UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN MECATRÓNICA

TEMA:

"ALARMA DE SEGURIDAD IoT PARA EDIFICACIONES RESIDENCIALES"

Línea de investigación:

Innovación y transferencia tecnológica/Automatización

AUTOR:

Marlon Steveen Negrete Ramirez

DIRECTOR:

PhD. Rosero Chandi Carlos Xavier

Ibarra – Ecuador 2025

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004842587		
APELLIDOS Y NOMBRES:	Negrete Ramirez Marlon Steveen		
DIRECCIÓN:	Calle Río Quinindé Y Hernan Gonzales de Saa		
EMAIL:	msnegreter@utn.edu.ec estevennegrete@gmail.com		
TELÉFONO FIJO:	-	**TELÉFONO MÓVIL:**	0998251124

DATOS DE LA OBRA	
TÍTULO:	Alarma de Seguridad IoT para Edificaciones Residenciales
AUTOR:	Marlon Steveen Negrete Ramirez
FECHA:	04/07/2025
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	■ **PREGRADO** ☐ **POSGRADO**
TITULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
DIRECTOR:	PhD. Rosero Chandi Carlos Xavier

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 4 días del mes de julio de 2025

EL AUTOR:

(f) .....
Nombre: Marlon Steveen Negrete Ramirez

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 04 de julio de 2025

PhD. Rosero Chandi Carlos Xavier

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del Trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

(f).....
PhD. Rosero Chandi Carlos Xavier
C.C.: 1002515821

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular "ALARMA DE SEGURIDAD IoT PARA EDIFICACIONES RESIDENCIALES" elaborado por MARLON STEVEEN NEGRETE RAMIREZ, previo a la obtención del título de INGENIERO EN MECATRÓNICA, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

(f).....

PhD. Rosero Chandi Carlos Xavier

C.C.: 1002515821

(f).....

MSc. Luz María Tobar Subía Contento

C.C.: 1002515821

DEDICATORIA

Con profunda gratitud, dedico este trabajo a mi familia y a todas las personas que me brindaron su apoyo, ánimo y confianza a lo largo de este proceso académico.

A mi madre, Nelva Ramirez, gracias por ser mi pilar más firme. Tus consejos, enseñanzas y, sobre todo, tu amor inquebrantable me han dado la fuerza para no rendirme incluso en los momentos más difíciles. Gracias por enseñarme con tu ejemplo que la perseverancia, la humildad y la fe son herramientas poderosas para avanzar. Este logro también es tuyo.

A mi padre, Francisco Negrete, por su respaldo silencioso, pero siempre presente, por confiar en mí y enseñarme a mirar con determinación los objetivos.

También quiero agradecer a los amigos, docentes y compañeros que, de una u otra forma, aportaron su tiempo y conocimientos en este trayecto.

A todos ustedes, gracias totales.

Marlon Steveen Negrete Ramirez

AGRADECIMIENTO

Quiero expresar mi sincero agradecimiento al PhD. Xavier Rosero por su valiosa guía, paciencia y los conocimientos compartidos a lo largo de este proceso. Además, expreso mi gratitud por las enseñanzas impartidas en las aulas, que han dejado una huella importante en mi formación académica.

Agradezco también a mis compañeros y amigos Joffre Túquerres, Edgar Villa y David Villarreal, por su compañía, apoyo y colaboración a lo largo de cada semestre. Juntos compartimos logros al superar todos los desafíos que se nos presentaban. Gracias por hacer de este camino académico una experiencia más llevadera.

Finalmente, extiendo mi agradecimiento a la MSc. Luz María Tobar Subía Contento y a todos los docentes que han formado parte de mi trayectoria universitaria. Cada uno, con sus enseñanzas y exigencias, contribuyó en la expansión de mis conocimientos y en el desarrollo de mis habilidades.

Marlon Steveen Negrete Ramirez

ÍNDICE DE CONTENIDOS

CAPÍTULO IV

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ALGORITMOS

RESUMEN

La seguridad en los espacios residenciales enfrenta desafíos constantes ante eventos que amenazan la integridad y seguridad de la propiedad privada en diferentes aspectos. En respuesta a esta problemática, se presenta un trabajo cuyo objetivo fue el desarrollo de un sistema de alarma de seguridad basado en tecnología del Internet de las Cosas (IoT), que permite la configuración, el monitoreo, detección, notificación y accionamiento de actuadores, para generar una alerta temprana ante eventos que comprometan la seguridad de las personas. De igual manera, se desarrolló una aplicación móvil para la plataforma Android, con una interfaz moderna e intuitiva que permite en sus funciones la autentificación del usuario, visualización del estado de alarma, así como la configuración y gestión del sistema. Por lo cual, se llevó a cabo la identificación de las necesidades del sistema de alarma, selección de plataformas y tecnologías disponibles, se desarrolló una arquitectura escalable basada en la nube haciendo uso servicios de autentificación, almacenamiento y mensajería, se diseñó, programo e implemento el hardware del sistema, así como el desarrollo de una aplicación móvil para la plataforma Android. Finalmente, se realizaron pruebas de conectividad y tiempos de reacción demostrando la confiabilidad, eficiencia y rendimiento del sistema de alarma.

Palabras clave: Internet de las Cosas (IoT), React Native, Raspberry Pi, Firebase.

ABSTRACT

Security in residential spaces faces constant challenges in the face of events that threaten the integrity and safety of private property in different aspects. In response to this problem, we present a work whose objective was the development of a security alarm system based on Internet of Things (IoT) technology, which allows the configuration, monitoring, detection, notification and actuation of actuators, to generate an early warning of events that compromise the safety of people. Similarly, a mobile application was developed for the Android platform, with a modern and intuitive interface that allows user authentication, visualisation of the alarm status, as well as configuration and management of the system. Therefore, we identified the needs of the alarm system, selected the available platforms and technologies, developed a scalable cloud-based architecture using authentication, storage and messaging services, designed, programmed and implemented the system hardware, as well as developed a mobile application for the Android platform. Finally, connectivity and reaction time tests were conducted to demonstrate the reliability, efficiency and performance of the alarm system.

Keywords: Internet of Things (IoT), React Native, Raspberry Pi, Firebase.

CAPÍTULO I

INTRODUCCIÓN

1.1 Problema

En el contexto de Ecuador, se observa un aumento significativo en la tasa de delitos como robos, asaltos y crimen organizado como se muestra en la Fig.1.1. [1].

	País	Tasa 2021	Tasa 2022	Variación (%)
1	Ecuador	13,7	25,0	82,5%
2	Trinidad y Tobago	32,0	43,2	35,0%
3	Haití	13,7	16,7	21,9%
4	Nicaragua	5,7	6,9	21,1%
5	Chile	3,6	4,3	19,4%
6	Perú	4,3	5,0	16,3%
7	Uruguay	8,5	9,4	10,6%
8	Costa Rica	11,5	12,6	9,6%
9	Guatemala	16,6	17,3	4,2%
10	Jamaica	49,4	50,6	2,4%

Fig. 1.1. Países con más crecimiento de violencia [1].

La situación que afronta el país ha generado una demanda creciente de sistemas de seguridad capaces de evitar ser víctimas de robos a negocios, fábricas y empresas, los dispositivos que más se han implementado se muestran en la Fig. 1.2.

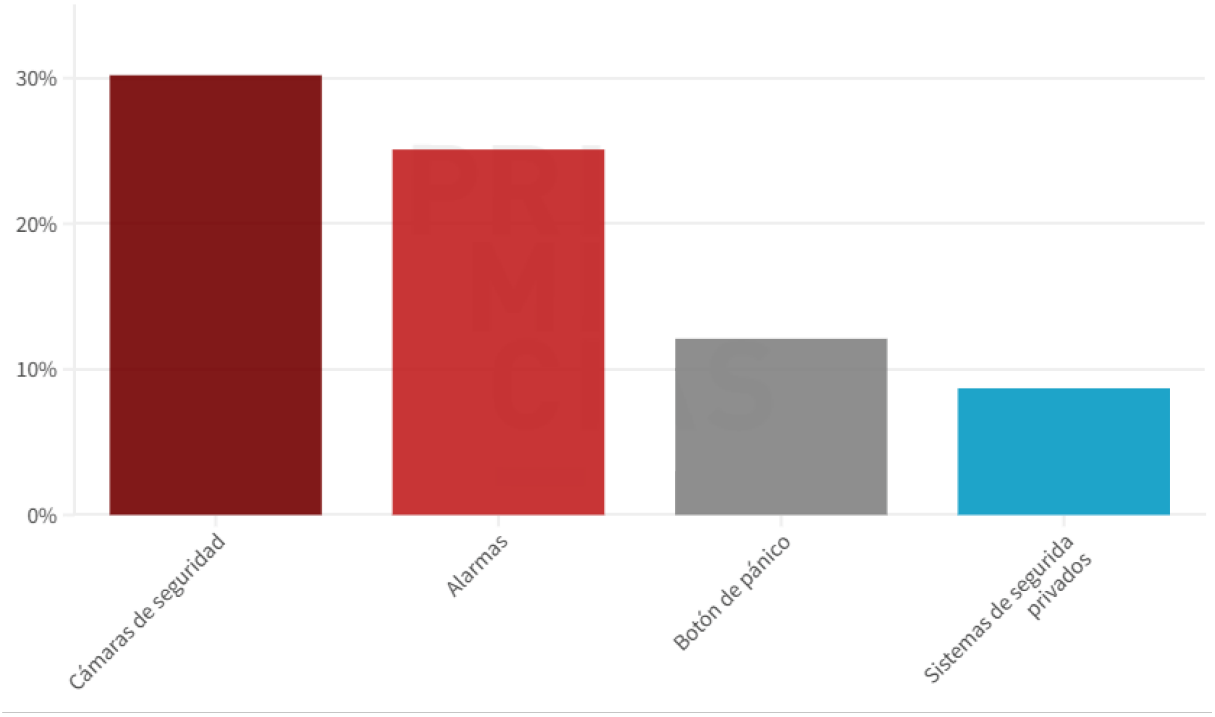


Fig. 1.2. Dispositivos instalados en negocio. Quito - Julio 2023. [2]

La Fig. 1.3. muestra cómo, en el primer semestre del 2023 el Servicio de Rentas Internas (SRI) registró ventas en actividades de seguridad de $590 millones, simbolizando un incremento del 15 % con respecto al año 2022 [2].

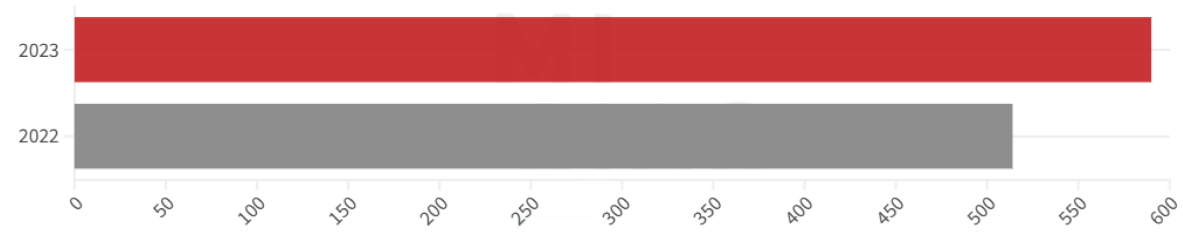


Fig. 1.3. Registro del SRI en actividades de seguridad. [2]

Los precios de los sistemas de alarmas disponibles en el país varían significativamente en su valor, oscilando en un rango que abarca desde $250 en marcas genéricas hasta $550 en marcas reconocidas. Sin embargo, se pueden considerar costos adicionales relacionados con instalación, ampliación, servicio técnico y monitoreo [3].

El Internet de las Cosas (IoT) representa un cambio en la calidad de vida de las personas

al brindar oportunidades en el ámbito de la seguridad. Esto se logra mediante sensores para la adquisición de información del entorno, lo que permite tomar de acciones, como la activación de alarmas para salvaguardar a los ocupantes de un edificio. Además, la reducción de los costos del ancho de banda y del procesamiento ha hecho posible que más dispositivos estén interconectados y sean lo suficientemente inteligentes para gestionar los crecientes volúmenes de datos generados o recibidos [4].

La creciente necesidad de fortalecer la seguridad en edificaciones ha ratificado la importancia de implementar sistemas de alarmas. Sin embargo, la falta de conocimiento en las nuevas tecnologías disponibles, opciones accesibles y los costos asociados limitan la implementación de estos. Por lo tanto, se propone el desarrollo de una alarma de seguridad IoT que se adapte a las necesidades locales, con el objetivo de mejorar la seguridad de edificaciones residenciales.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un sistema de alarma de seguridad basado en tecnología IoT para edificaciones residenciales de diferentes tamaños y características.

1.2.2 Objetivos Específicos

- Determinar los parámetros y características principales de una alarma de seguridad centrándose en la precisión de la detección de amenazas y la gestión eficiente de la información.
- Diseñar el sistema de seguridad en base a parámetros determinados utilizando componentes disponibles en el mercado.
- Proponer una arquitectura basada en la nube que permita a los usuarios la configuración y gestión a través de internet.

- Validar el rendimiento y la confiabilidad de la alarma en diferentes escenarios de trabajo.

1.3 Alcance

El proyecto tiene como objetivo el diseño, construcción y pruebas de funcionamiento de un prototipo de Alarma de seguridad IoT que contenga diferentes características adaptándose a distintos tipos de edificaciones residenciales. El prototipo se construirá utilizando componentes disponibles en el mercado, con una unidad central de procesamiento, diferentes tipos de sensores y funciones de comunicación. Los datos recopilados se transmitirán a una plataforma en la nube, y los resultados podrán ser visualizados a través de una aplicación móvil. Para acceder los usuarios deberán autenticarse con las credenciales adecuadas, lo que les permitirá visualizar los datos de manera comprensible y accesible.

1.4 Justificación

En el ámbito de tecnología, el Internet de las Cosas (IoT) evoluciona continuamente la forma en que se abordan los desafíos de seguridad debido a la incorporación de nuevas tecnologías, que permiten fortalecer los sistemas, permitiendo una gestión más eficiente y escalable de seguridad demostrando nuevas aplicaciones en que se abordan los riesgos y emergencias.

En el ámbito de seguridad, el funcionamiento de un sistema de alarma IoT en una edificación permite elevar la percepción de seguridad de los usuarios al implementar dispositivos de comunicación y sensores, que permita detectar las condiciones del entorno y a su vez tomar acciones en caso de una amenaza en la integridad y seguridad de la propiedad privada.

En el ámbito económico, se estaría proporcionando un dispositivo accesible, lo que permitirá su adopción en una amplia variedad de contextos, incluidos aquellos con recursos limitados. La accesibilidad en términos de costos es esencial para garantizar que la seguridad y la protección estén al alcance de un público amplio.

Finalmente, este proyecto tendrá un impacto social al fortalecer la seguridad y protección en edificaciones residenciales, contribuyendo de esta manera al bienestar de las personas y mejorando la calidad de vida.

CAPÍTULO II

MARCO TEÓRICO

2.1 Estado del arte

En [5] se diseña e implementa un sistema de control de incendios que utiliza la tecnología Field Programmable Gate Array (FPGA) de National Instruments. Se desarrolla un sistema versátil con entradas y salidas analógicas, conexiones a la alarma con sus activadores y un sistema de mensajes de texto. Como resultado, se obtiene una alarma para la detección de incendios monitoreada en tiempo real junto con una interfaz de usuario en LABVIEW que muestra el estado de la alarma.

En [6] se implementa un sistema de alerta que se basa en el Internet de las cosas (IoT) y utiliza tecnologías de código abierto. Este sistema emplea sensores de movimiento, temperatura y humo para identificar diferentes eventos. Para lograr esta capacidad de detección, se utilizó una Raspberry Pi como un dispositivo puente (gateway), que facilita la conexión entre los sensores y la nube. De esta manera, se obtiene un sistema de alarma IoT escalable y se adapta a diversas necesidades.

En [7] se desarrolla un sistema de alerta IoT destinado a la detección de robos en entornos residenciales. Este sistema consta de sensores de contacto, un módulo NodeMCU ESP8266 y una conexión WiFi de alta velocidad. Se realiza una plataforma en la nube para analizar los datos que se obtienen y, en caso necesario, activar una alarma. La eficiencia que se obtiene de este sistema es del 97.4 %, tanto en la activación como en la desactivación de la alarma.

En [8] se diseña e implementa un sistema de alarma de intrusión basado en el protocolo ESP Now de Internet de las Cosas. Se opta por utilizar el protocolo ESP-Now debido a su eficiencia

en aplicaciones de IoT con bajo consumo de energía, y se integra con una red de nodos sensores independientes. Como resultado, se logra desarrollar un prototipo que cumple con los estándares predefinidos en términos de funcionamiento y costos.

En [9] se diseña e implementa un prototipo para el monitoreo y purificación del aire en espacios interiores. El sistema utiliza una pantalla HMI Nextion y sensores electroquímicos capaces de detectar gases contaminantes mediante variaciones de resistencia interna, con opción de calibración para mejorar la precisión. El monitoreo se puede realizar localmente o de forma remota a través de la plataforma Ubidots IoT, donde se almacenan y analizan los datos. El prototipo incluye filtros HEPA, de polvo, de carbón activado y una turbina que se activa según las concentraciones detectadas. Se mide la presencia de gases $CO_2, NO_2, NH_3, CO, O_3, PM_{10}$ y $PM_{2,5}$, además de temperatura y humedad.

En [10] se diseña e implementa un prototipo de monitoreo para medir y visualizar en tiempo real la calidad del aire en el edificio de la Carrera de Ingeniería en Networking y Telecomunicaciones. El sistema se basa en tecnología IoT contemplando un diseño escalable en su infraestructura, mientras hace uso de sensores especializados para medir concentraciones de contaminantes como dióxido de azufre, dióxido de nitrógeno, monóxido de carbono y ozono. Los datos se envían a una plataforma que permite la visualización de manera sencilla para el monitoreo en tiempo real. Los resultados demuestran que el sistema desarrollado es fiable y escalable a distintos entornos.

En [11] se diseña un prototipo de sistema de seguridad que utiliza sensores de movimiento y cámaras IP para videovigilancia, todo dentro de una infraestructura IoT. Este sistema implementa el protocolo MQTT para facilitar la comunicación y el intercambio de datos entre los dispositivos. Detecta intrusos a través de los sensores de movimiento y graba eventos de manera continua. Si se detecta alguna actividad sospechosa, se envía una alerta al usuario mediante una aplicación móvil, permitiendo su visualización. Los resultados demuestran la fiabilidad y escalabilidad del sistema.

En [12] se implementa un sistema de seguridad con IoT para un local comercial, basándose

en un botón de pánico y una alarma con módulo GPS. El proyecto se desarrolla para mejorar la seguridad del local debido al aumento de la delincuencia en la provincia de Santa Elena. Utiliza una arquitectura IoT que incluye un microcontrolador Arduino, un módulo Shield GPRS, y una aplicación móvil desarrollada para dispositivos Android. La aplicación implementada permite de forma sencilla a los usuarios activar la alarma y enviar una señal de alerta a la policía más cercana. El sistema se diseña para ser controlado de manera remota, proporcionando una solución efectiva y accesible para la seguridad del establecimiento.

En [13] se realiza un prototipo de sistema de seguridad doméstico basado en la tecnología WPAN (Wireless Personal Area Network) y la red IoT. El sistema cuenta con cinco nodos, el nodo sensor, el nodo actuador, el nodo coordinador, el nodo de conexión a internet. Las funciones del sistema son detectar la presencia de intrusos o nivel de gas fuera de los límites establecidos, envió de datos, análisis de datos y envío de datos a internet. Para el prototipo se implementa sensores de movimiento y de gas, placas Arduino, software de comunicación inalámbrica Zigbee y una base de datos en MySQL. Las pruebas muestran una precisión del 95 % en detecciones intrusiones y niveles de gas.

En [14] se realiza un prototipo de sistema de seguridad para control domótico basado en tecnología IoT. En la parte de hardware se usá sensores de movimiento, sensores de apertura de puertas y ventanas, actuadores para controlar la apertura de puertas y ventanas; en la parte de software se usá un controlador de hardware desarrollado en Python, una aplicación móvil usando Kotlin y una base de datos en Firebase. El sistema detecta la presencia de intrusos o la apertura no autorizada de puertas y ventanas; además, transmite, almacena y analiza los datos para ejecutar acciones específicas. A través de las pruebas realizadas, se concluye que el prototipo cumple satisfactoriamente con los requerimientos establecidos, centrados en el confort, la calidad de vida y el incremento de la seguridad.

2.2 Tecnologías implícitas en la solución

Esta sección describe los principales componentes tecnológicos que intervienen en la solución propuesta, incluyendo los protocolos de comunicación utilizados para el intercambio de datos entre los sensores y la unidad central, las tecnologías inalámbricas que permiten la conectividad entre dispositivos, y la base de datos encargada del almacenamiento y gestión de la información en tiempo real. Además, se detallan las herramientas y lenguajes empleados en el desarrollo del software.

2.2.1 Protocolos de comunicación

2.2.1.1 Protocolo MQTT

El protocolo de mensajería MQTT es un sistema de mensajería ligero que se basa en un modelo publicación/suscripción. Es ideal por su amplia implementación en proyectos IoT, debido a que minimiza el consumo de recursos del dispositivo y ancho de banda [15]. Entre sus características principales son el uso de TCP, Calidad de Servicio (QoS, Quality of Service) y funciones como Ultimo Testamento (LWT,Last Will and Testament), permiten garantizar una comunicación estable entre dispositivos e identificar desconexiones de manera oportuna.

2.2.1.2 HTTP (HyperText Transfer Protocol)

El protocolo HTTP es un estándar de comunicación que permite la transferencia de información como texto, gráficos, sonido, video y otros archivos multimedia. Este protocolo, basado en un modelo de solicitudes y respuestas, es adecuado para interacciones puntuales o aplicaciones donde la transferencia de datos no es frecuente y la latencia no es un factor crítico [16].

De esta manera se utiliza para enviar parámetros clave, como credenciales de red y configuraciones específicas, permitiendo la integración de dispositivos en sistemas más amplios. Este

enfoque resulta eficiente para configurar conexiones iniciales.

2.2.2 Tecnologías inalámbricas IoT

2.2.2.1 Wi-Fi

El estándar IEEE 802.11, conocido como WiFi, es un protocolo de comunicación inalámbrica que opera en 2.4 GHz, siendo utilizado para aplicaciones de adquisición de datos, medición, control de procesos e intercambio de contenido multimedia [17]. El funcionamiento de esta red puede ser de dos tipos:

- Ad-hoc

  La red *ad-hoc* se emplea de forma ocasional, ya que admite solo un número reducido de usuarios y facilita la comunicación entre ellos sin necesidad de jerarquías. Esta configuración tiene un uso limitado, principalmente en entornos cerrados sin acceso a redes externas, donde su propósito principal es permitir un intercambio eficiente de información entre dispositivos. La transmisión de datos es efectiva una vez que se establece la conexión entre el emisor y el receptor [17].

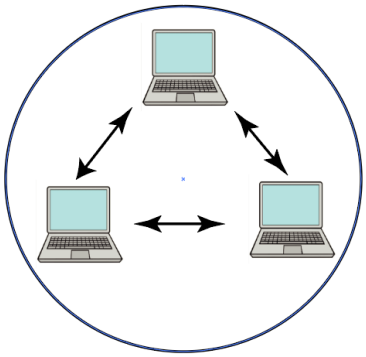


Fig. 2.1. Red Ad-Hoc.

- Infraestructura

  La red de *infraestructura* es la más comúnmente utilizada, ya que una única estación,

conocida como punto de acceso, se encarga de gestionar el acceso y centralizar la información transmitida. Esta red se organiza en una jerarquía de dos niveles: por un lado, los dispositivos que conforman la infraestructura base, y por otro, los terminales de los clientes que forman la red propiamente dicha [17].

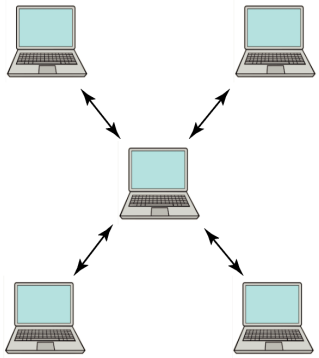


Fig. 2.2. Red infraestructura

2.3 Base de datos

2.3.1 Firebase Realtime Database

Firebase Realtime Database proporciona una solución de almacenamiento en la nube que gestiona y sincroniza datos en tiempo real. Estructura los datos en formato JSON, lo que permite mantenerlos actualizados de forma inmediata en todos los clientes conectados. Este enfoque ofrece una arquitectura escalable y segura, diseñada para manejar altos volúmenes de información con latencia mínima [18]. Además que permite la integración con otros servicios de firebase como Firebase Authentication y Firebase Cloud Messaging.

- Firebase Authentication

  El servicio de *Firebase Authentication* ofrece la autenticación de usuarios mediante contraseñas, correos electrónicos y otros métodos. Facilita su integración con servicios de backend y SDK, proporcionando una solución integral de seguridad y autenticación [19].

- Firebase Cloud Messaging

  El servicio de *Firebase Cloud Messaging* permite el envío de mensajes multiplataforma, proporcionando una solución eficiente para notificar al cliente con mensajes personalizados. Este servicio es compatible con dispositivos móviles, navegadores web y aplicaciones backend, lo que lo convierte en una herramienta versátil para gestionar notificaciones en tiempo real. [20].

2.4 Desarrollo de software

2.4.1 Python

Python es un lenguaje de programación de alto nivel, versátil y ampliamente utilizado en el desarrollo de soluciones IoT debido a si amplia documentación, simplicidad y disponibilidad de bibliotecas [21]. De esta manera, permite desarrollar la comunicación entre dispositivos, procesar datos y ejecutar acciones automatizadas, ofreciendo un entorno eficiente, escalable y versátil adaptándose a las necesidades de cada proyecto.

2.4.2 React Native CLI

React Native es un framework de código abierto mediante el cual se puede crear aplicaciones móviles nativas utilizando JavaScript, TypeScript y React. Mediante un enfoque basado en componentes, facilita la creación de interfaces modernas, dinámicas y responsivas tanto para iOS y Andriod con una sola base de código. De igual manera, permite emplear funciones nativas de dispositivo móvil manteniendo un alto rendimiento. Además, cuenta con una amplia documentación, siendo una opción ideal para el desarrollo de aplicaciones móviles rápidas y escalables [22]. Sin embargo, React Native también permite la adopción de TypeScript para el desarrollo.

- TypeScript

  TypeScript se caracteriza por agregar un tipado estricto y otras características a JavaScript, permitiendo la detección temprana de errores, facilitado la depuración del código antes de su implementación, evitando de esta manera afectar a la escalabilidad del proyecto a medida que este se expande y mejorando el ambiente de desarrollo [23].

2.5 Análisis del estado del arte

En esta revisión se demuestra que los sistemas de alarmas compartes similitudes en características tanto en variables de monitores como en la gestión de diversos parámetros. Todas estas funciones son implementadas a través de diferentes tipos de sensores como infrarrojos, ultrasonido, microondas, fotoeléctricos, entre otros. Por lo que de esta manera se resalta la diversidad y complejidad de los dispositivos de seguridad.

La Tabla 2.1 resume los principales trabajos relacionados en el estado del arte, donde se evidencio el uso de diferentes tecnologías y metodologías en el diseño e implementación de sistemas de alarma. Como tendencia se observa la aplicación de plataformas como Arduino, ESP32, Raspberry Pi, las cuales cuenta con una amplia documentación en aplicaciones de diversos proyectos. Además, se evidencia la búsqueda de flexibilidad y personalización para la elección de estas plataformas, con el fin de tener un entorno adecuado y cumplir con los requisitos para cada proyecto.

Por otra parte, se evidencia la importancia de los protocolos de comunicación para la gestión eficiente de los datos, donde protocolos como MQTT, CoAP y HTTP son ampliamente utilizados, por lo cual la elección de cada uno dependerá de los requisitos específicos y restricciones de cada proyecto, donde cada protocolo poseerá ventajas y desventajas que deberán ser evaluadas antes de su implementación.

Tabla 2.1

Resumen de trabajos relacionados en el estado del arte

Características y Resultados Principales	**Tecnología Utilizada**	**Referencia**
Sistema de control de incendios con monitoreo en tiempo real y notificaciones vía SMS.	FPGA, LABVIEW	[5]
Alerta IoT escalable con detección de humo, temperatura y movimiento.	Raspberry Pi, Sensores IoT	[6]
Sistema de detección de robos con eficiencia del 97.4 % en activación de alarmas.	NodeMCU ESP8266, WiFi	[7]
Alarma de intrusión eficiente y de bajo consumo energético.	Protocolo ESP-Now, Nodos Sensores	[8]
Sistema para detectar incendios y gases inflamables en cocinas, con alertas móviles confiables.	Sensores de gases, IoT	**[9]bajana2020**
Infraestructura escalable para monitorear contaminantes en tiempo real.	Sensores de Calidad del Aire, IoT	[10]
Sistema de seguridad IoT con detección de intrusos y notificaciones móviles.	MQTT, Cámaras IP	[11]
Sistema de seguridad para locales comerciales con botón de pánico y monitoreo remoto.	Arduino, Shield GPRS, GPS	[12]
Sistema doméstico IoT con detección de gas y presencia, logrando precisión del 95 %.	WPAN, Zigbee, MySQL	[13]
Control domótico para detectar intrusiones y gestionar aperturas de puertas y ventanas.	Sensores IoT, Firebase	[14]

CAPÍTULO III

MARCO METODOLÓGICO

3.1 Enfoque y tipos de investigación

La investigación es de tipo aplicada ya que se centra en una aplicación directa en la resolución de un problema basándose en la teoría aprendida [24]. En este caso específico se propone una alternativa de un sistema de alarma de seguridad IoT para edificaciones residenciales.

La investigación también adopta un enfoque documental, ya que requiere la recolección, recopilación y selección de información proveniente de diversas fuentes, como documentos, revistas, libros y artículos [25]. Este proceso permite establecer los parámetros y características fundamentales de un sistema de alarma.

De igual modo, la investigación adopta un enfoque descriptivo, ya que se detallan las características y propiedades de sensores, dispositivos de hardware, protocolos de comunicación y tecnologías inalámbricas [26].

Finalmente, la investigación adopta un enfoque experimental, ya que requiere la realización de pruebas de funcionamiento del prototipo para su validación, mediante la introducción de variables de estudio y la observación de sus efectos en condiciones controladas [27].

3.2 Diseño de la investigación

Considerando lo anteriormente mencionado, se plantea la ejecución de diversas actividades orientadas al cumplimiento de los objetivos específicos y del objetivo general. Estas actividades se organizan de manera secuencial con el propósito de desarrollar de forma eficiente un sistema

de alarma basado en el Internet de las Cosas (IoT).

3.2.1 Fase 1: Determinación de los parámetros y características principales de una alarma de seguridad centrándose en la precisión de la detección de amenazas y la gestión eficiente de la información.

- Actividad 1.1 Investigación de los parámetros y características para alarmas de seguridad.
  En esta actividad, se realiza una investigación focalizada en los parámetros y características esenciales para el desarrollo de sistemas de alarmas de seguridad.

- Actividad 1.2: Definición de parámetros y características principales del sistema de alarma de seguridad.
  En esta actividad, se define los parámetros y características principales del sistema de alarma, evaluando aspectos como el tipo de detección de amenazas, acciones a realizarse durante el monitoreo y la activación del sistema, la integración diferentes tecnologías y la integración con servicios en la nube.

3.2.2 Fase 2: Diseño del sistema de seguridad en base a parámetros determinados utilizando componentes disponibles en el mercado.

- Actividad 2.1 Identificación de sensores y dispositivos necesarios.
  En esta actividad, se selecciona los sensores adecuados para la detección de eventos en el sistema de alarma, incluyendo diferentes tipos de sensores. De igual manera, se elige las plataformas de procesamiento en función de sus capacidades y compatibilidad.

- Actividad 2.2: Evaluación de protocolos de comunicación para la transmisión de datos.
  En esta actividad, se analiza diferentes protocolos de comunicación utilizados en sistemas IoT, con el fin de determinar aspectos relevantes como limitaciones, características, configuración y eficiencia en la transmisión de datos.

- Actividad 2.3 Adquisión de materiales

  En esta actividad, se realiza la adquisición de los materiales necesarios para su implementación, asegurando que sean los más adecuados en cuanto a calidad, compatibilidad y costo.

- Actividad 2.4 Construcción y ensamblaje de componentes.

  En esta actividad, se lleva acabo la construcción y ensamblaje de los componentes seleccionados. Se procede a la implementación física de sensores y otros dispositivos.

3.3 Fase 3: Propuesta la arquitectura basada en la nube que permita a los usuarios la configuración y gestión a través de internet.

- Actividad 3.1 Definición de los requisitos para la gestión remota de la alarma.

  En esta actividad, se define los requisitos para la gestión remota del sistema de alarma, donde se establece parámetros y estrategias que permitan la construcción de la base de datos, el cambio de modos de alarma, las acciones en caso de activación, el procesamiento de notificaciones y el acceso a los registros del historial de eventos.

- Actividad 3.2 Diseño de una interfaz de usuario intuitiva y segura.

  En esta actividad, se diseña la interfaz de usuario que permita ser intuitiva y segura para su fácil comprensión y manejo para la interacción con el usuario, cuidando aspectos visuales, una estructura funcional e implementando medidas de seguridad para el correcto intercambio de información.

- Actividad 3.3 Implementación de la interfaz de usuario

  En esta actividad, se implementa la interfaz de usuario diseñada. Se despliega la aplicación en el sistema operativo móvil donde verificara la fluidez de la aplicación, la operatividad y la usabilidad.

3.4 Fase 4: Validación del rendimiento y la confiabilidad de la alarma en diferentes escenarios de trabajo.

- Actividad 4.1 Ensayos de funcionamiento

  En esta actividad, se realizan ensayos de funcionamiento, así como las últimas modificaciones, verificando el correcto desempeño del sistema, incluyendo la activación de alarmas, la respuesta a eventos específicos y la validación de la interfaz de usuario.

- Actividad 4.2 Ejecución y registro de los resultados.

  En esta etapa, se ejecutan pruebas de funcionamiento del sistema de seguridad. Se registran y documentan los resultados.

- Actividad 4.3 Presentación del documento final escrito.

  Esta actividad hace referencia al desarrollo de todo el documento del Trabajo de Integración Curricular, tanto del Capítulo 1, 2, 3 y 4, así como las secciones complementarias.

CAPÍTULO IV

RESULTADOS Y ANÁLISIS

4.1 Especificaciones del sistema a diseñar

A continuación, se describen los criterios y requerimientos que se tendrán en cuenta para el diseño del sistema de alarma.

Criterios

1. **Funcionalidad:**
    - Detección: El sistema debe identificar y reaccionar ante eventos que comprometa la seguridad del espacio residencial, como el movimiento, la apertura de puertas o la ruptura de ventanas.
    - Notificaciones: El usuario debe recibir alertas que se visualicen mediante la aplicación móvil, manteniéndolo informado de cualquier evento.
    - Control remoto: El usuario debe contar con una alternativa para la desactivación del sistema de alarma, en caso de no tener acceso a la aplicación móvil.
2. **Conectividad Wi-Fi:** El sistema debe estar en un entorno con acceso a internet, para su correcto funcionamiento.
3. **Aplicación móvil:** La aplicación debe ser fácil de usar en un entorno amigable y permitir mostrar información clara para el usuario.
4. **Instalación sencilla:** El sistema tiene que ser sencillo de instalar y configurar por el usuario, de manera que no necesite tener conocimientos técnicos avanzados.

5. **Fácil expansión:** El sistema debe permitir la fácil incorporación de nuevos sensores, facilitando la expansión del sistema.

6. **Diferentes modos de configuración:** El sistema debe contar con distintos modos de configuración, permitiendo adaptarse a las necesidades del usuario.

Restricciones

1. **Base de datos y servicios de Firebase con plan gratuito:** El uso de un plan gratuito en Firebase implica ciertas restricciones en el uso de los servicios, por lo que se debe considerar optimizar el uso de estos servicios para evitar un impacto tanto en el rendimiento como en la escalabilidad del sistema.

2. **Capacidad del hardware para análisis de múltiples sensores:** Las limitaciones del hardware influirán directamente en el desempeño del sistema, por lo que se debe considerar la capacidad de este para el manejo simultaneo de la información de los módulos sensores.

3. **Dependencia de la conexión a internet:** Tanto el sistema de alarma como la aplicación móvil dependerán de una conexión estable a internet para su correcto funcionamiento, por lo que se debe ubicar el sistema de alarma en un espació con acceso a internet.

4.2 Solución propuesta

4.2.1 Descripción de solución propuesta

En la propuesta de solución, el sistema de alarma IoT se basa en una red de infraestructura, como se muestra en la Fig. 4.1 donde se opera a través de una central de procesamiento que gestiona la información recibida de sensores inalámbricos, como detectores de movimiento, vibración, apertura y humo. Los datos se almacenan en Firebase Realtime Database para su consulta en tiempo real.

La interacción con el usuario se da mediante una aplicación móvil donde ingresa sus credenciales de usuario para realizar la autentificación, posteriormente accede a las funciones como configurar su alarma, agregar sensores, gestionar modos de alarma, accionamiento del botón de pánico. Ante un evento el sistema activa la sirena y envía una notificación al usuario, el cual accederá a la aplicación par la desactivación de la alarma o en su defecto mediante un control infrarrojo.

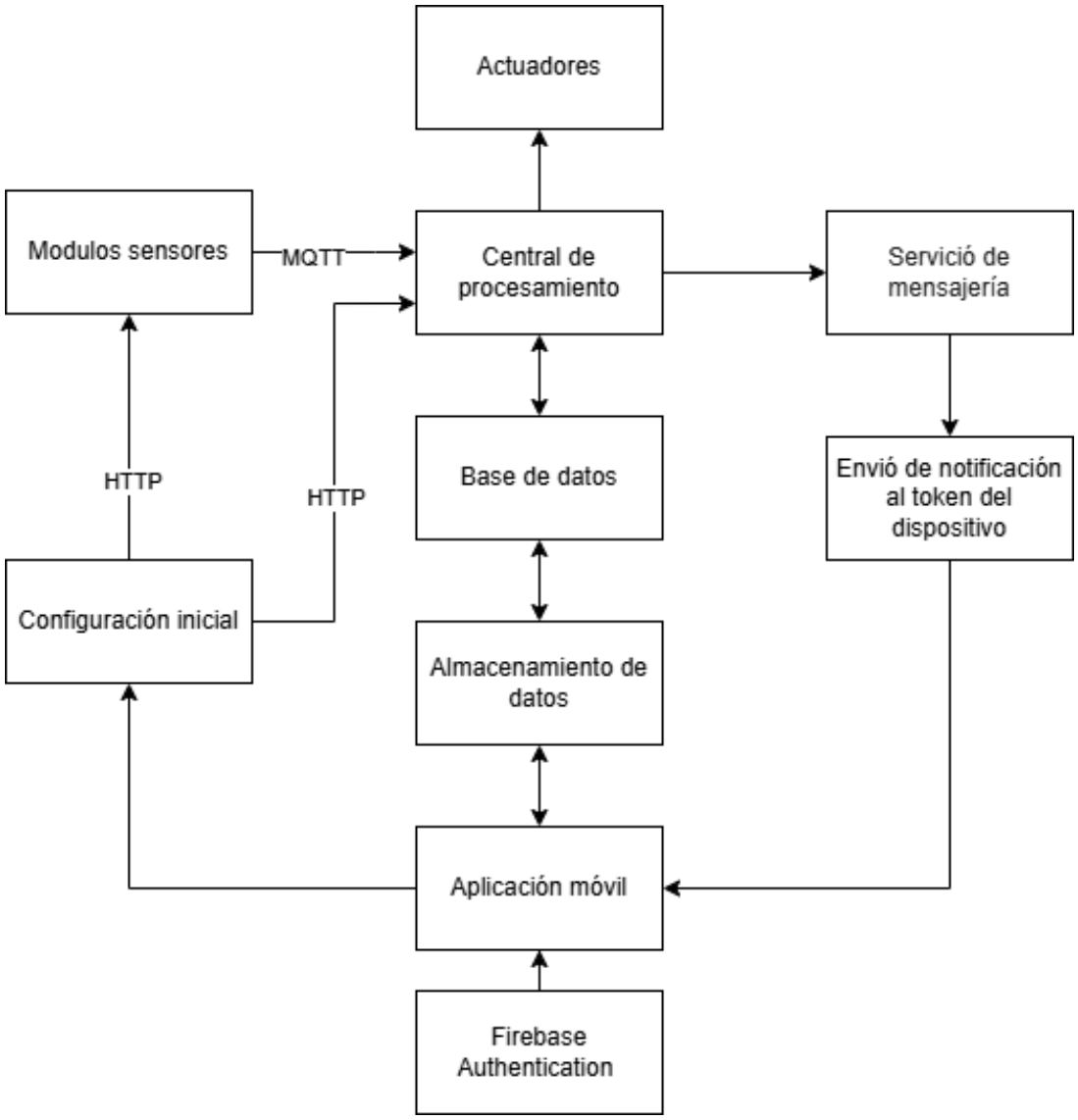


Fig. 4.1. Arquitectura del sistema.

4.2.2 Central de procesamiento

La central de procesamiento está conectada a la red WiFi y se encarga de gestionar la comunicación entre los sensores mediante MQTT y la base de datos. Para ello, ejecuta un broker Mosquitto MQTT, que facilita el intercambio de datos con los sensores. A través de la base de datos, publica y recibe información que permite configurar los parámetros de la alarma, incluyendo la adición de nuevos sensores, la activación o desactivación del sistema y la supervisión del estado de cada sensor agregado. En la Fig. 4.2 se visualiza los elementos que conforman la central la central de procesamiento, así como la lista de partes en la Tabla. 4.1.

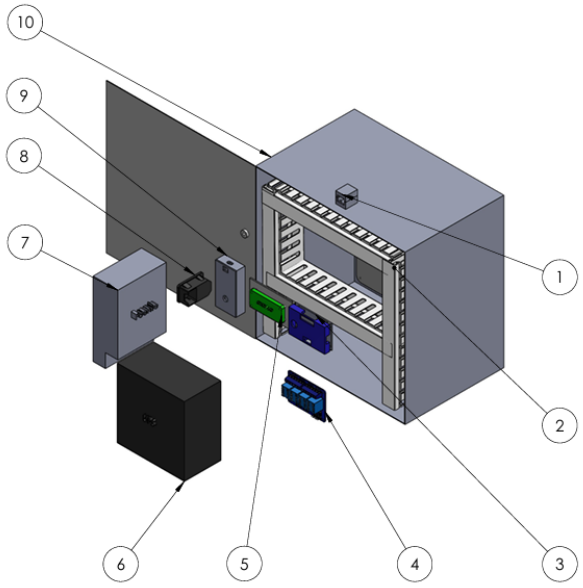


Fig. 4.2. Central de procesamiento.

Tabla 4.1
Lista de elementos central de procesamiento

Elemento	Descripción	Cantidad
1	Sensor receptor infrarrojo ky-022	1
2	Canaletas ranuradas	4
3	Módulo con pulsador de reinició y apagado	1
4	Módulo relé	1
5	Raspberry Pi Zero 2 W	1
6	Batería de 12V	1
7	Fuente de poder	1
8	Switch interruptor con fusible	1
9	Modulo regulador de voltaje Step Down LM2596	1
10	Gabinete metálico 30x30	1

4.2.3 Módulo de proximidad

El módulo de proximidad, basado en la Raspberry Pi Pico W, permite detectar intrusiones dentro de un área específica. Al detectar movimiento o la presencia de un objeto en el espacio monitorizado, el módulo envía un mensaje por MQTT en su tópico correspondiente hacia la central de procesamiento. Este mensaje activa la alarma según la configuración establecida, se generan alertas para el usuario a través de la aplicación móvil y activa los actuadores. En la Fig. 4.3 se visualizan los elementos que conforman el módulo de proximidad, así como la lista de partes en la Tabla 4.2.

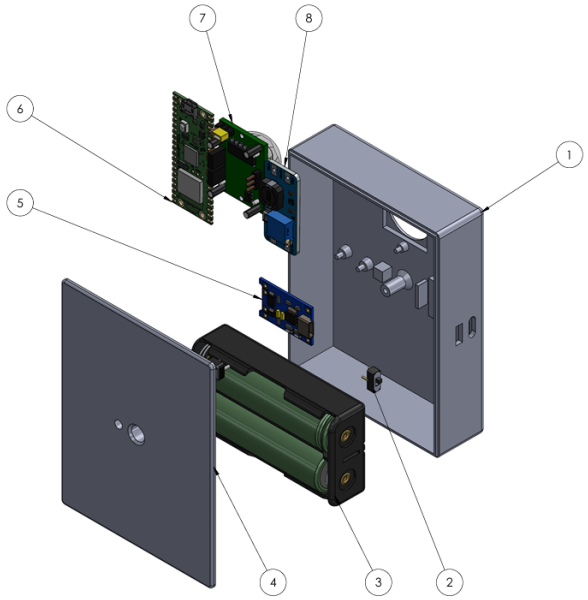


Fig. 4.3. Módulo de proximidad.

Tabla 4.2
Lista de elementos módulo de proximidad

Elemento	Descripción	Cantidad	Material
1	Base	1	PLA
2	Switch deslizante	1	-
3	Portapilas con baterías 18650	1	-
4	Tapa	1	PLA
5	Módulo de protección de carga TP4056	1	-
6	Raspberry Pico W	1	-
7	Sensor PIR HC-SR501	1	-
8	Elevador de voltaje MT3608	1	-

4.2.4 Módulo de apertura

El módulo de apertura, basado en la Raspberry Pi Pico W, supervisa puertas, ventanas u otros puntos de acceso. Cuando se detecta que uno de estos puntos se abre de manera no autorizada, el módulo envía un mensaje por MQTT en su tópico correspondiente hacia de la centra de procesamiento. Este mensaje activa la alarma según la configuración establecida, se generan alertas para el usuario a través de la aplicación móvil y activa los actuadores. En la Fig. 4.4 se visualiza los elementos que conforman el módulo de proximidad, así como la lista de partes en la Tabla 4.3.

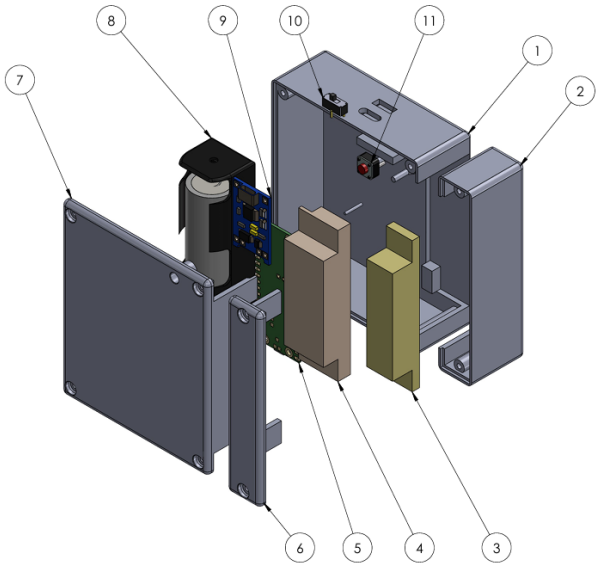


Fig. 4.4. Módulo de apertura.

Tabla 4.3
Lista de elementos módulo de apertura

Elemento	Descripción	Cantidad	Material
1	Base	1	PLA
2	Base imán	1	PLA
3	Imán	1	-
4	Interruptor	1	-
5	Raspberry Pico W	1	-
6	Tapa imán	1	PLA
7	Tapa	1	PLA
8	Portapilas con bateria 18650	1	-
9	Módulo de protección de carga TP4056	1	-
10	Switch deslizante	1	-
11	Pulsador	1	-

4.2.5 Módulo de vibración

El módulo de vibración, basado en la Raspberry Pi Pico W, detecta movimientos o vibraciones anómalas en superficies o estructuras, como puertas, ventanas o paredes. Al percibir vibraciones que puedan ser indicativas de intentos de forzar una entrada o romper una ventana, el módulo envía un mensaje por MQTT en su tópico correspondiente hacia de la centra de procesamiento. Este mensaje activa la alarma según la configuración establecida, se generan alertas para el usuario a través de la aplicación móvil y activa los actuadores. En la Fig. 4.5 se visualiza los elementos que conforman el módulo de vibración, así como la lista de partes en la Tabla 4.4.

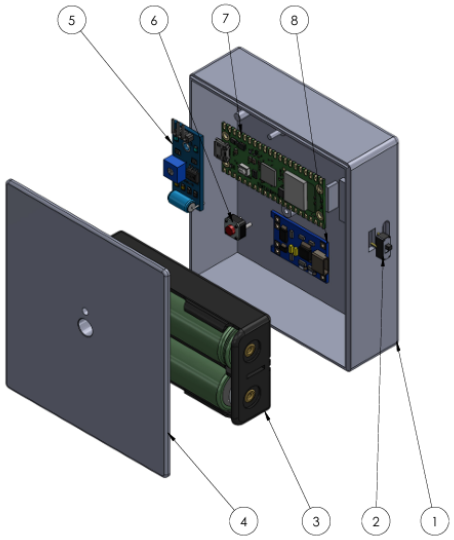


Fig. 4.5. Módulo de vibración.

Tabla 4.4
Lista de elementos módulo de vibración

Elemento	Descripción	Cantidad	Material
1	Base	1	PLA
2	Switch deslizante	1	-
3	Portapilas con baterías 18650	1	-
4	Tapa	1	PLA
5	Sensor vibración SW-420	1	-
6	Pulsador	1	-
7	Raspberry Pico W	1	-
8	Módulo de protección de carga TP4056	1	-

4.2.6 Módulo de detección de humo

El módulo de detección de humo, basado en la Raspberry Pi Pico W, monitorea la calidad del aire en busca de la presencia de humo o gases que puedan indicar un posible incendio. Al detectar una concentración de humo que supera un umbral preestablecido, el módulo envía un mensaje por MQTT en su tópico correspondiente hacia de la centra de procesamiento. Este mensaje activa la alarma según la configuración establecida, se generan alertas para el usuario a través de la aplicación móvil y activa los actuadores. En la Fig. 4.6 se visualiza los elementos que conforman el módulo de vibración, así como la lista de partes en la Tabla 4.5.

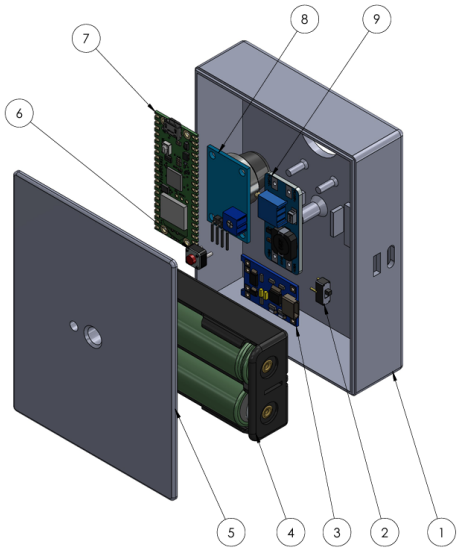


Fig. 4.6. Módulo de detección de humo

Tabla 4.5
Lista de elementos módulo de detección de humo

Elemento	Descripción	Cantidad	Material
1	Base	1	PLA
2	Switch deslizante	1	-
3	Módulo de protección de carga TP4056	1	-
4	Portapilas con baterías 18650	1	-
5	Tapa	1	PLA
6	Pulsador	1	-
7	Raspberry Pico W	1	-
8	Detector de Gas MQ-2	1	-
9	Elevador de voltaje MT3608	1	-

4.2.7 Especificaciones técnicas del sistema

4.2.7.1 Central de procesamiento

La central de procesamiento está basada en la Raspberry Pi Zero 2 W, seleccionada por su bajo costo, reducido consumo energético, tamaño compacto y capacidades adecuadas para la lógica del sistema y la gestión de la comunicación con los módulos sensores [28].

Componentes principales A continuación, se describen los componentes utilizados en el diseño del módulo central:

- **Raspberry Pi Zero 2 W:** Microcomputadora que actúa como unidad central del sistema de alarma.
- **Fuente de poder 220/110V a 12V 10A (con UPS):** Permite la alimentación del sistema convirtiendo la corriente alterna (220/110V) a una salida regulable de 12V 10A. Además, cuenta con la funcionalidad UPS que permite tomar la alimentación de una batería de 12 V de manera automática en caso de corte de energía. Además, su salida regulable permite ajustar el voltaje al nivel requerido [29].
- **Regulador de voltaje de 12V a 5V 3A:** Permite adaptar la tensión de entrada de 12V a 10A a los 5V a 2.5A requeridos por la Raspberry Pi Zero 2 W, garantizando una alimentación estable. Su alta eficiencia (hasta 92 %) y capacidad de corriente continua de 3A lo hacen adecuado para mantener el sistema operativo con bajo calentamiento y tamaño compacto [30].
- **Pulsador:** Se emplea para realizar el reinicio y apagado manual del sistema de forma segura.
- **Módulo infrarrojo KY-022:** Mediante un control remoto permite la desactivación remota del sistema de alarma en caso de no tener acceso a la aplicación móvil, haciendo uso de protocolos estándar IR. [31].

- **Módulo relé de 4 canales:** Permite el control de actuadores de manera independiente, mediante señales de bajo voltaje enviadas desde la central de procesamiento. [32].

Esquema de conexión En la Fig. 4.7 se muestra el circuito completo del módulo central. Este integra todos los componentes mencionados, donde la fuente de alimentación se conecta a la batería y alimenta tanto al regulador de voltaje como al módulo relé. La Raspberry Pi se encarga de gestionar los eventos, leer señales del sensor infrarrojo y activar las salidas mediante los relés según la lógica programada.

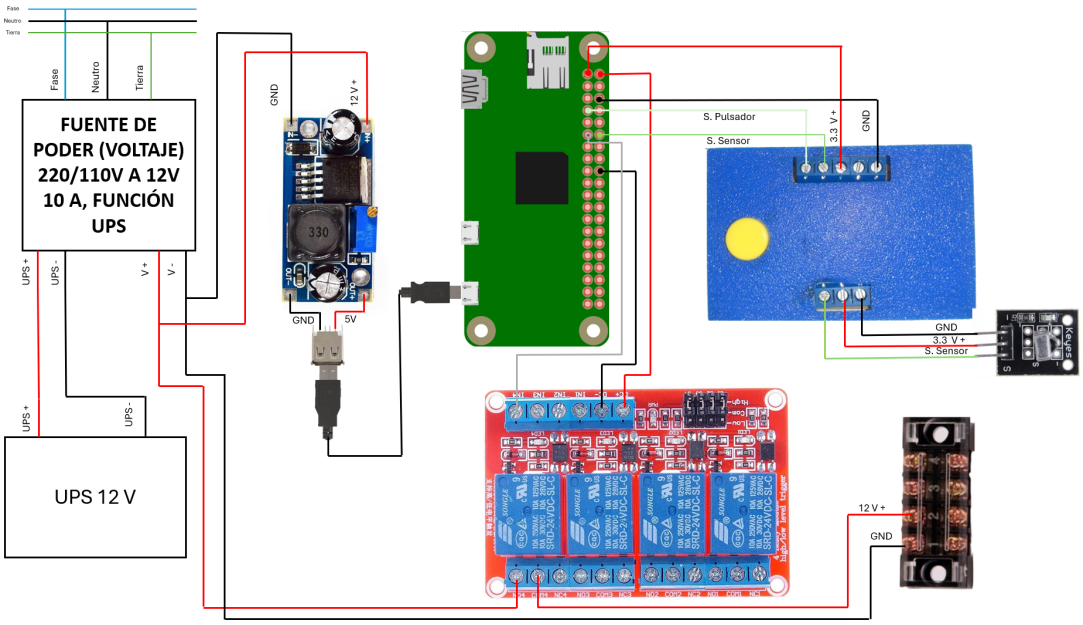


Fig. 4.7. Circuito de central de procesamiento

4.2.8 Módulos sensores

Los módulos sensores del sistema están basados en la Raspberry Pi Pico W, seleccionada por su bajo consumo energético, conectividad Wi-Fi integrada y tamaño compacto, características que la hacen adecuada para aplicaciones distribuidas como los sensores de una alarma IoT [33].

Los circuitos de cada módulo sensor cuentan con características similares, donde varía el tipo de sensor utilizado y en los casos necesarios se hace uso de un elevador de voltaje para la alimentación de estos. La elección de estos componentes se basa en criterios de rendimiento,

tamaño compacto y la estabilidad en su funcionamiento.

Componentes comunes A continuación, se detallan los componentes utilizados de forma común en los módulos sensores:

- **Raspberry Pi Pico W:**Microcontrolador que gestiona el sensor, la lógica programada para la activación, la lectura del sensor y la comunicación mediante MQTT.
- **Baterías 18650 de 3.7V, 3000mAh:** Fuente de alimentación recargable para cada módulo sensor. Módulo TP4056: Permite la carga segura de baterías de litio de 3.7V tipo 18650 mediante micro-USB o fuente externa. Incorpora protección contra sobrecarga, sobredescarga y cortocircuitos, asegurando la integridad de la batería y del sistema [34].
- **Interruptor deslizante:** Encendido/apagado manual del módulo.
- **Pulsador:** Reinicio manual del sistema en caso de desincronización o fallo.
- **Elevador de voltaje MT3608 (solo en algunos casos):** Se utiliza cuando el sensor requiere una tensión superior a la proporcionada por la batería [35].

Variantes por tipo de sensor La siguiente tabla resume las diferencias específicas entre los módulos, según el sensor empleado:

Tabla 4.6

Resumen de sensores utilizados en los módulos del sistema

Módulo	Sensor utilizado	Ventajas técnicas que motivaron su selección	Figura
Proximidad	PIR HC-SR501	Sensor ampliamente utilizado en sistemas de seguridad. Alta sensibilidad al movimiento humano, bajo consumo en reposo y fácil integración con microcontroladores [36].	Fig. 4.8
Vibración	SW-420	Sensor digital confiable y económico, ideal para detectar golpes o vibraciones bruscas en estructuras. Su respuesta rápida lo hace útil para detectar intentos de forzar accesos [37].	Fig. 4.9
Apertura	Interruptor magnético tipo reed	Alta fiabilidad mecánica, sin necesidad de alimentación. Ideal para la detección de aperturas en puertas o ventanas por su simplicidad y bajo costo [38].	Fig. 4.10
Humo	MQ-2	Sensor versátil y de bajo costo capaz de detectar humo y gases inflamables. Adecuado para ambientes interiores donde se desea detectar incendios o fugas tempranas [39].	Fig. 4.11

Esquemas de conexión A continuación, se presentan los esquemas eléctricos de cada módulo sensor. Como se observa, comparten gran parte de su circuito, variando únicamente el sensor principal y los componentes necesarios para su correcto funcionamiento.

4.2.8.1 Circuito del módulo de proximidad

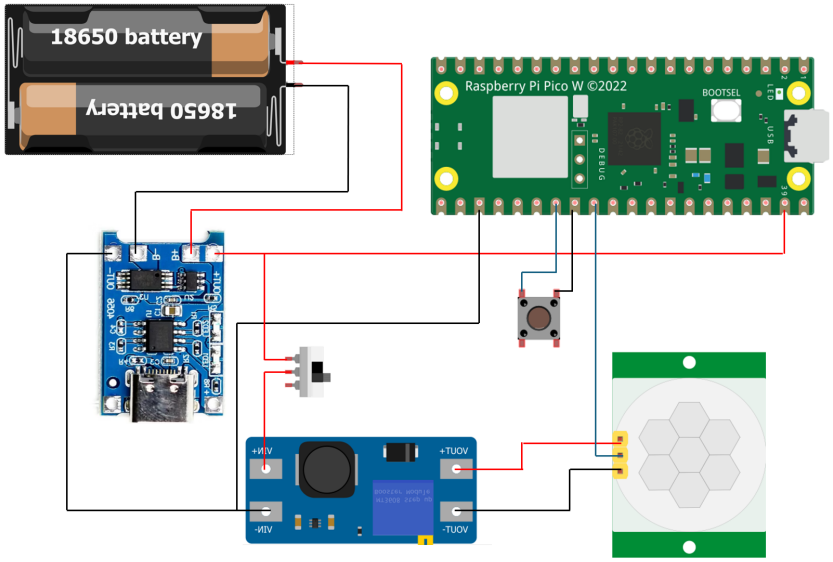


Fig. 4.8. Circuito del módulo de proximidad

4.2.8.2 Circuito del módulo de vibración

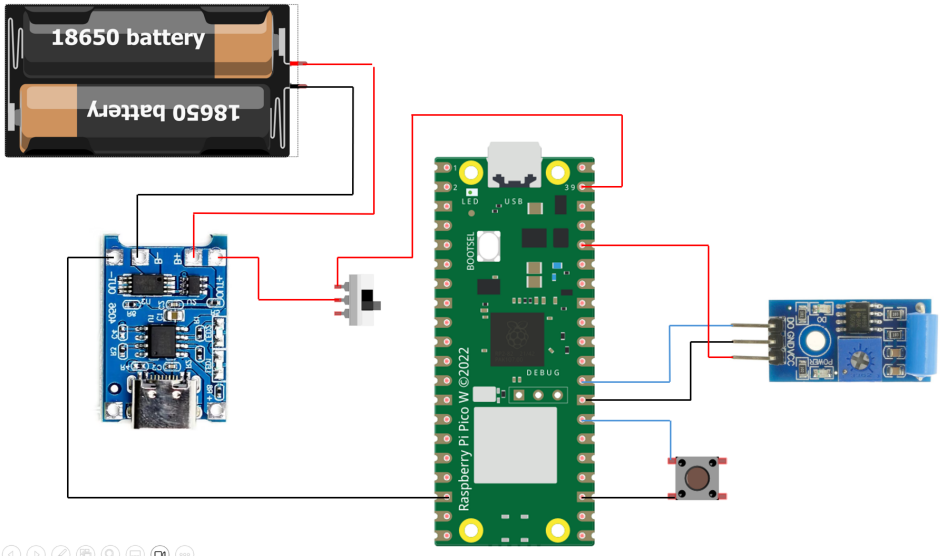


Fig. 4.9. Circuito del módulo de vibración

4.2.8.3 Circuito del módulo de apertura

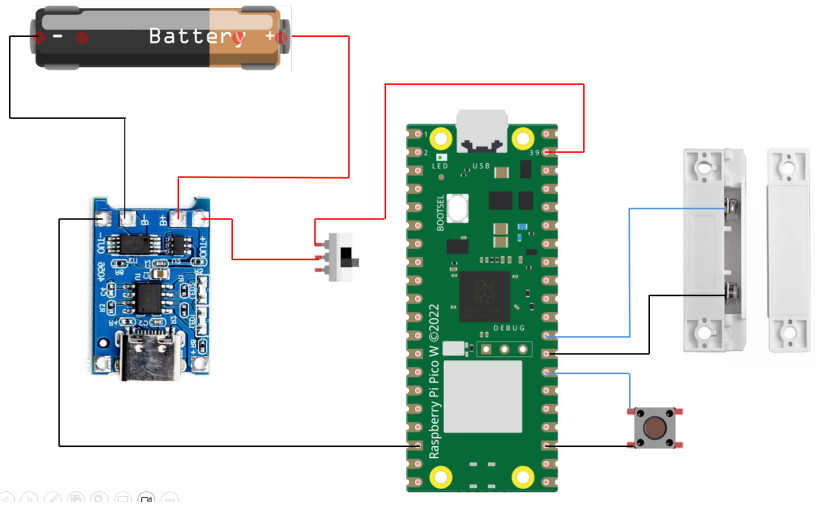


Fig. 4.10. Circuito del módulo de apertura

4.2.8.4 Circuito del módulo de humo

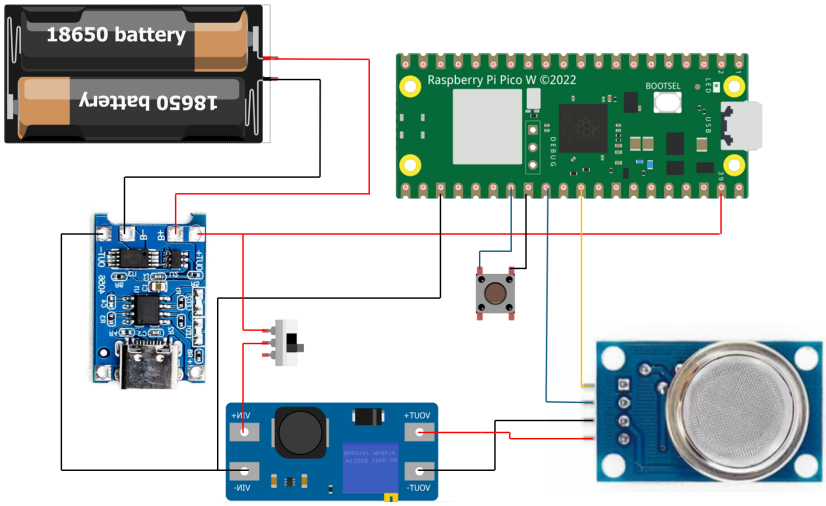


Fig. 4.11. Circuito del módulo de humo

4.2.9 Cálculos de consumo de corriente

Para calcular el tiempo de duración de las baterías acopladas en cada módulo sensor, es necesario especificar el consumo de corriente de cada uno de los componentes que los conforman. En la Tabla 4.7 se presentan los valores de corriente estimados para cada módulo sensor.

Tabla 4.7

Consumo estimado de corriente por módulo sensor

Módulo Sensor	**Componente**	**Corriente (mA)**
Proximidad (PIR)	Raspberry Pi Pico W	130
	Sensor PIR HC-SR501	50
	MT3608 (conversión)	10
	TP4056	1
	Total estimado	**191**
Vibración (SW-420)	Raspberry Pi Pico W	130
	Sensor de vibración SW-420	20
	TP4056	1
	Total estimado	**151**
Apertura (magnético)	Raspberry Pi Pico W	130
	Sensor magnético de apertura	5
	TP4056	1
	Total estimado	**136**
Detector de humo (MQ-2)	Raspberry Pi Pico W	130
	Sensor de gas MQ-2	150
	MT3608 (conversión)	15
	TP4056	1
	Total estimado	**296**

Una vez conocido el consumo estimado de corriente de cada módulo sensor, es posible calcular el tiempo de duración (en horas) mediante

$$\text{Duración (h)} = \frac{\text{Capacidad de la batería (mAh)}}{\text{Consumo total del módulo (mA)}}. \tag{4.1}$$

Utilizando la Ecuación 4.1 , se obtiene los tiempos de operación estimados para cada módulo sensor. En la Tabla 4.8 se presentan estos valores, considerando que los módulos de proximidad, vibración y detector de humo se alimentan con dos baterías 18650 de 3.7V y 3000 mAh conectadas en paralelo (6000 mAh totales), mientras que el módulo de apertura utiliza una única batería de 3000 mAh.

Tabla 4.8
Duración estimada de los módulos sensores

Módulo Sensor	Consumo Total (mA)	Duración Estimada (h)
Proximidad (PIR)	191	31.41
Vibración (SW-420)	151	39.74
Apertura (magnético)	136	22.06
Detector de Humo (MQ-2)	296	20.27

4.3 Programación y lógica del sistema

En la programación, se desarrolla la lógica que permita la configuración de la central de procesamiento y módulos sensores, para que estos generen una red Ad Hoc, reciban credenciales y se enlacen a la red especificada, por otra parte, una vez configuradas se implementan funciones que permitan el envío y recibimiento de mensajes por MQTT, el monitoreo continuo, la activación del sistema y la comunicación con los servicios Firebase por parte de la central de procesamiento.

4.3.1 Programación Raspberry Pi Zero 2 W

En la programación de la Raspberry Pi Zero 2 W, se utiliza el sistema operativo Raspberry Pi OS Lite junto con Python. A través de este entorno, la central de procesamiento configura tanto el levantamiento de la red Ad Hoc como el enlace con la red especificada. Por otra parte, se corre un bróker mosquitto MQTT para la comunicación con los sensores mediante, de igual manera, interactúa con los servicios de Firebase para el almacenamiento, notificación eventos y controla la activación del sistema de alarma IoT en respuesta a las detecciones realizadas

A continuación, se detallan las bibliotecas empleadas en la programación de la Raspberry Pi Zero 2 W: `pyrebase` [1], utilizada para establecer una conexión simultánea con Firebase y permitir la interacción en tiempo real con la base de datos; `paho.mqtt.client` [2], empleada para

[1]Enlace: `https://github.com/thisbejim/Pyrebase`
[2]Enlace: `https://pypi.org/project/paho-mqtt/`

gestionar la comunicación MQTT entre los sensores y el broker; `firebase-admin` [3], que permite el acceso al servicio de mensajería de Firebase; `http.server` [4], usada para implementar un servidor HTTP encargado de recibir las credenciales de red enviadas por la aplicación móvil; y `urllib.parse` [5], que facilita el análisis y procesamiento de URLs para el manejo de solicitudes HTTP.

Por otra parte, en el sistema operativo se emplean herramientas adicionales. El paquete `LIRC` [6] se utiliza para habilitar el soporte de infrarrojos del sensor y permitir el reconocimiento del control remoto, registrando su código para asegurar que solo el dispositivo autenticado sea aceptado por el sistema. Asimismo, el broker `Mosquitto` [7] actúa como intermediario MQTT de código abierto, gestionando de manera eficiente el envío y recepción de mensajes entre los sensores y la central de procesamiento del sistema de alarma IoT.

4.3.1.1 Diagramas de flujo

En la Fig. 4.12 se muestra el diagrama de flujo con la secuencia lógica para la autenticación del usuario y la configuración de la central de procesamiento.

[3]Enlace: `https://firebase.google.com/docs/admin/setup?hl=es#python`
[4]Enlace: `https://docs.python.org/es/3.13/library/http.server.html`
[5]Enlace: `https://docs.python.org/3/library/urllib.parse.html`
[6]Enlace: `https://www.lirc.org/`
[7]Enlace: `https://mosquitto.org/blog/2013/01/mosquitto-debian-repository/`

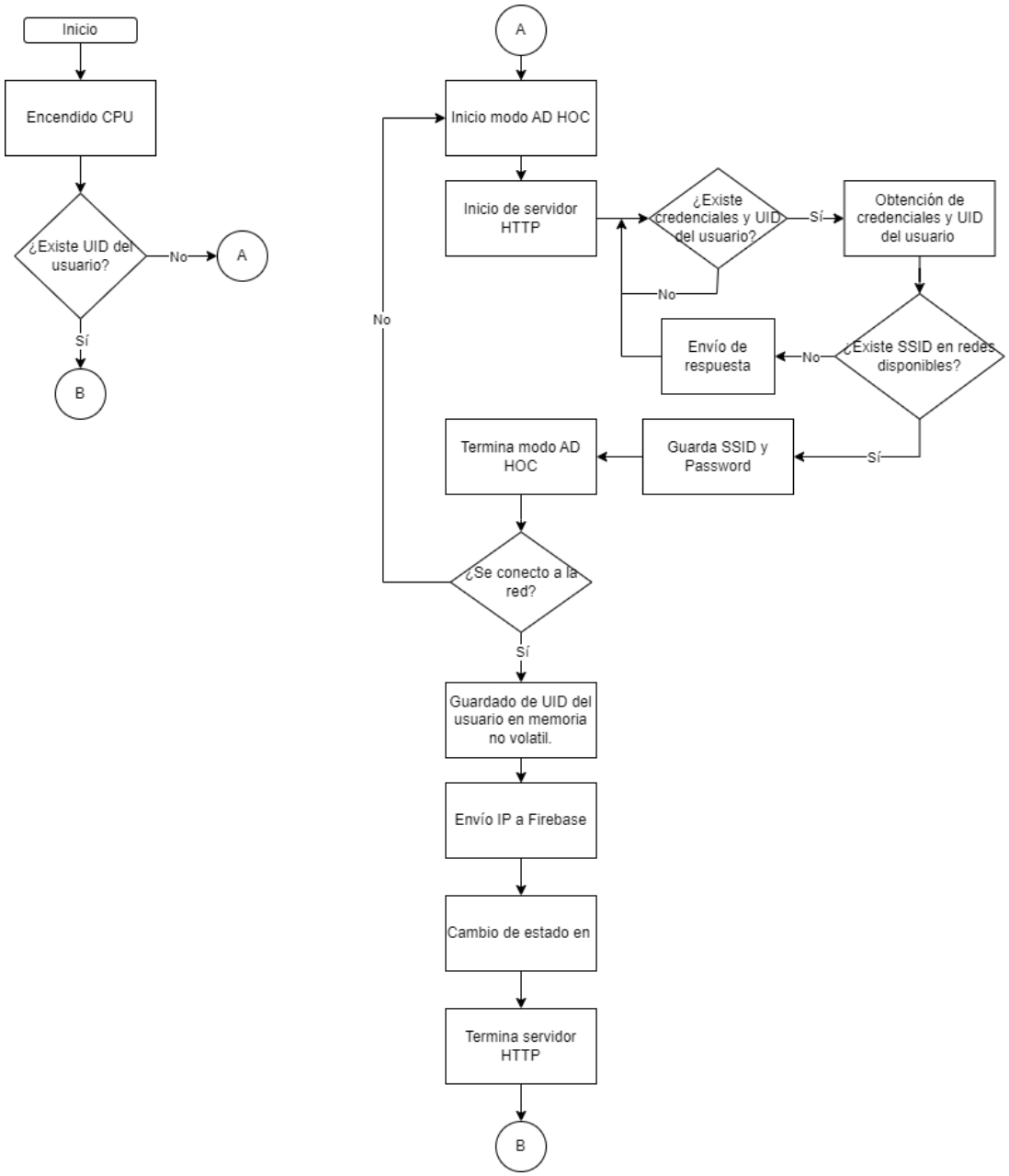


Fig. 4.12. Diagrama de flujo de la parte A.

En la Fig. 4.13 se muestra el diagrama de flujo con la secuencia lógica para la obtención de datos del usuario, la comunicación con Firebase y la gestión de los modos de alarma.

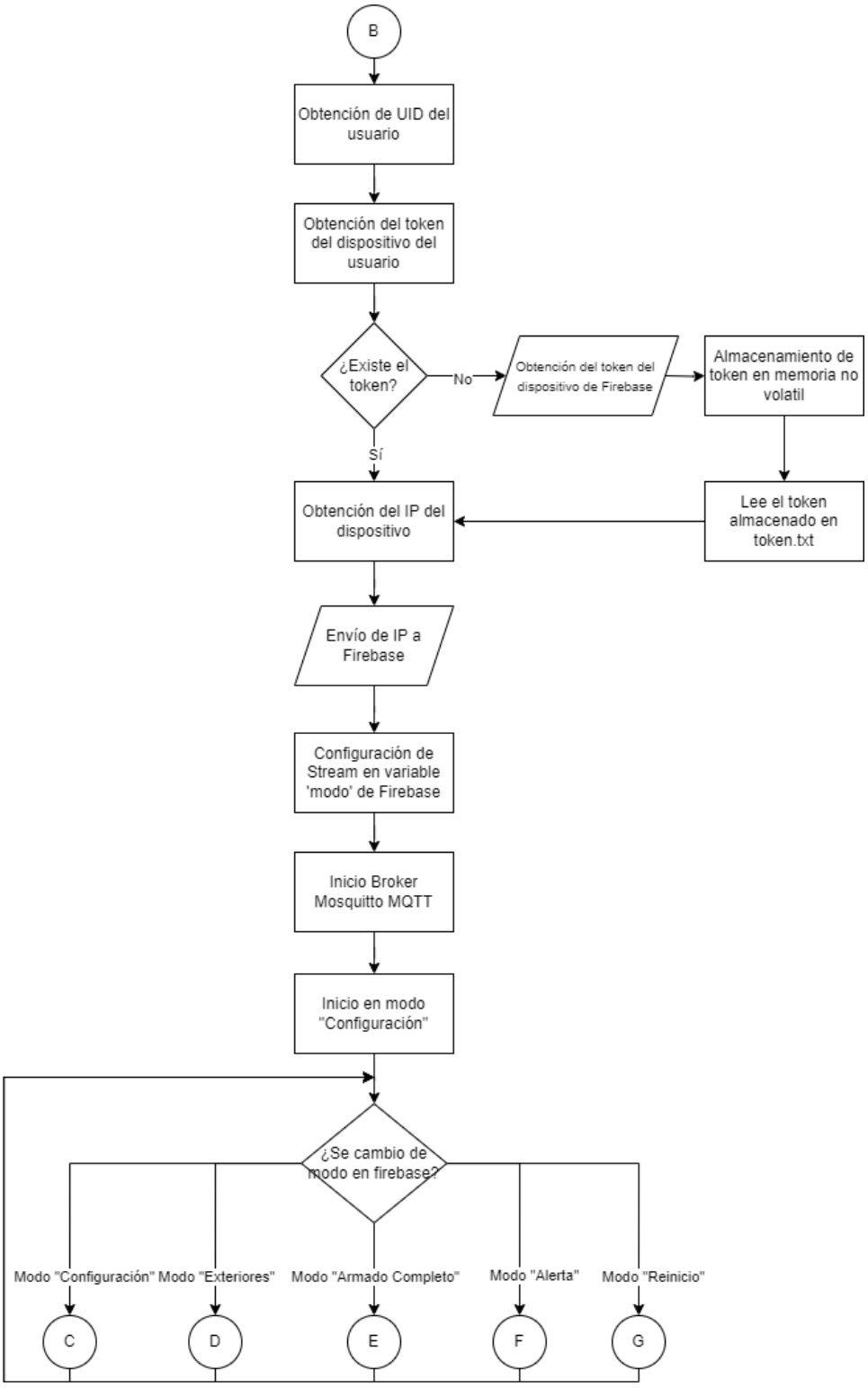


Fig. 4.13. Diagrama de flujo de la parte B.

En la Fig. 4.14 se muestra el diagrama de flujo con la secuencia lógica que se realiza en cada modo de alarma: Configuración, Exteriores y Armado Completo.

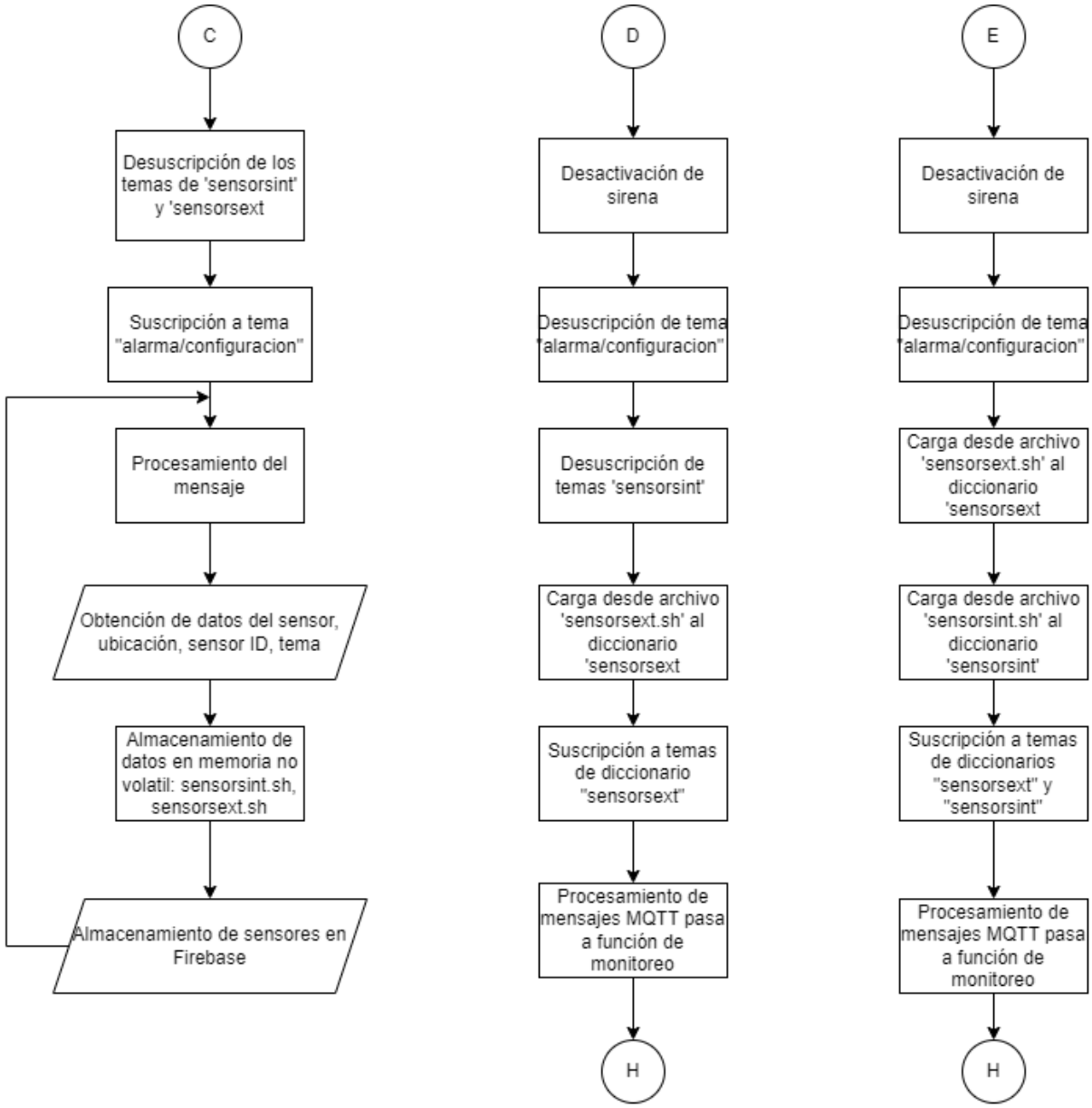


Fig. 4.14. Diagrama de flujo de la parte C, D, E.

En la Fig. 4.15 se muestra el diagrama de flujo con la secuencia lógica para el monitoreo y la gestión de los mensajes de los sensores.

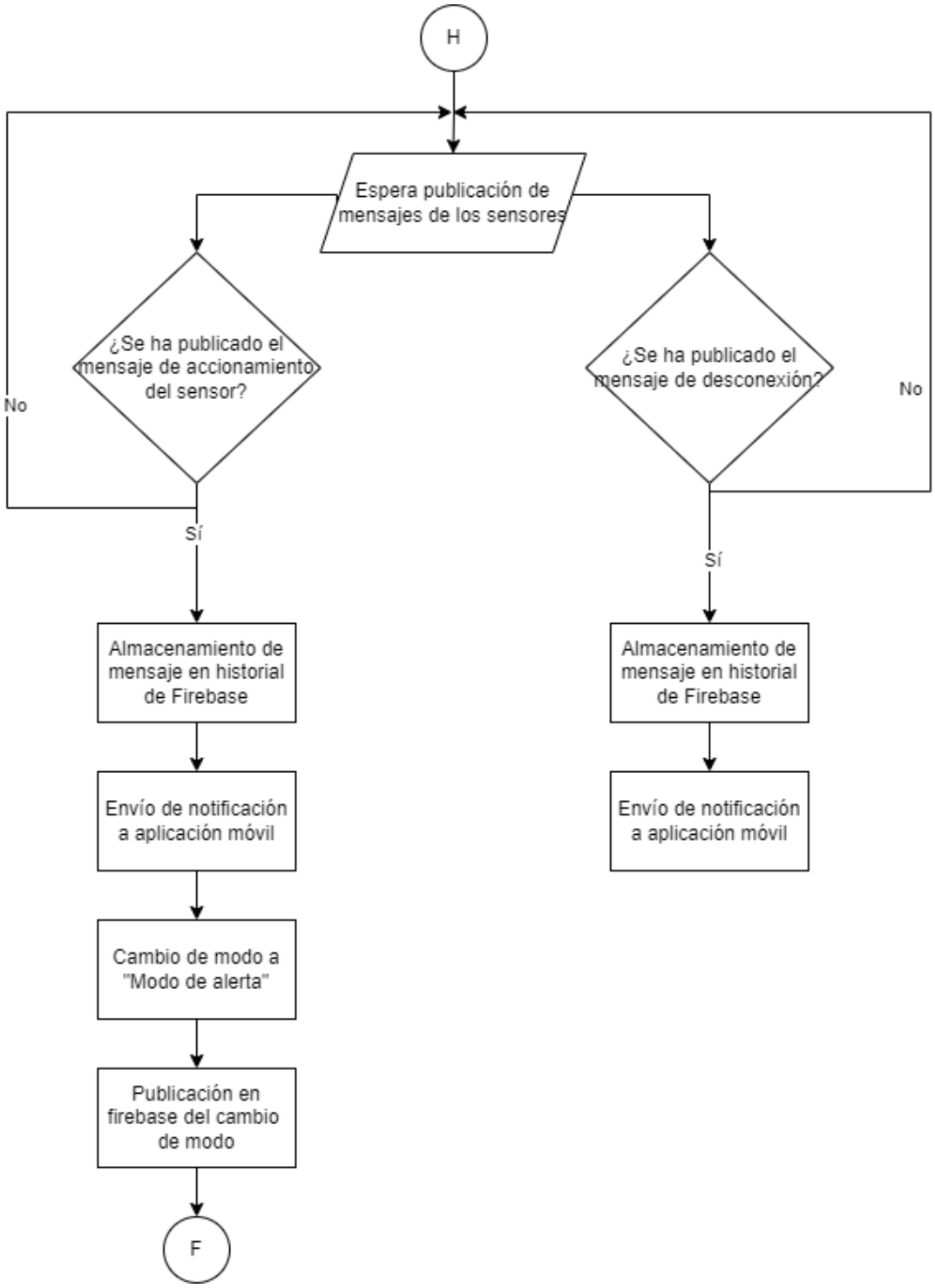


Fig. 4.15. Diagrama de flujo de la parte H.

En la Fig. 4.16 se muestra el diagrama de flujo con la secuencia lógica que se ejecuta al activarse el sistema de alarma.

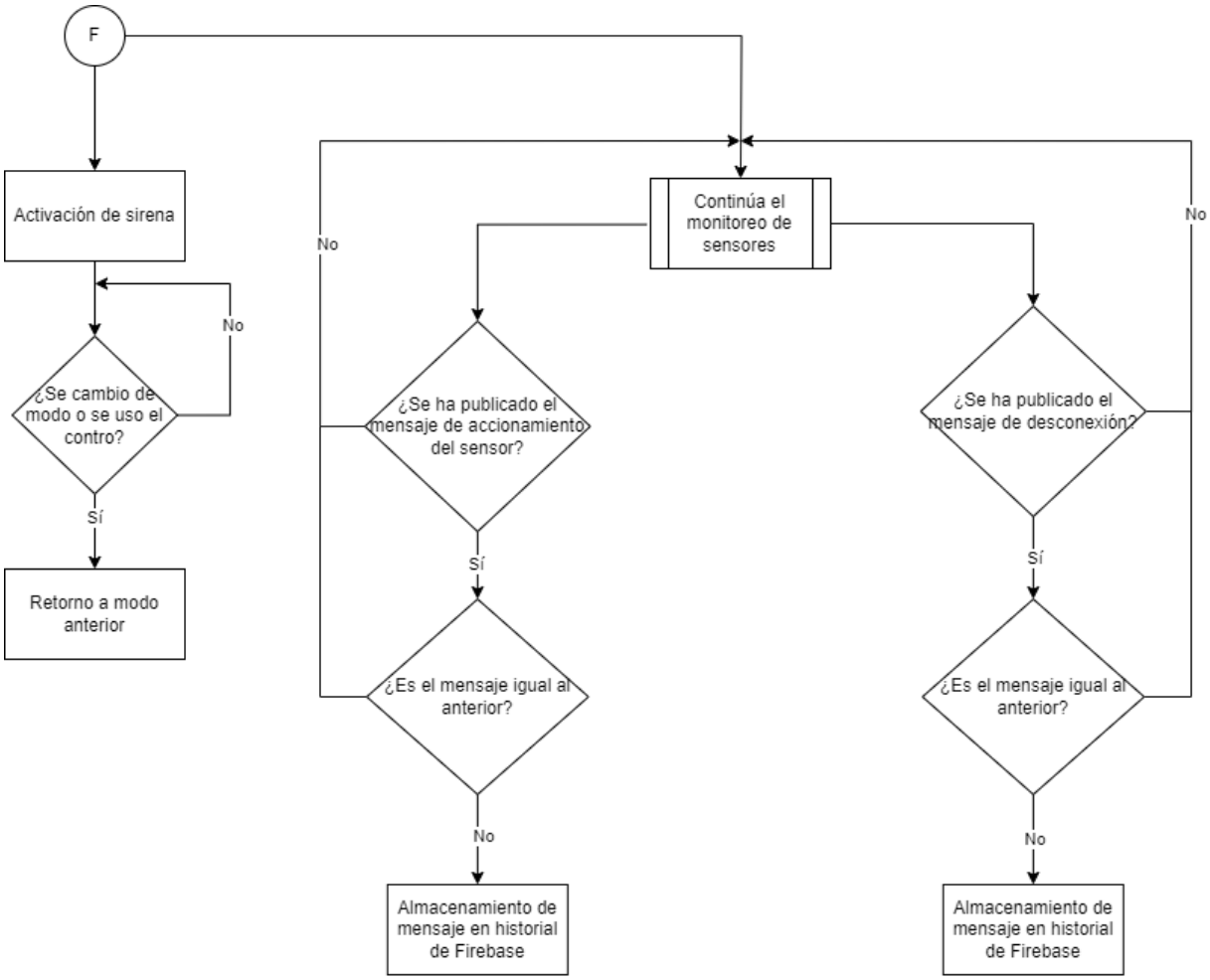


Fig. 4.16. Diagrama de flujo de la parte F.

En la Fig. 4.17 se muestra el diagrama de flujo con la secuencia lógica que se ejecuta al momento de reiniciar la central de procesamiento.

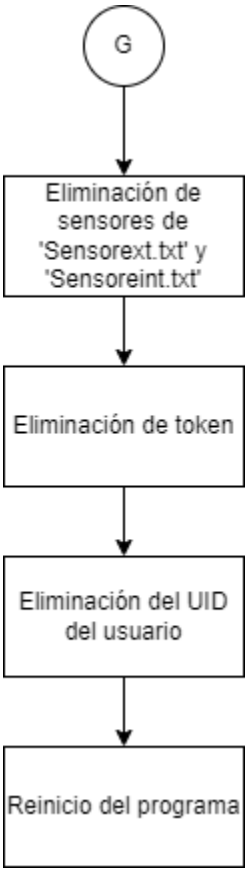


Fig. 4.17. Diagrama de flujo de la parte G.

4.3.2 Programación Raspberry Pico W

En la programación de la Raspberry Pi Pico W, se emplea el entorno Thonny junto con MicroPython. Cada módulo sensor gestiona su configuración de red, la lectura de datos desde Firebase, la conexión con el broker MQTT, la detección de eventos y la transmisión de información a través de MQTT.

A continuación se detallan las bibliotecas empleadas en la programación de la Raspberry Pico W:

En la Raspberry Pi Pico W se emplean las bibliotecas `urequests` [8], utilizada para realizar solicitudes HTTP desde MicroPython y permitir la interacción con Firebase para el envío y

[8]Enlace: `https://pypi.org/project/micropython-urequests/`

recepción de datos, y `umqtt.robust` [9], una extensión de `umqtt` [10] que gestiona la comunicación mediante el protocolo MQTT. Esta biblioteca permite no solo la publicación y suscripción a topics en el broker, sino también la reconexión automática en caso de pérdida de conexión.

4.3.2.1 Diagramas de flujo

En la Fig. 4.18 se muestra el diagrama de flujo con la secuencia lógica para el inicio del programa y en la Fig. 4.19 la configuración del módulo sensor para el recibimiento de credenciales de red.

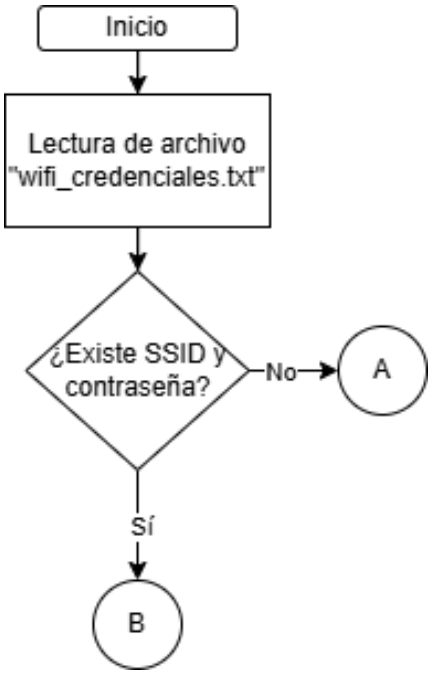


Fig. 4.18. Diagrama de flujo del inicio del programa.

[9] Enlace: `https://github.com/micropython/micropython-lib/tree/master/micropython/umqtt.robust`

[10] Enlace: `https://mpython.readthedocs.io/en/v2.2.1/library/mPython/umqtt.simple.html`

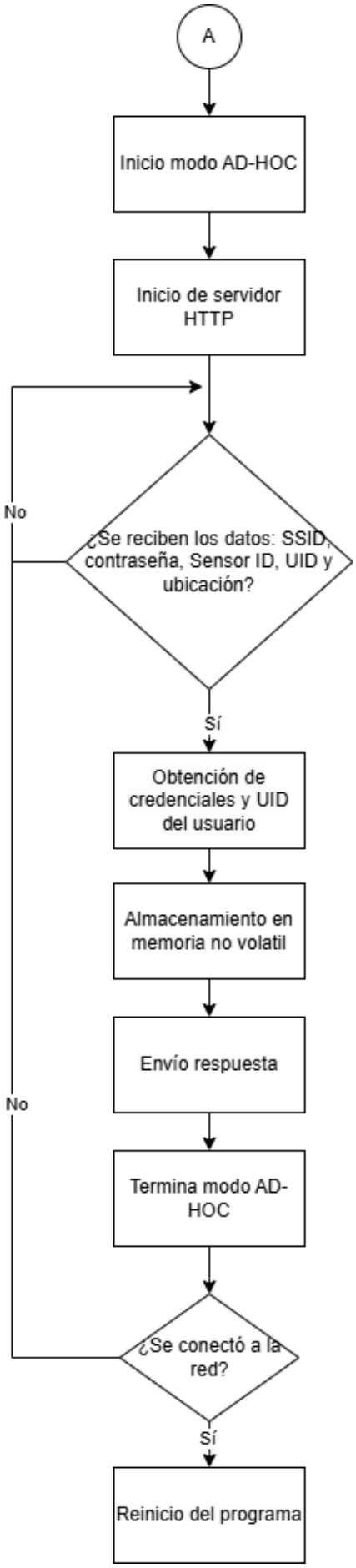


Fig. 4.19. Diagrama de flujo de la parte A.

En la Fig. 4.20 se muestra el diagrama de flujo con la secuencia lógica para la conexión a la red, el proceso de agregado del sensor, la lectura de datos en Firebase y la conexión con el broker MQTT.

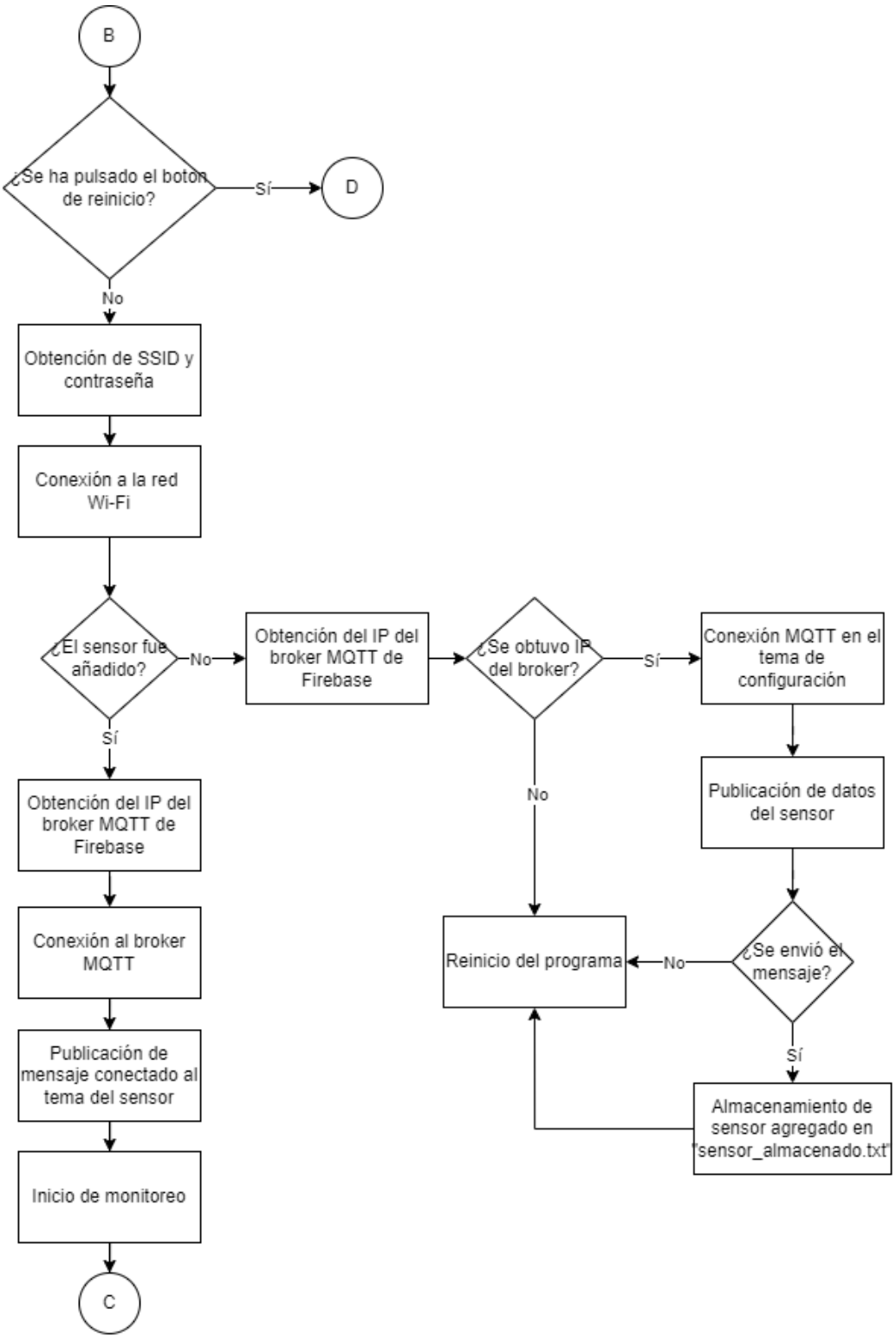


Fig. 4.20. Diagrama de flujo de la parte B.

En la Fig. 4.21 se muestra el diagrama de flujo con la secuencia lógica que se realiza durante el monitoreo del sensor.

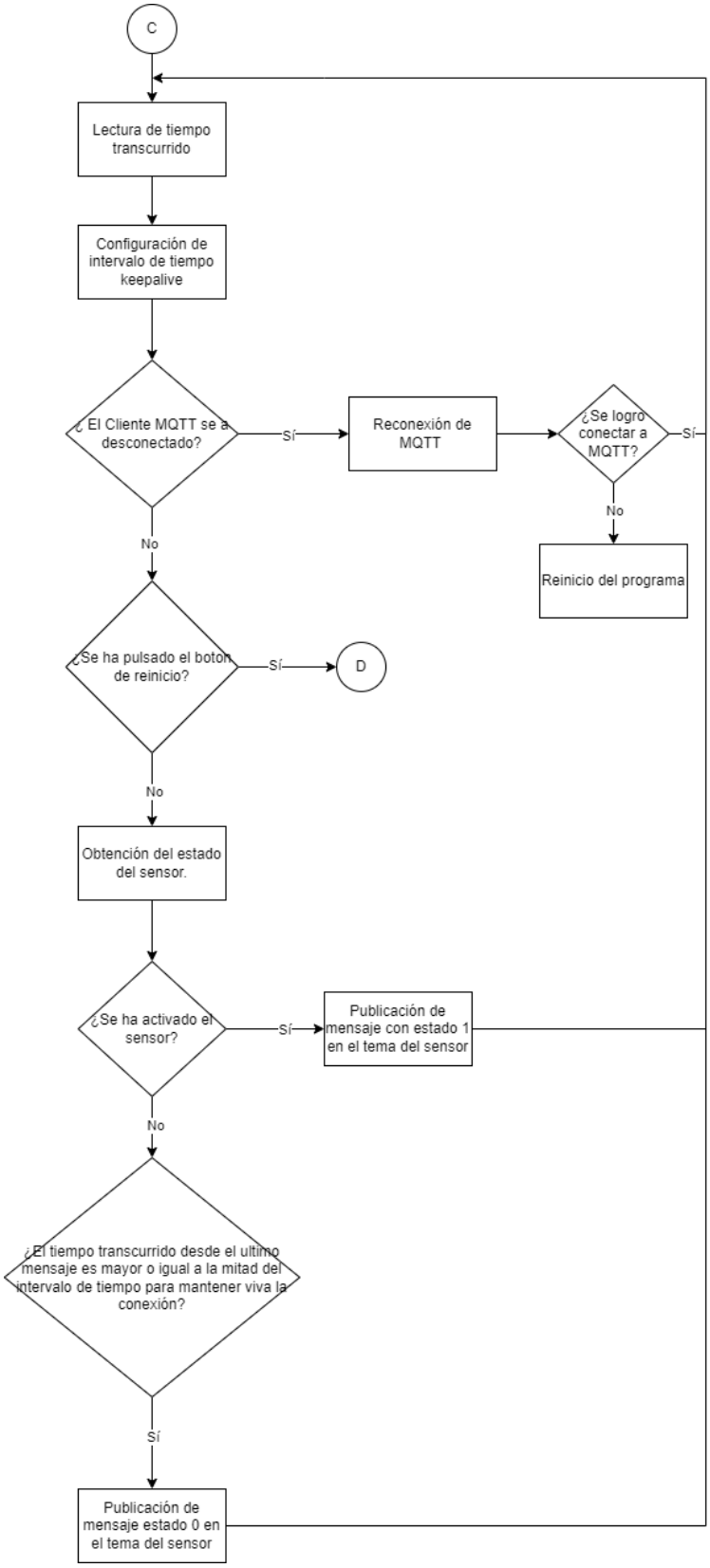


Fig. 4.21. Diagrama de flujo de la parte C.

En la Fig. 4.22 se muestra el diagrama de flujo con la secuencia lógica que se ejecuta al momento de pulsar el botón de reinicio.

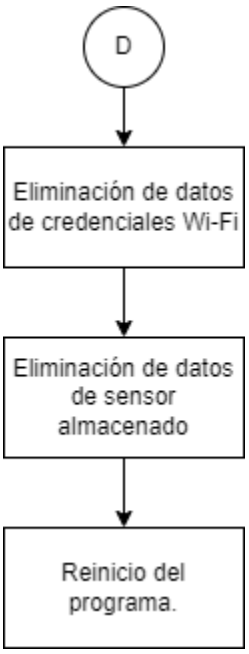


Fig. 4.22. Diagrama de flujo de la parte D.

4.4 Almacenamiento de datos

Mediante la integración de Firebase Realtime Database [11] se logra realizar una comunicación bidireccional entre la central de procesamiento y la aplicación móvil. Permitiendo la gestión remota por parte del usuario, donde la central de procesamiento registrar eventos, actualizar estados y recibir configuraciones, mientras que la aplicación móvil y los sensores pueden acceder a estos datos para su correcto funcionamiento.

En la Tabla 4.9 se indican los nodos generados para cada usuario, donde mediante el Identificador de Usuario (UID, Unique Identifier) que se le asigna a cada usuario permite la estructuración y organización de los nodos en la base de datos, permitiendo la gestión personalizada de la información.

[11] Enlace: `https://firebase.google.com/docs/database/`

Tabla 4.9

Nodos generados en Firebase.

Nodo	**Tipo**	**Descripción**
UID	Nodo	Nodo principal en base al identificador único de cada usuario
UID/Alerta	String	Indica el estado de alarma
UID/Configuracion	Nodo	Nodo que contiene datos de configuración
UID/Configuracion/IPCPU	String	Almacena la dirección IP del broker MQTT
UID/Configuracion/Nsensores	Number	Indica el número de sensor siguiente a agregar
UID/Configuracion/modo	Number	Indica el modo actual de la alarma (0: Configuración, 1: Exterior, 2: Completo, 10: Alerta, 11: Reinicio)
UID/Configuracion/tokenFCM	String	Indica el token del dispositivo al que se enviarán las notificaciones
UID/Historial	Nodo	Contiene los eventos registrados según su fecha y hora
UID/Historial/fecha/hora	String	Nodo individual de evento ocurrido
UID/Sensores	Nodo	Nodo que contiene los sensores registrados en la alarma
UID/Sensores/Exterior	Nodo	Nodo que contiene datos de sensores exteriores
UID/Sensores/Exterior/sensorX	Nodo	Nodo individual de cada sensor
UID/Sensores/Exterior/sensorX/topic	String	Indica el tema del sensor almacenado
UID/Sensores/Exterior/sensorX/estado	String	Indica el estado del sensor
UID/Sensores/Interior	Nodo	Nodo que contiene datos de sensores interiores
UID/Sensores/Interior/sensorX	Nodo	Nodo individual de cada sensor
UID/Sensores/Interior/sensorX/topic	String	Indica el tema del sensor almacenado
UID/Sensores/Interior/sensorX/estado	String	Indica el estado del sensor

4.5 Aplicación móvil

La aplicación está desarrollada en React Native CLI [12] con TypeScript [13], enfocándose en la plataforma Android, por lo cual solo los usuarios con dispositivos Android pueden instalarla y acceder a sus funcionalidades.

4.5.1 Comunicación con Firebase

En la aplicación móvil se implementa múltiples servicios de Firebase, incluyendo autenticación, Realtime Database y Firebase Cloud Messaging, los cuales permiten una gestión segura, almacenamiento en tiempo real y la notificación al dispositivo móvil.

En la Fig. 4.23 se muestra el diagrama de flujo con la secuencia lógica que se realiza para el inicio de sesión o para el registro de un nuevo usuario

[12] Enlace: `https://reactnative.dev/docs/environment-setup`
[13] Enlace: `https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.htmlf`

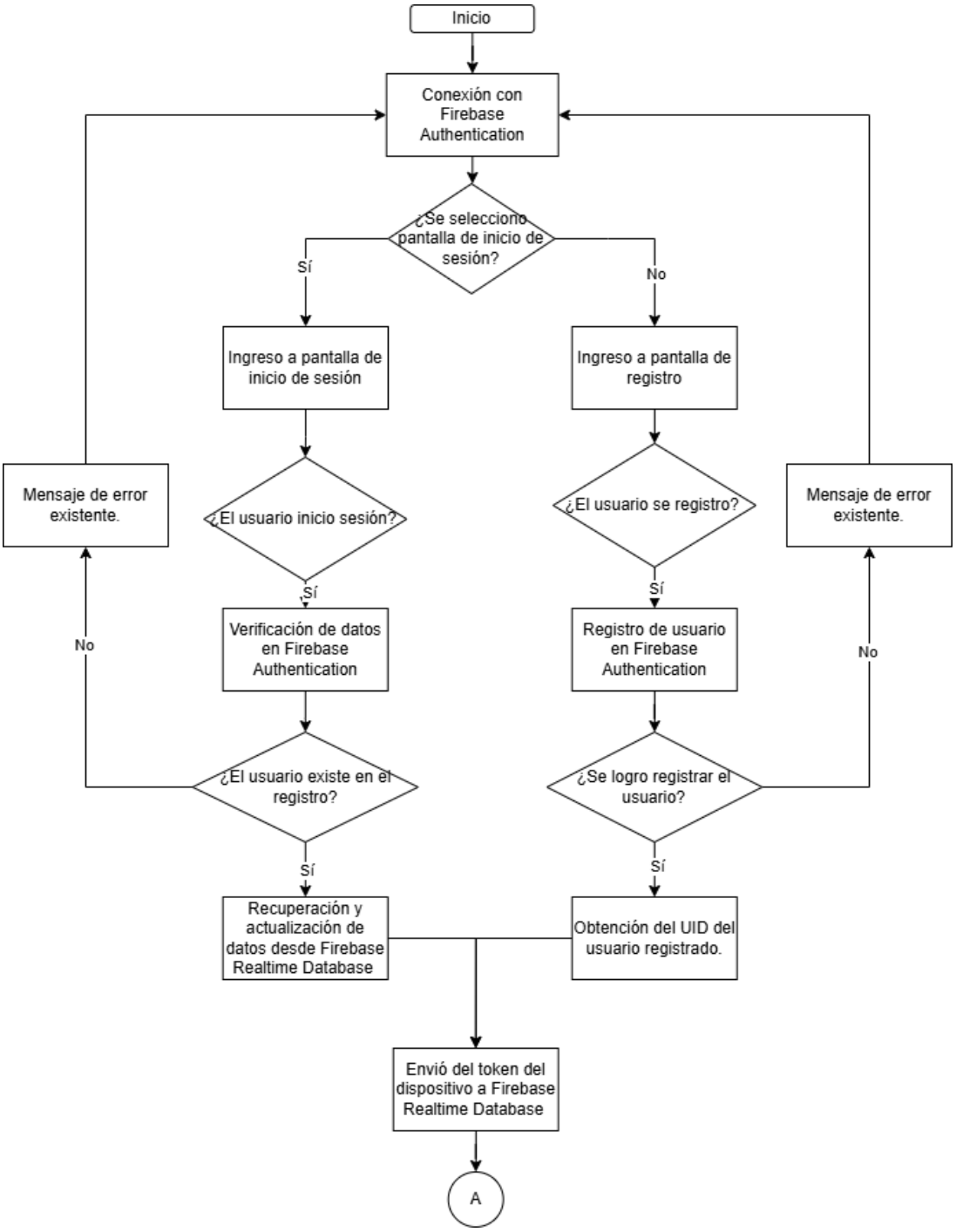


Fig. 4.23. Diagrama de flujo inicio de sesión y registro en la aplicación

En la Fig. 4.24 se muestra el diagrama de flujo que representa la secuencia lógica para la actualización del modo de alarma, la recepción de notificaciones a través de Firebase Cloud Messaging y la conexión simultánea para la actualización del estado de la alarma desde Firebase.

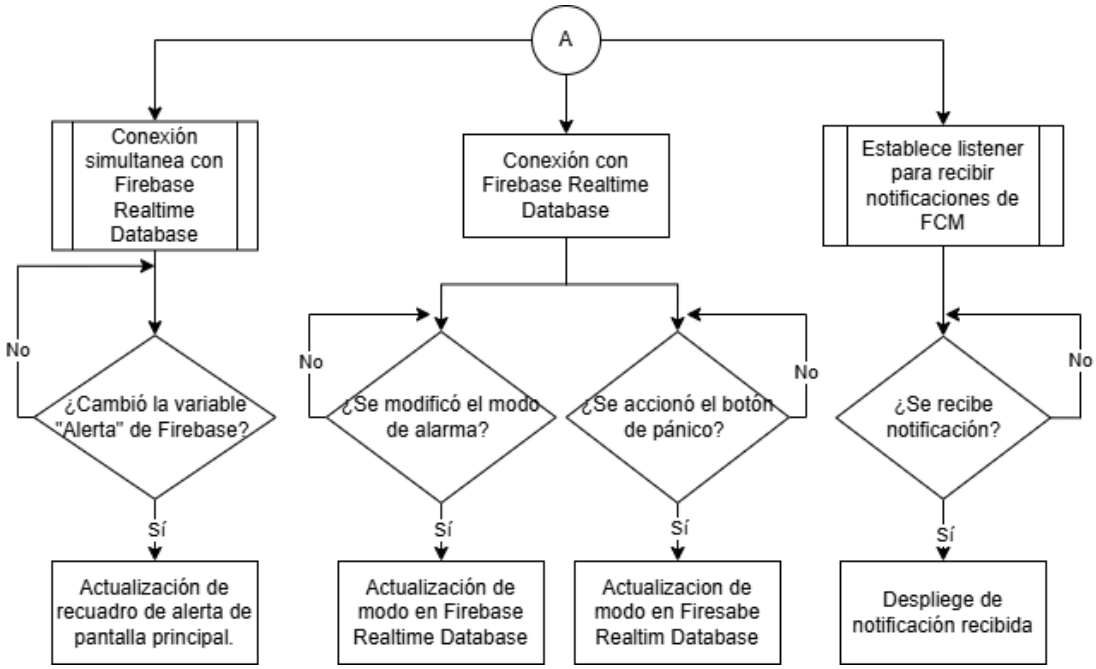
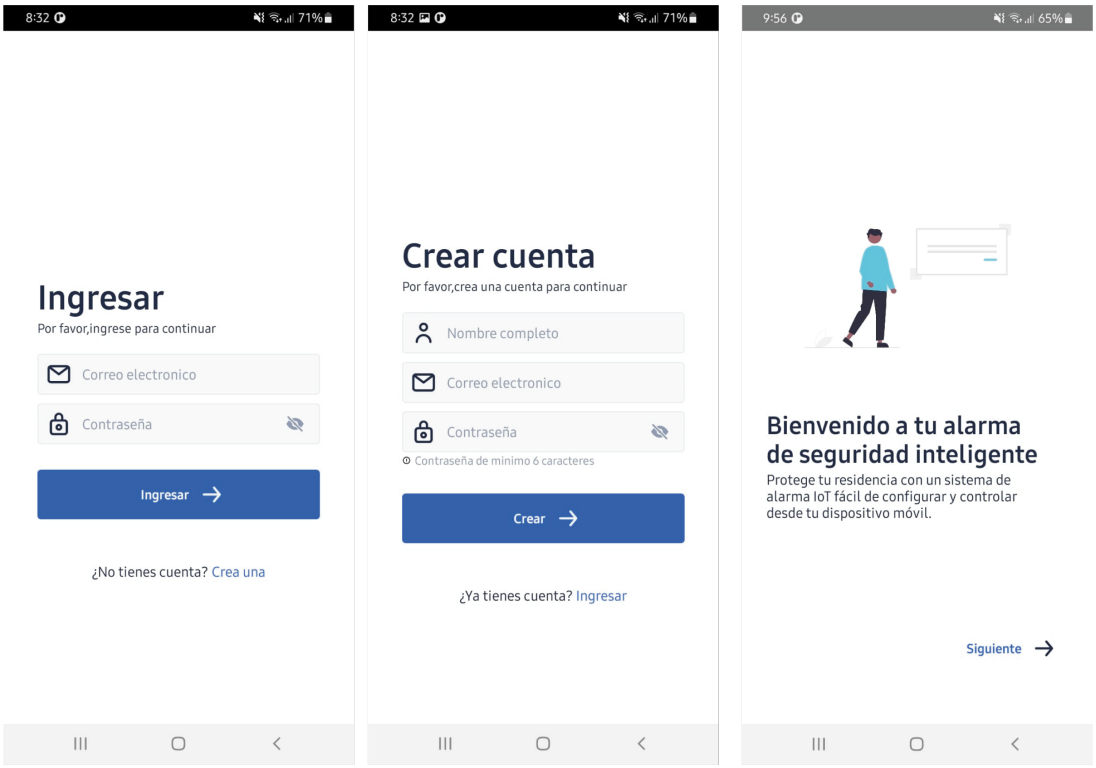


Fig. 4.24. Diagrama de flujo lectura y actualización de datos en la aplicación

4.5.2 Interfaz de Usuario

En la Fig. 4.25 (a) se muestra la pantalla de ingreso del usuario, la cual facilita el proceso de autenticación dentro del sistema de alarma. Por otro lado, la Fig. 4.25 (b) presenta la pantalla de registro, donde el usuario ingresa sus datos para crear una cuenta y acceder por primera vez a la aplicación y en la Fig. 4.25 (c) se muestra la pantalla con diapositivas de bienvenida al usuario nuevo.



a) b) c)

Fig. 4.25. Pantallas de inicio. (a) Pantalla de ingreso, (b) Pantalla de registro, (c) Pantalla de bienvenida.

En la Fig. 4.26 (a) se muestra la pantalla principal de la aplicación una vez que el usuario ya se ha autentificado, que incluye un recuadro del estado de alarma, botones para cambiar el modo de alarma, acceder a la lista de sensores, activar el botón de pánico, visualizar el historial y gestionar los contactos de emergencia. También en la parte superior cuenta con opciones de ayuda y para acceder a la pantalla de configuración. En la Fig. 4.26 (b) se presenta la ventana modal para la selección de modos de alarma. En la Fig. 4.26 (c) se muestra la pantalla con la lista de sensores agregados. La Fig. 4.26 (d) exhibe la pantalla de historial, donde se almacenan y visualizan los eventos ocurridos. En la Fig. 4.26 (e) se observa la ventana modal con la lista de contactos registrados. Finalmente, en la Fig. 4.26 (f) se muestra la pantalla con diapositivas de ayuda para el usuario.

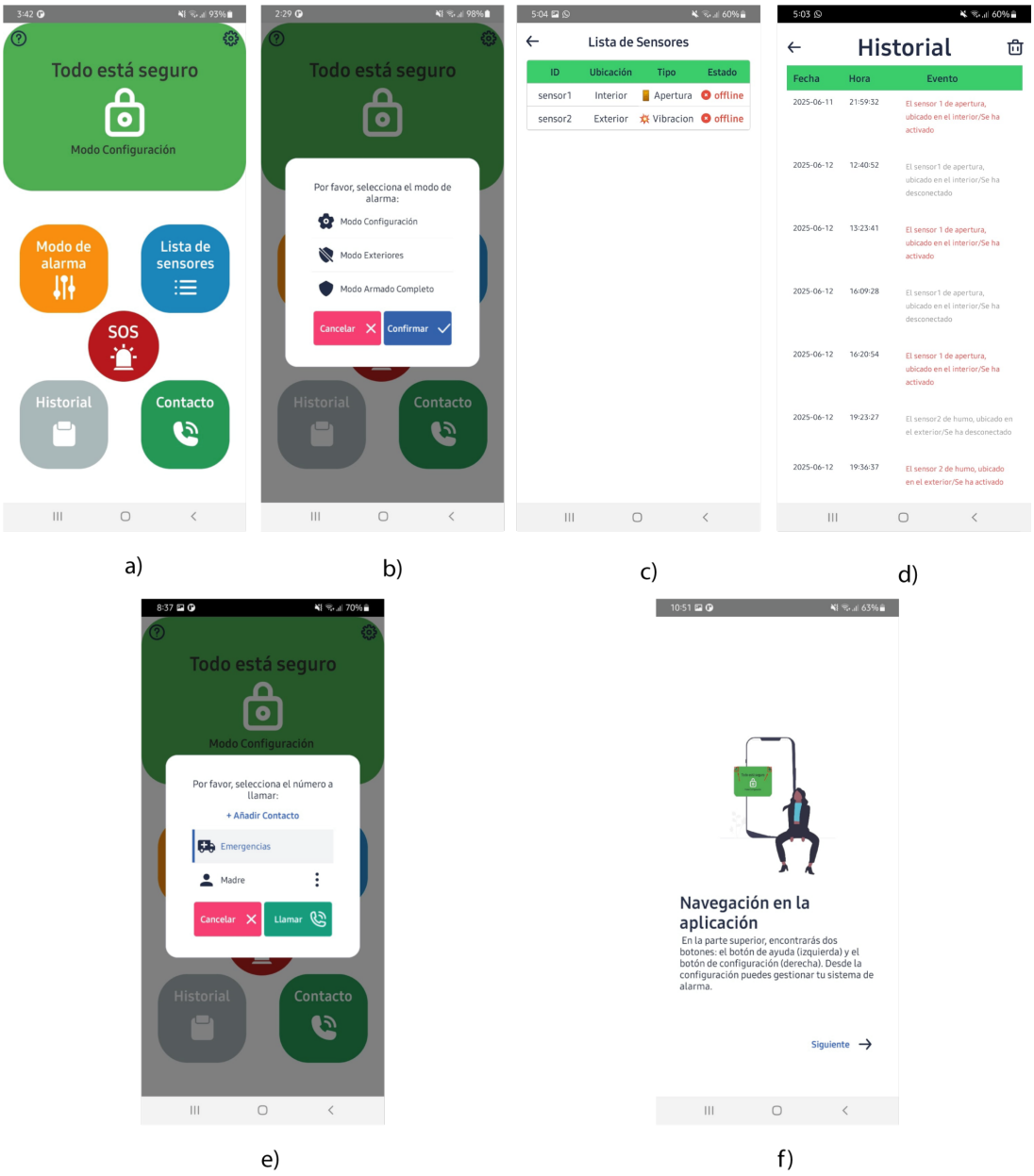


Fig. 4.26. Componentes de pantalla principal. (a) pantalla principal, (b) ventana modal de cambio de modo, (c) pantalla de lista de sensores, (d) pantalla de historial, (e) ventana modal de contactos, (f) pantalla de ayuda.

En la Fig. 4.27 (a) se muestra la pantalla de configuración, que incluye botones para la configuración de red de la central de procesamiento, la adición de nuevos sensores, el reinicio de la central de procesamiento y el cierre de sesión. En la Fig. 4.27 (b) se presenta la pantalla de configuración de red, donde se ingresan las credenciales de red que serán enviadas a la central de procesamiento. En la Fig. 4.27 (c) se muestra la ventana modal para agregar nuevos sensores, en la cual se selecciona la ubicación del sensor y se envían las credenciales de red, las mismas que utiliza la central de procesamiento. Finalmente, en la Fig. 4.27 (d) se muestra la pantalla

con diapositivas informativas sobre el proceso de reinicio del dispositivo.

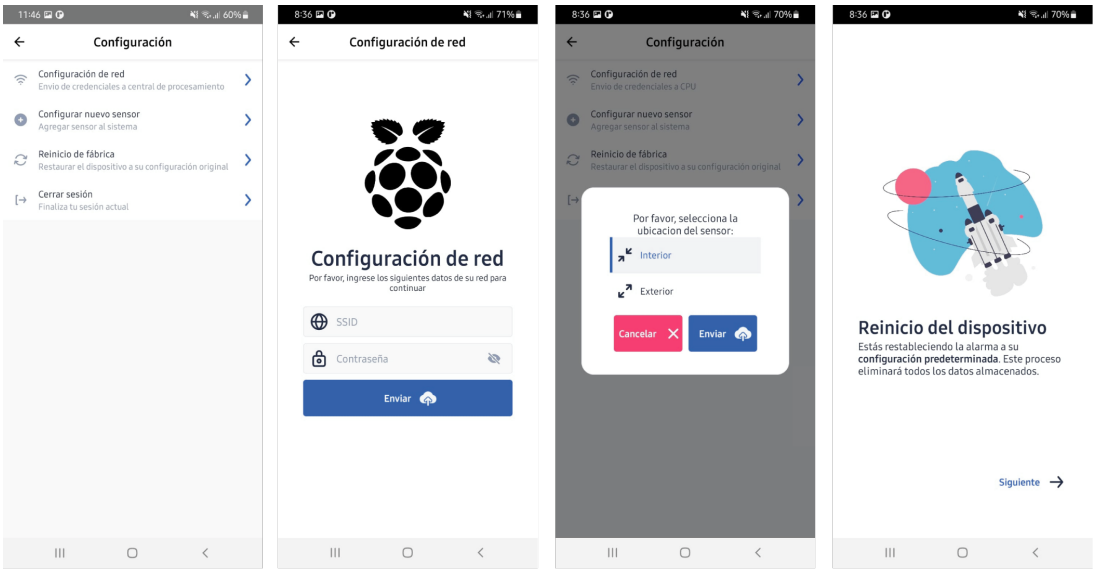


Fig. 4.27. Componentes de pantalla de configuración. (a) pantalla configuración, (b) pantalla de configuración de central de procesamiento, (c) ventana modal de configuración de sensores, (d) pantalla de reinicio de dispositivo.

4.6 Pruebas de funcionamiento

A continuación, se presentan las diferentes pruebas que se realizan para validar el funcionamiento del sistema de alarma, detallando datos y resultados obtenidos en pruebas de conectividad, tiempos de reacción y estabilidad.

4.6.1 Conectividad de los módulos sensores

Mediante la evaluación de la conectividad de los módulos sensores en diferentes ubicaciones dentro del entorno de pruebas, se verifica el correcto establecimiento de la conexión Wi-Fi y la transmisión efectiva del mensaje MQTT hacia la unidad central. Cada sensor, al ser activado, intenta conectarse a la red configurada y enviar un evento al broker MQTT, simulando condiciones reales de funcionamiento.

Las pruebas se realizan en diferentes ubicaciones tanto en zonas cercanas como alejadas del

punto de acceso a la red Wi-Fi, con el fin de observar el comportamiento de los módulos sensores ante variaciones de intensidad de la red. Durante la evaluación el módulo sensor se posiciona en los distintos puntos que se muestran en la Fig. 4.28. El croquis muestra el área de pruebas, incluyendo la referencia espacial como el posicionamiento del router y las ubicaciones donde se instala el módulo sensor.

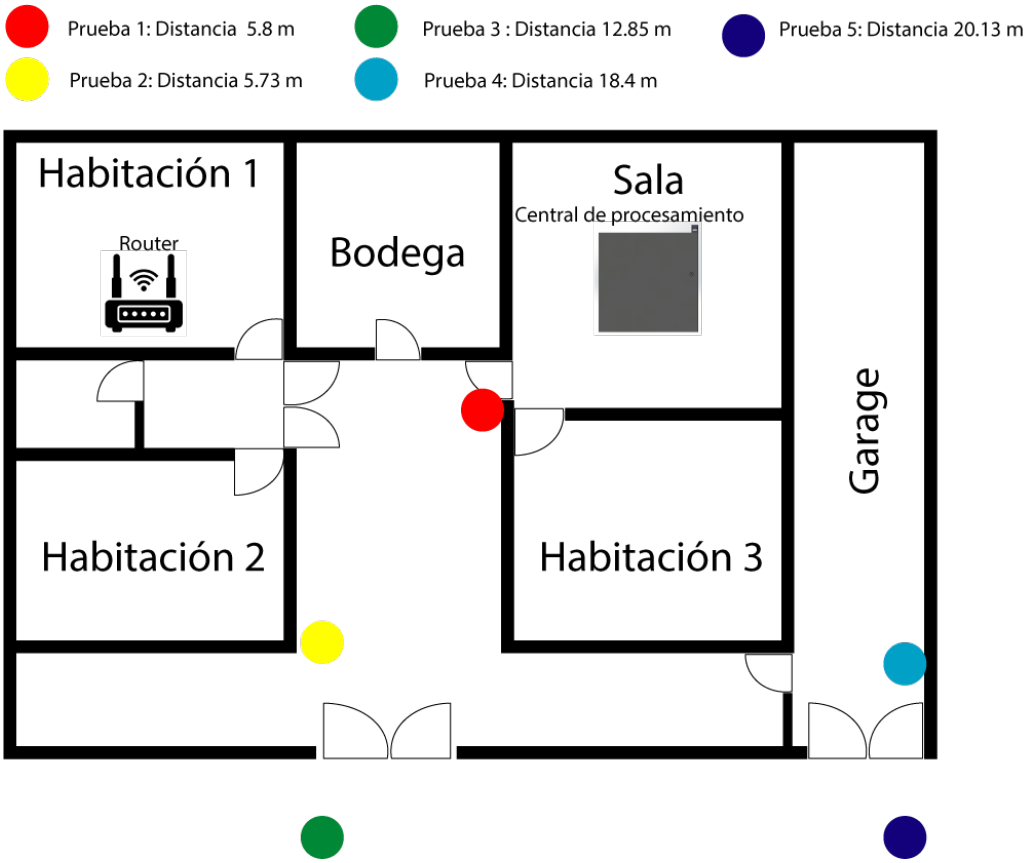


Fig. 4.28. Croquis

Mediante la observación de los mensajes impresos en consola por la Raspberry Pi Zero 2 W, es posible identificar el momento en que se produce la desconexión de un módulo sensor durante las pruebas de alcance. Esta información, como se evidencia en la Fig. 4.29, permite determinar la distancia máxima a la que el sensor mantiene una conexión estable con la red WiFi.

```
Sensor 4 está 'offline'. Actualizando estado...
Mensaje de historial enviado
Estado del sensor 4 actualizado en Firebase a: offline
Notificación enviada con éxito: projects/alarma-xibernetiq/messages/0:1744770621213887%b3e2c2c5b3e2c2c5
```

Fig. 4.29. Mensaje de desconexión del sensor.

En todas las ubicaciones señaladas realizadas se obtiene que la conexión se mantiene estable, permitiendo el envió de mensaje por MQTT desde los módulos sensores hacia la central de procesamiento, Sin embargo, en la ubicación correspondiente a 20.13 metros el módulo sensor registra una pérdida de conexión, evidenciada mediante el mensaje en consola de la Raspberry Pi Zero 2 W. De esta manera, se establece esa distancia como límite aproximado donde la comunicación es efectiva en las condiciones físicas del entorno evaluado.

4.6.2 Tiempo de reacción y estabilidad

Mediante la incorporación de marcas de tiempo en cada etapa del proceso —desde el envío del mensaje MQTT por parte del módulo sensor, el procesamiento del mensaje para la activación de la sirena, hasta el envío de la notificación a la aplicación móvil— es posible registrar datos precisos para el análisis de los tiempos de reacción del sistema. Estas marcas se generan en los distintos entornos de desarrollo utilizados (MicroPython, Python y React Native), tomando como referencia el Tiempo Universal Coordinado (UTC, Coordinated Universal Time), lo que permite una comparación consistente entre eventos.

Para lograr registrar las marcas de tiempo en la Raspberry Pi Pico W se vuelve necesario usar el Protocolo de Tiempo de Red (NTP, Network Time Protocol), debido la placa no cuenta con un Reloj en Tiempo Real (RTC, real-time clock). Por lo cual, se utiliza la librería **ntptime** [14], que permite obtener el tiempo UTC desde servidores de tiempo en línea y actualizar el reloj interno.

En el Algoritmo 1 se muestra la marca de tiempo que se coloca en la Raspberry Pi Pico W en el momento del envío del mensaje MQTT.

[14] librería ntptime:`https://mpython.readthedocs.io/en/v2.2.1/library/micropython/ntptime.html`

Alg.1. Algoritmo donde se ubica la marca de tiempo en Raspberry Pi Pico W (Micropython).

```
Leer el estado del sensor: estado ← 1 si sensor_apertura.value() = 1, de lo contrario 0
if estado = 1 then
    mensaje ← json.dumps({"sensor_id : sensor_id, .estado : estado})
    Imprimir .Apertura detectada, enviando mensaje: mensaje"
    Publicar en client.publish(f.alarma/sensor{sensor_id}/{tipo_sensor}, mensaje, qos = 1)
    timestamp ← time.time()
    fecha_utc ← time.gmtime(timestamp)
    Imprimir ÜTC: fecha_utc"
    Esperar 5 segundos: time.sleep(5)
else
    if current_time − last_ping_time ≥ keepalive_interval/2 then
        mensaje ← json.dumps({"sensor_id : sensor_id, .estado : estado})
        Publicar en client.publish(f.alarma/sensor{sensor_id}/{tipo_sensor}, mensaje, qos = 1)
        last_ping_time ← current_time
        Imprimir "Sin eventos detectados."
    else
        Imprimir "Sin eventos detectados durante el tiempo"
    end if
end if
```

En en Algoritmo 2 se muestra la marca de tiempo que se coloca en la Raspberry Pi Zero 2 W al realizar el procesamiento del mensaje recibido por MQTT, con el fin de activar la sirena y enviar la notificación.

Alg.2. Algoritmo donde se ubica la marca de tiempo en Raspberry Pi Zero 2 W (Python).

```
function activar_sirena
    Activar pin GPIO 17: GPIO.output(17, GPIO.HIGH)
    Imprimir 'UTC'
    now ← datetime.datetime.utcnow()
    current_time ← now.time().isoformat(timespec = 'seconds')
    Imprimir current_time
end function
```

En el Algoritmo 3 se muestra la marca de tiempo que se coloca en React Native al recibir la notificación por parte de FCM.

Alg.3. Algoritmo donde se ubica la marca de tiempo en aplicación móvil (Rect Native).

```
function handleForegroundNotifications
    Escuchar mensajes: messaging().onMessage(remoteMessage)
    Imprimir "Mensaje recibido en primer plano:"remoteMessage
    {notification, sentTime} ← remoteMessage
    if sentTime ≠ null then
        fecha ← new Date(sentTime)
        Imprimir "Marca de tiempo UTC:"fecha.toUTCString()
    end if
    authStatus ← await getAuthStatus()
    if notification ∧ authStatus = 'authenticated' then
        NotificationService.showNotification(remoteMessage.messageId ∨
"", notification.title ∨ 'Notificación', notification.body ∨ "")
    end if
end function
```

Una vez se activa el módulo sensor, se registran marcas de tiempo en los diferentes puntos del sistema. El envío del mensaje desde el entorno MicroPython se evidencia en la Fig. 4.30 (a), el procesamiento en Python en la Fig. 4.30 (b), y la notificación recibida en la aplicación móvil desarrollada en React Native se muestra en la Fig. 4.30 (c). Estas evidencias permiten realizar un análisis cuantitativo del tiempo de respuesta del sistema.

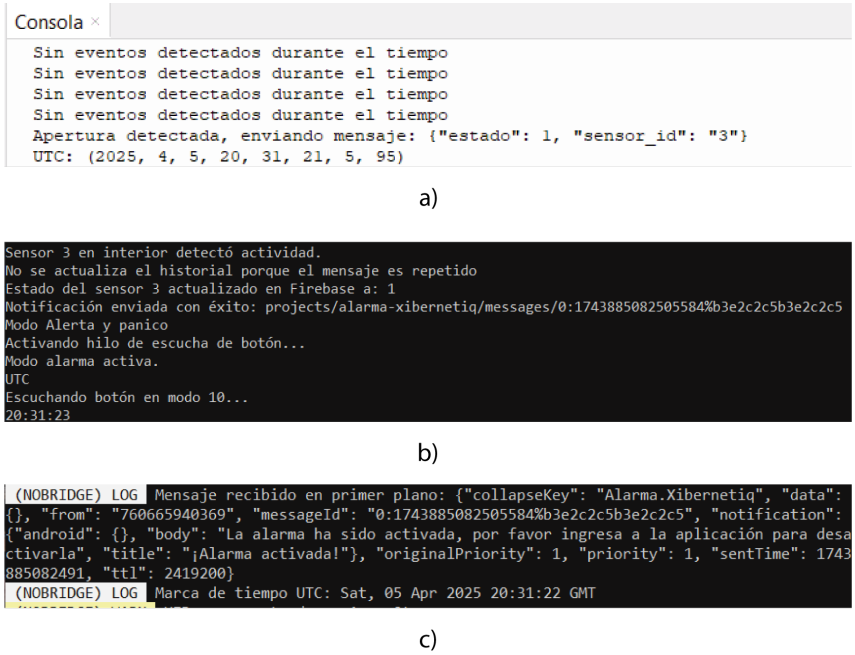


Fig. 4.30. Impresiones en consola de marcas de tiempo. (a) Marca de tiempo Raspberry pico W , (b) Marca de tiempo Raspberry Zero 2 W, (c) Marca de tiempo aplicación móvil.

Como se observa en la Tabla 4.10, los tiempos de reacción obtenidos mediante impresiones en consola, mantienen una similitud en todas las pruebas realizadas. Sin embargo, se obtiene que en promedio que la activación de la sirena ocurre entre 1.8 y 2.6 segundos después de la detección del evento por parte del módulo sensor, mientras que la notificación en la aplicación móvil se recibe y muestra entre 0.8 y 1.8 segundos. Estas cifras demuestran una respuesta rápida y adecuada del sistema ante eventos de seguridad.

Tabla 4.10

Tiempos de reacción a eventos

Prueba	Sensor (UTC)	Sirena (UTC)	Notificación App	T. Sirena (s)	T. App (s)
1	04:58:05.000	04:58:06.864	04:58:06.182	1.864	1.182
2	05:02:27.000	05:02:29.144	05:02:28.359	2.144	1.359
3	05:10:13.000	05:10:15.638	05:10:14.800	2.638	1.800
4	05:15:36.000	05:15:37.995	05:15:36.853	1.995	0.853
5	05:20:24.000	05:20:26.310	05:20:25.401	2.310	1.401

CONCLUSIONES

Se desarrolló un sistema de alarma de seguridad basado en tecnología IoT, adaptable a edificaciones residenciales con distintas características. El sistema integra módulos sensores distribuidos, comunicación inalámbrica, gestión remota a través de la nube y una aplicación móvil con interfaz moderna que permite al usuario configurar y supervisar el sistema de manera intuitiva.

Se determinaron parámetros y características fundamentales para una alarma de seguridad orientada a edificaciones residenciales centrándose tanto en aspectos internos como la detección pronta de amenazas y la gestión de la información, así como en aspectos finales como la fácil instalación, configuración y administración del sistema por parte del usuario.

Así mismo, el sistema de seguridad diseñado utilizando componente disponible en el mercado permite la configuración y gestión tanto de sensores como de la central de procesamiento de manera sencilla a través de la aplicación móvil. Además, se consideró la escalabilidad de sistema permitiendo que sea adaptable a diferentes áreas de una edificación.

Por otro lado, la arquitectura propuesta permite la gestión remota a través e la nube, donde se aprovecha los servicios de Firebase tanto para la comunicación entre dispositivo, almacenamiento de datos, notificación a la aplicación móvil y autentificación de los usuarios para acceder a la configuración y gestión de la alarma.

Finalmente, a partir de las pruebas realizadas, se demostró que el sistema de alarma presenta un buen desempeño en sus funcionalidades. Se observó una respuesta rápida frente a eventos reales, así como una notable estabilidad ante cambios en la red.

RECOMENDACIONES

Con la finalidad de reducir falsos positivos y mejorar el sistema de alarma se podrían implementar técnicas que permitan analizar patrones comunes en el comportamiento de los sensores ante diferentes escenarios.

Adicionalmente, la arquitectura propuesta puede ampliarse mediante la integración de tecnologías emergentes como inteligencia artificial o aprendizaje automático, permitiendo la toma de decisiones automatizada basada en el historial de eventos o condiciones del entorno.

Finalmente, se recomienda explorar el potencial del sistema dentro del campo de la domótica, facilitando la interacción del sistema de alarma con otros dispositivos inteligentes del hogar, como cerraduras electrónicas, cámaras IP o asistentes virtuales.

BIBLIOGRAFÍA

[1] M. Gonzales. "Ecuador lidera el incremento de violencia criminal en Latinoamérica." [En línea]. (ene. de 2023), dirección: https://www.primicias.ec/noticias/en-exclusiva/ecuador-incremento-muertes-violentas-latinoamerica/.

[2] G. Coba. "Ventas de empresas de seguridad crecen por escalada de violencia." [En línea]. (ago. de 2023), dirección: https://www.primicias.ec/noticias/economia/ventas-seguridad-ecuador-empresas-robos-secuestros/.

[3] P. Naula, "Más demanda en sistemas de seguridad y vigilancia," *El Mercurio*, abr. de 2023, [En línea]. dirección: https://www.elmercurio.com.ec/2023/04/26/demanda-sistemas-seguridad-vigilancia-cuenca/.

[4] J. Salazar y S. Silvestre, *Internet de las cosas*, [En línea], S.F. dirección: https://upcommons.upc.edu/bitstream/handle/2117/100921/LM08_R_ES.pdf.

[5] O. G. C. Meneses, "Implementación de un sistema de alarma para detección de incendios, en el edificio de la carrera de Ingeniería en Mantenimiento Eléctrico en el campus universitario El Olivo," [En línea], M.S. Thesis, Universidad Técnica del Norte, 2019. dirección: http://repositorio.utn.edu.ec/handle/123456789/9568.

[6] F. M. Moreno, "Diseño e implementación de un sistema de alarma IoT basada en tecnologías Open Source," [En línea], M.S. Thesis, Universidad Politécnica de Cartagena, 2019. dirección: https://repositorio.upct.es/entities/publication/81ada08e-c927-4412-86a3-64a9ffe1abd7.

[7] J. V. Cardenas, *Diseño y construcción de un sistema internet de las cosas para alertar el robo en viviendas*, B.S. Thesis, [En línea], 2022. dirección: https://apirepositorio.unh.edu.pe/server/api/core/bitstreams/e0d8da07-75d7-44ec-8555-5e1bc69dd657/content.

[8] J. A. R. Sandoval, *Diseño e implementación de un sistema de alarma de intrusión basado en el protocolo ESP-Now de internet de las cosas*, Tesis de grado, [En línea], 2021. dirección: http://repositorio.untels.edu.pe/jspui/handle/123456789/594.

[9] M. Guatapi y E. García, *Diseño e implementación de un prototipo para un sistema de medición, análisis y purificador de gases contaminantes en el aire utilizando Arduino y Ubidots IoTl*, Tesis, [En línea], 2022. dirección: http://dspace.ups.edu.ec/handle/123456789/24088.

[10] J. A. P. Herrera y J. A. L. Heredia, *Diseño e implementación de un prototipo escalable con infraestructura IoT para la medición y visualización de la calidad del aire usando tecnología open source para el edificio de la Carrera de Ingeniería en Networking y Telecomunicaciones*, Tesis, [En línea], 2020. dirección: http://repositorio.ug.edu.ec/handle/redug/49504.

[11] L. M. T. Peñafiel, *Diseño y análisis de prototipo de un sistema de seguridad con sensores de movimiento y cámaras IP de videovigilancia aplicando una infraestructura IOT para el envío y recepción de datos entre dispositivos*, Tesis, [En línea], 2022. dirección: http://repositorio.ug.edu.ec/handle/redug/59942.

[12] F. L. García Olivo, *Implementación de un sistema de seguridad con IoT para el Asadero Gilgal basado en un botón de pánico y alarma con módulo GPS*, Tesis, [En línea.], 2024. dirección: https://repositorio.upse.edu.ec/handle/46000/10931.

[13] D. A. V. Uvidia, *Implementación de un prototipo de sistema de seguridad doméstico basado en WPAN para una red IoT*, Tesis, [En línea], 2019. dirección: http://dspace.espoch.edu.ec/handle/123456789/13493.

[14] D. W. H. Chávez, *Diseño e implementación de un prototipo de seguridad para control domótico basado en IoT bajo ambientes de dispositivos móviles con Android*, Tesis, [En línea], Quito, Ecuador, 2020. dirección: https://bibdigital.epn.edu.ec/handle/15000/20857.

[15] MQTT.org, *MQTT - The Standard for IoT Messaging*, https://mqtt.org/, 2024.

[16] IBM, *Protocolo HTTP*, https://www.ibm.com/docs/es/cics-ts/5.5?topic=concepts-http-protocol, IBM.

[17] J. M. Huidobro, "Wi-Fi. Conectividad en todo lugar y momento," Spanish, *Revista Digital de Manuales Formativos*, n.º 035, pág. 9, 2005. dirección: https://www.acta.es/recursos/revista-digital-manuales-formativos/272-035.

[18] G. Cloud, *Firebase Realtime Database*, https://firebase.google.com/docs/database?hl=es-419, Google.

[19] Google, *Firebase Authentication*, https://firebase.google.com/docs/auth?hl=es-419, Google.

[20] Google, *Firebase Cloud Messaging*, https://firebase.google.com/docs/cloud-messaging?hl=es-419, Google.

[21] Python Software Foundation, *Python Programming Language*, https://www.python.org/, Python.

[22] React Native Community, *React Native*, https://reactnative.dev/, React Native.

[23] TypeScript Language Specification, *TypeScript para el nuevo programador*, https://whttps://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html, TypeScript.

[24] J. Lozada, "Investigación Aplicada: Definición, Propiedad Intelectual e Industria," *Dialnet*, dic. de 2014. dirección: https://dialnet.unirioja.es/servlet/articulo?codigo=6163749.

[25] L. Reyes-Ruiz y F. Carmona, *La investigación documental para la comprensión ontológica del objeto de estudio*, Repositorio Institucional, 2020. dirección: https://hdl.handle.net/20.500.12442/6630.

[26] F. d. C. P. y. S. UNAM, *Introducción a la Investigación: guía interactiva*, Universidad Veracruzana, 2017. dirección: https://www.uv.mx/apps/bdh/investigacion/unidad1/investigacion-tipos.html.

[27] C. Ramos-Galarza, "Diseños de investigación experimental," *CienciAmérica*, vol. 10, n.º 1, ene. de 2021, Editorial, issn: 1390-9592. doi: 10.33210/ca.v10i1.356.

[28] Raspberry Pi, *Raspberry Pi Zero 2 W*. dirección: https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/.

[29] Grupo Electrostore, *Fuente de Poder (voltaje) 220/110v a 12V 10A SC120W-12 Función Ups Salida Ajustable (no date a)*. dirección: https://grupoelectrostore.com/shop/fuentes-cargadores-y-adaptadores-de-voltaje/fuente-de-poder-voltaje-220-110v-a-12v-10a-sc120w-12-funcion-ups-salida-ajustable/.

[30] Grupo Electrostore, *MÓDULO REDUCTOR DE VOLTAJE 3A LM2596 AJUSTABLE STEP DOWN BUCK*. dirección: https://grupoelectrostore.com/shop/modulos-y-shields/reguladores-de-voltaje/modulo-reductor-de-voltaje-3a-lm2596-ajustable-step-down-buck/.

[31] UNIT Electronics, *Sensor ir HX1838 con control inalámbrico (2025)*. dirección: https://uelectronics.com/producto/kit-sensor-ir-hx1838-con-control-inalambrico/?srsltid=AfmBOoqsv4GxCZR3HjfPPLVz55U1MUucqGvl8_6mZWwRKuE4SvoygY5n.

[32] Grupo Electrostore, *Módulo Relé 5v 4 canales con optoacopladores high/low alto/bajo (rojo)*. dirección: https://grupoelectrostore.com/shop/modulos-y-shields/modulos-rele/modulo-rele-5v-4-canales-con-optoacopladores-high-low-alto-bajo-rojo/.

[33] Raspberry Pi, *Pico Series. Raspberry Pi Documentation*. dirección: https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html#pico-1-family.

[34] Grupo Electrostore, *Módulo TP4056 Cargador Baterías Litio Con Protección*. dirección: https://grupoelectrostore.com/shop/modulos-y-shields/cargadores-para-baterias/modulo-tp4056-5v-micro-usb-1a-cargador-para-baterias-litio-con-proteccion/.

[35] UNIT Electronics, *MT3608 Elevador de Voltaje Boost Step up 6W 2A (2025)*. dirección: https://uelectronics.com/producto/modulo-regulador-mt3608-step-up-5-28v-2a/?srsltid=AfmBOorUy2v6roFuqJGos3T191BbynqTfh80A9SY9MByLz4Qg3n9QD7M.

[36] Naylamp Mechatronics - Perú, *Módulo Sensor PIR HC-SR501*. dirección: https://naylampmechatronics.com/sensores-proximidad/55-modulo-sensor-pir-hc-sr501.html.

[37] UNIT Electronics, *SW-420 módulo sensor de Vibración (2025)*. dirección: https://uelectronics.com/producto/sw-420-modulo-sensor-de-vibracion/?srsltid=AfmBOorsErdEglxnAVaG9aV7n1cl1bdyJ9tqYtEfJ3qKkpqX8nk97HDE.

[38] Electronilab, *Sensor Interruptor Magnético de Puerta o Ventana MC-31 - Electronilab*. dirección: https://electronilab.co/tienda/sensor-interruptor-magnetico-de-puerta-o-ventana/.

[39] Naylamp Mechatronics - Perú, *Sensor de gas MQ2*. dirección: https://naylampmechatronics.com/sensores-gas/71-sensor-mq-2-gas-glp-gnv.html.

ANEXOS

Códigos y scripts disponibles en: `https://github.com/Marlon4910/TesisAlarmaIoT`