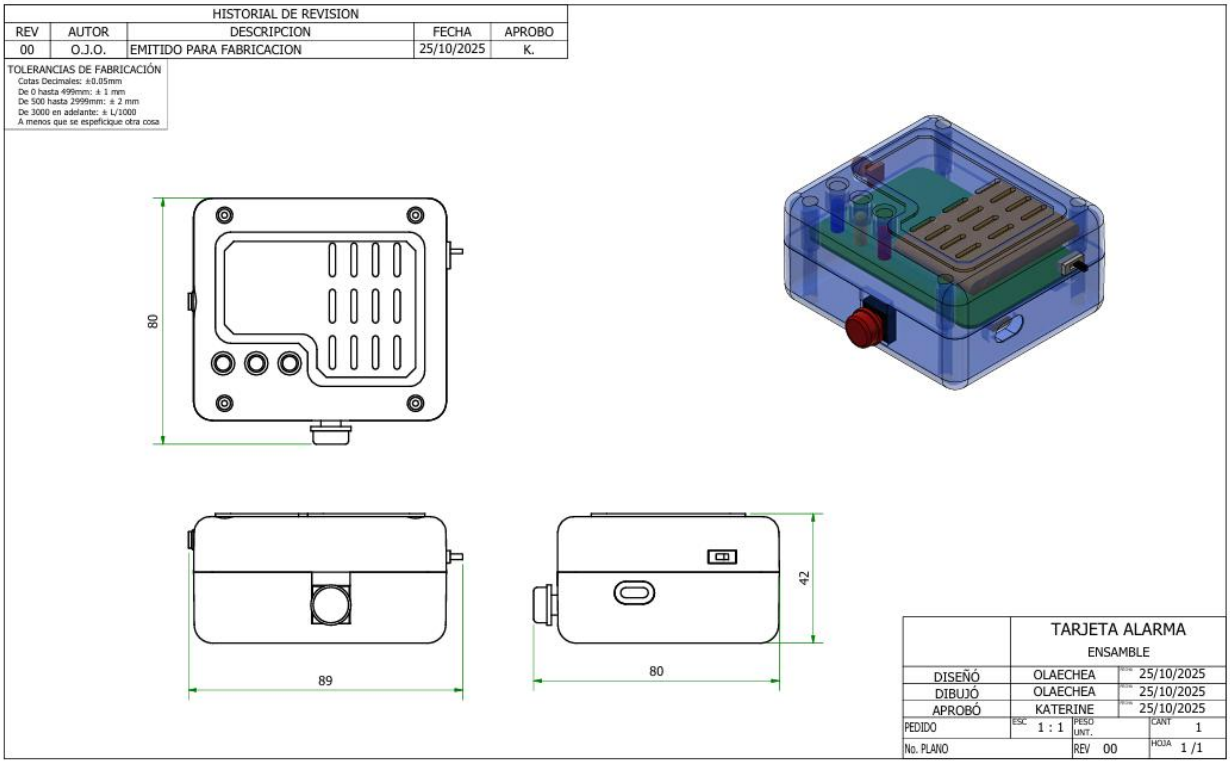
Diseño de la tapa de la carcasa




*Nota.* Fuente: elaborado por Olaechea, Omar (2025)

## Anexo J

Diseño de la tapa de la carcasa




*Nota.* Fuente: elaborado por Olaechea, Omar (2025)