



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE INGENIERÍA INDUSTRIAL

Tema:

**SISTEMA DE CONTROL Y MONITORIZACIÓN DE ALARMAS
COMUNITARIAS PARA EL TALLER DE INNOVACIÓN Y DESARROLLO
TECNOLÓGICO FISEI**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero Industrial

ÁREA: Automatización

LÍNEA DE INVESTIGACIÓN: Diseño, materiales, producción, identidad,
sostenibilidad y tecnologías aplicadas

AUTOR: Marco Fabricio Tacoamán Yauli

TUTOR: Ing. Franklin Wilfrido Salazar Logroño, Mg.

Ambato - Ecuador

febrero - 2025

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA DE CONTROL Y MONITORIZACIÓN DE ALARMAS COMUNITARIAS PARA EL TALLER DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO FISEI, desarrollado bajo la modalidad Proyecto de Investigación por el señor Marco Fabricio Tacoamán Yauli, estudiante de la Carrera de Ingeniería Industrial, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2025.



Ing. Franklin Wilfrido Salazar Logroño, Mg.
TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA DE CONTROL Y MONITORIZACIÓN DE ALARMAS COMUNITARIAS PARA EL TALLER DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO FISEI es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2025.



Marco Fabricio Tacoamán Yauli

C.C. 180515668-2

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2025.



Marco Fabricio Tacoamán Yauli

C.C. 180515668-2

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Marco Fabricio Tacoamán Yauli, estudiante de la Carrera de Ingeniería Industrial, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE CONTROL Y MONITORIZACIÓN DE ALARMAS COMUNITARIAS PARA EL TALLER DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO FISEI, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, febrero 2025.



Firmado electrónicamente por:
**FRANKLIN OSWALDO
MAYORGA MAYORGA**

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

PRESIDENTE DEL TRIBUNAL



Firmado electrónicamente por:
**MAURICIO XAVIER
LOPEZ FLORES**

Ing. Mauricio Xavier López Flores, Mg.

PROFESOR CALIFICADOR



Firmado electrónicamente por:
**JAIME BOLIVAR RUIZ
BANDA**

Ing. Jaime Bolívar Ruiz Banda, Mg.

PROFESOR CALIFICADOR

DEDICATORIA

A mi madre Lida, quien fue y siempre será mi inspiración. Su ejemplo de fortaleza y dedicación me enseñó que no hay límites cuando se lucha con perseverancia por los sueños.

A mi padre Marco, que aunque físicamente ya no está con nosotros, su legado vive en cada enseñanza y consejo que nos dio.

A mis hermanas Paulina y Maribel, quienes mediante sus consejos me expresaban su apoyo incondicional en cada etapa de mi vida.

A mis compañeros y amigos con los cuales se compartieron momentos amenos durante la carrera.

AGRADECIMIENTO

En primer lugar a Dios por haberme permitido culminar esta etapa más de mi vida.

A mi madre por haberme apoyado financieramente a lo largo de la carrera.

Al ingeniero Franklin Salazar, quien, además de desempeñarse como mi tutor, se convirtió en un guía constante a lo largo de este proceso de titulación. Su apoyo continuo, su generosidad al compartir sus conocimientos y su valiosa orientación a través de sus consejos han sido pilares fundamentales en mi desarrollo académico y personal.

A los ingenieros Edwin y José por su apoyo y colaboración durante el desarrollo de esta tesis. Su experiencia, orientación y la disposición con la que siempre compartieron sus conocimientos hicieron una gran diferencia en cada paso del camino.

A todos los docentes y amigos con quienes tuve el honor de compartir, expreso mi más sincero agradecimiento por haberme transmitido sus conocimientos, que se convirtieron en enseñanzas valiosas para la vida.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS.....	xiv
ÍNDICE DE ANEXOS	xvii
RESUMEN EJECUTIVO	xviii
ABSTRACT	xix
CAPÍTULO I. MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	5

1.3.1 Seguridad electrónica	5
1.3.2 Alarmas	5
1.3.3 Sistemas de alarmas	6
1.3.4 Sistema de alarmas comunitarias	6
1.3.5 Tipos de alarmas comunitarias.....	8
1.3.6 Especificaciones funcionales de un sistema de alarmas comunitarias.....	8
1.3.7 Propósito de los sistemas de alarmas comunitarias.....	9
1.3.8 Protocolos de comunicación de los sistemas de alarmas comunitarias.....	9
1.3.9 Características de los protocolos de comunicación.....	13
1.3.10 Redes inalámbricas.....	14
1.3.11 IoT (Internet de las Cosas)	15
1.3.12 Arquitectura IoT.....	16
1.3.13 Desafíos del IoT	17
1.3.14 LoRa.....	17
1.4 Objetivos	18
1.4.1 Objetivo general.....	18
1.4.2 Objetivos específicos	18
CAPÍTULO II. METODOLOGÍA	19
2.1 Materiales.....	19
2.2 Métodos.....	22

2.2.1 Modalidad de la investigación	22
2.2.2 Recolección de información.....	26
2.2.3 Procesamiento y análisis de datos	27
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	29
3.1 Análisis y discusión de resultados.....	29
3.1.1 Análisis de los sistemas de alarmas comunitarias	29
3.2 Desarrollo e implementación del sistema de alarmas	35
3.3 Primera fase.....	37
3.3.1 Diseño y construcción de placas electrónicas para el sistema.	37
3.3.2 Botón físico de activación.....	41
3.3.3 Batería LiPo para respaldo de alimentación del sistema.....	45
3.3.4 Dispositivo de notificación “Sirena”	49
3.3.5 Placa electrónica.....	53
3.3.6 Construcción de la placa a partir del diseño PCB	55
3.3.7 Carcasa de alojamiento	56
3.3.8 Señalética de seguridad	57
3.4 Segunda fase.....	58
3.4.1 Conexión eléctrico de los componentes físicos del sistema.....	58
3.4.2 Flujograma del sistema	63
3.5 Tercera fase	65

3.5.1 Programación y aplicación web	65
3.6 Cuarta fase.....	80
3.6.1 Pruebas de desempeño y validación del sistema de alarmas.....	80
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	94
4.1 Conclusiones	94
4.2 Recomendaciones.....	95
REFERENCIAS BIBLIOGRÁFICAS	97
ANEXOS	101

ÍNDICE DE TABLAS

Tabla 1. Protocolos de comunicación.	11
Tabla 2. Recursos materiales empleados en el sistema.	19
Tabla 3. Softwares empleados para el sistema.	21
Tabla 4. Preguntas de investigación.	23
Tabla 5. Términos de búsqueda.	23
Tabla 6. Criterios de búsqueda.	24
Tabla 7. Recolección de la información.	26
Tabla 8. Resultados de investigación.	29
Tabla 9. Características principales de las placas ESP 32.	37
Tabla 10. Descripción de los factores para la selección de la placa de desarrollo.	38
Tabla 11. Escala de relevancia para la evaluación de criterios.	39
Tabla 12. Ponderación de factores para placas ESP 32.	39
Tabla 13. Escala de evaluación para clasificar los factores de ponderación.	40
Tabla 14. Comparación de las placas de desarrollo ESP 32.	40
Tabla 15. Características de los botones de emergencia.	42
Tabla 16. Descripción de los factores para la selección del botón.	43
Tabla 17. Ponderación de factores para el botón.	43
Tabla 18. Comparación de los botones de emergencia.	44
Tabla 19. Características de las baterías LiPo.	46
Tabla 20. Descripción de los factores para la selección de la batería.	46

Tabla 21. Ponderación de factores para la batería de respaldo.	47
Tabla 22. Comparación cuantitativa entre las baterías de respaldo.	48
Tabla 23. Características de las sirenas.....	49
Tabla 24. Descripción de los factores para la selección de la sirena.	50
Tabla 25. Ponderación de factores para la sirena.	51
Tabla 26. Comparación cuantitativa entre los tipos de sirenas.	52
Tabla 27. Descripción de las entradas físicas de la placa.	60
Tabla 28. Descripción de las salidas físicas de la placa.	61
Tabla 29. Descripción de los pines GPIO.	62
Tabla 30. Simbología utilizada.	66
Tabla 31. Población conformada por el taller.	81
Tabla 32. Pruebas de funcionamiento del sistema de alarmas.	82
Tabla 33. Cálculo del error absoluto y relativo de las pruebas.	84
Tabla 34. Tiempo de retraso en la activación del sistema.....	86
Tabla 35. Cálculo del tiempo promedio en el retraso del sistema.	88
Tabla 36. Descripción de los factores para la validación del sistema.	90
Tabla 37. Escala de relevancia para la evaluación de los criterios de validación.	90
Tabla 38. Características principales de los sistemas de alarmas.	91
Tabla 39. Ponderación de los factores para la validación del sistema.	91
Tabla 40. Comparación de los sistemas de alarmas.	92

ÍNDICE DE FIGURAS

Figura 1. Estructura de un sistema de seguridad electrónica	5
Figura 2. Sistema de alarma comunitaria en hogares.....	7
Figura 3. Internet de las cosas.	16
Figura 4. Arquitectura IoT.	16
Figura 5. Diagrama de flujo de la Metodología PRISMA.	25
Figura 6. Esquema general del sistema.	36
Figura 7. Botones físicos de activación.....	41
Figura 8. Batería para el respaldo de energía.	45
Figura 9. Sirena física de notificaciones.	49
Figura 10. Esquema del conexionado de la placa base.	53
Figura 11. Diseño PCB de la placa.	54
Figura 12. Visualización 3D de la placa.	55
Figura 13. Placa final elaborada.....	56
Figura 14. Carcasa de alojamiento de las placas.....	57
Figura 15. Señalética de seguridad.....	57
Figura 16. Diagrama de conexionado eléctrico.....	59
Figura 17. Diagrama de conexionado entradas físicas.....	60
Figura 18. Diagrama de conexionado salidas físicas.	61
Figura 19. Diagrama de conexionado de los pines GPIO.	62
Figura 20. Diagrama de flujo del sistema.	64

Figura 21. Programa para la detección de alarmas.	65
Figura 22. Definición de los tópicos para la comunicación MQTT.....	65
Figura 23. Configuración de la pantalla OLED.	66
Figura 24. Desactivación remota de los pines para el control de la sirena.	66
Figura 25. Control en los cambios de las entradas digitales.	67
Figura 26. Impresión de los estados en la pantalla OLED.	67
Figura 27. Configuraciones necesarias para conectarse con la base de datos.....	68
Figura 28. Obtención de las coordenadas desde la BD.....	68
Figura 29. Control de las coordenadas de activación.....	69
Figura 30. Credenciales de acceso para a comunicación de los sensores.	69
Figura 31. Actualización de los tópicos como activo o inactivo.....	70
Figura 32. Generar nuevas alarmas en la base de datos.....	70
Figura 33. Credenciales de acceso para a comunicación de las sirenas.....	71
Figura 34. Publicación de mensajes en los tópicos.	71
Figura 35. Actualización de los estados de la sirena.....	72
Figura 36. Envío de los estados a los tópicos.....	72
Figura 37. Herramientas para habilitar el servidor.....	73
Figura 38. Seguridad de la aplicación y solicitudes de otros dominios.	73
Figura 39. Credenciales de la conexión MySQL.	73
Figura 40. Lista de alertas generadas en base a la tabla mencionada.	73
Figura 41. Registro de una alerta en la tabla de alarmas.....	74

Figura 42. Procesamiento de imágenes para guardarlas en la BD.	74
Figura 43. Aplicación botón de pánico.	75
Figura 44. Alarma detallada.	76
Figura 45. Listado de las posibles alerta a generarse.	77
Figura 46. Apartado para enviar información adicional.	77
Figura 47. Apartado para enviar fotos o imágenes.	78
Figura 48. Interfaz de monitoreo de las alertas generadas.	79
Figura 49. Interfaz de monitoreo “ubicación de la alerta generada”.....	79
Figura 50. Notificación de la situación generada.	80
Figura 51. Porcentaje de error en la activación del sistema.	85
Figura 52. Características del servidor virtual.	89

ÍNDICE DE ANEXOS

Anexo A. Código completo para la detección de las alarmas generadas.....	101
Anexo B. Código completo para la localización y georreferencia.....	103
Anexo C. Geolocalización del Taller de Innovación y Desarrollo Tecnológico.	104
Anexo D. Latitud y longitud de los 4 puntos del polígono del taller.	105
Anexo E. Código completo para los sensores.....	106
Anexo F. Código completo para la activación de la sirena.....	108
Anexo G. Código para el servidor lógico.....	109
Anexo H. Código para la generación de mensajes en Telegram.....	112
Anexo I. Esquema de correlación entre las diferentes tablas.....	113
Anexo J. Tendido del cable para la conexión con los botones y la sirena.	113
Anexo K. Disposición final de la carcasa y los componentes.....	114
Anexo L. Manual de operación y funcionamiento del sistema.....	115

RESUMEN EJECUTIVO

El presente proyecto investigativo desarrolló un piloto para un sistema de control y monitorización de alarmas comunitarias para el Taller de Innovación y Desarrollo Tecnológico FISEI de la Universidad Técnica de Ambato. El sistema integra múltiples dispositivos conectados mediante tecnología IoT, diseñado para emitir notificaciones de alertas de forma rápida, precisa y eficiente. Esta solución aborda problemas fundamentales, como retrasos en la transmisión de información y la falta de coordinación en situaciones críticas, mejorando significativamente la capacidad de respuesta ante emergencias y fortaleciendo la seguridad comunitaria.

El sistema incluye una interfaz intuitiva, accesible tanto desde dispositivos móviles como computadoras, que permite a los usuarios gestionar alarmas, consultar el historial de eventos y activar funciones clave, como el botón de pánico o la notificación de alarmas detalladas.

Para garantizar la fiabilidad y velocidad de las notificaciones, se emplearon protocolos de comunicación basados en MQTT. Las pruebas realizadas evidenciaron un margen de error aproximado del 9%, mostrando que el sistema responde de manera efectiva en 9 de cada 10 activaciones, ya sea mediante botones físicos, la aplicación móvil o mediante la estación contra incendios y detección de humo.

En conclusión, este sistema representa una solución escalable y eficiente, con el potencial de integrarse en proyectos de mayor relevancia, contribuyendo a la seguridad y coordinación en escenarios de emergencia.

Palabras clave: Integración de tecnologías, Esp32, MQTT, monitoreo, internet de las cosas, interfaz, escalabilidad, fiabilidad.

ABSTRACT

This research project developed a pilot for a community alarm control and monitoring system for the Innovation and Technological Development Workshop FISEI of the Technical University of Ambato. The system integrates multiple devices connected through IoT technology, designed to issue alert notifications quickly, accurately and efficiently. This solution addresses fundamental problems, such as delays in the transmission of information and lack of coordination in critical situations, significantly improving emergency response capacity and strengthening community safety.

The system includes an intuitive interface, accessible from both mobile devices and computers, which allows users to manage alarms, consult event history and activate key functions, such as the panic button or detailed alarm notification.

To ensure the reliability and speed of notifications, communication protocols based on MQTT were used. Tests evidenced a margin of error of approximately 9%, showing that the system responds effectively in 9 out of 10 activations, either via physical buttons, the mobile app or via the fire and smoke detection station.

In conclusion, this system represents a scalable and efficient solution, with the potential to be integrated in more relevant projects, contributing to security and coordination in emergency scenarios.

Keywords: Technology integration, Esp32, MQTT, Security, monitoring, internet of things, interface, scalability, reliability.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA DE CONTROL Y MONITORIZACIÓN DE ALARMAS COMUNITARIAS PARA EL TALLER DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO FISEI.

1.1.1 Planteamiento del problema

A nivel mundial, los sistemas de seguridad, control y monitoreo de eventualidades cotidianas se han convertido en una prioridad creciente para la sociedad. Cada vez más personas y organizaciones, tanto públicas como privadas, reconocen la importancia de contar con sistemas eficientes que permitan la detección y divulgación rápida de contingencias. En muchas regiones, se están implementando redes de comunicación para alertas de emergencias, diseñadas para difundir información crítica de manera ágil y oportuna, lo que facilita una respuesta rápida ante situaciones imprevistas [1].

La inseguridad ha aumentado de manera alarmante en diversos entornos, como: barrios, centros comerciales, parques, instituciones financieras, medios de comunicación, y especialmente en instituciones educativas. Este fenómeno se atribuye, en gran parte, al elevado índice de migrantes que, por necesidad, se han visto obligados a abandonar sus países de origen. En múltiples localidades, las unidades educativas presentan notables deficiencias en la gestión de situaciones de inseguridad, ya que a menudo carecen del equipo adecuado para la detección y comunicación eficiente de dichos eventos [2].

En Ecuador, el ámbito educativo enfrenta una creciente demanda de personas que buscan formarse en instituciones académicas debido a la alta competitividad del mercado laboral actual. Sin embargo, durante su formación, los estudiantes se ven expuestos constantemente a diversas situaciones de inseguridad tanto dentro como fuera de los centros educativos, lo que ha llevado a un aumento en la tasa de accidentes [3].

La ausencia de un sistema centralizado de alarmas comunitarias dificulta que la universidad y su comunidad accedan de manera inmediata y eficiente a información crítica sobre eventos que ocurren diariamente en el campus educativo. Esta carencia dificulta la toma de decisiones en circunstancias de emergencia que requieren atención inmediata, ya que retrasa la respuesta y coordinación necesaria para gestionar estos incidentes de manera eficiente y oportuna [4].

Si bien la implementación de un sistema de alarmas comunitarias podría ser una solución, este enfrenta desafíos significativos, como altos costos operativos, compatibilidad con otros sistemas y mantenimiento correctivos a largo plazo. Esto resalta la necesidad de diseñar y desarrollar un sistema de alarmas comunitarias que aborde estas problemáticas, proporcionando una solución más eficiente y sostenible en el tiempo, garantizando la seguridad y el monitoreo continuo sin incurrir en elevados costos operativos. Por tal motivo se implementará un sistema de control y monitorización de alarmas comunitarias en el Taller de Innovación y Desarrollo Tecnológico FISEI de la Universidad Técnica de Ambato.

1.2 Antecedentes investigativos

Los sistemas de alarmas se remontan a miles de años atrás, donde la protección se basaba en métodos simples como el uso de perros guardianes y armas letales. Hoy, la seguridad se ha vuelto extremadamente elaborada gracias a los avances tecnológicos y las innovaciones, permitiendo conocer los sucesos en tiempo real. Tras llevar a cabo una exhaustiva revisión bibliográfica a nivel nacional e internacional, se han identificado investigaciones afines al tema de estudio que servirán como pilares fundamentales para el desarrollo del trabajo. Entre estas investigaciones, se destacan:

Haz et al., en su estudio realizado en 2020, desarrollaron un “ Prototipo de alarma comunitaria inteligente para monitoreo de eventos e incidentes de seguridad ciudadana” con el objetivo integrar el uso de la tecnología en diferentes servicios que brinden una solución práctica y confiable ante escenarios de riesgos. La principal contribución del estudio realizado fue facilitar la comunicación de eventos e incidentes de seguridad entre los ciudadanos y los organismos de socorro. Todo ello en base al

uso de una aplicación móvil, placas Arduino y módulos GSM (Sistemas globales para telefonía móvil) – GPRS (Servicio general de radio por paquetes) [5].

En el año 2020, Cuadrado, desarrolla un “Prototipo de sistema integrado de seguridad comunitaria con tecnología Plug and Play a bajo costo”, diseñado para mejorar la seguridad en barrios vulnerables mediante una solución accesible y de fácil implementación. El sistema permitió a los habitantes alertar a la comunidad y a la Unidad de Policía Comunitaria (UPC) en caso de emergencia. Para ello, incorporó una aplicación móvil basada en Android, equipada con un botón de pánico que, al activarse, envía notificaciones en tiempo real a vecinos y autoridades, utilizando geolocalización para identificar con precisión el lugar del incidente. La infraestructura del sistema estuvo compuesta por una cámara IP, un servidor en la nube, una base de datos Firebase y tecnología Plug and Play, lo que facilita su instalación y uso sin requerir conocimientos técnicos avanzados [6].

Chicaiza, en su estudio realizado en 2020, diseña un “Sistema de alarma comunitaria para el mercado San Juan de la ciudad Santiago de Píllaro”, con el objetivo de brindar una solución eficaz ante emergencias dentro de la edificación. El prototipo fue diseñado utilizando una Raspberry Pi 3B+ como unidad central de procesamiento, operando en dos modos según los horarios de apertura del mercado. El sistema incorporó las aplicaciones de mensajería instantánea Telegram y WhatsApp para el envío de alertas visuales y auditivas a los usuarios y unidades policiales. Su desarrollo se llevó a cabo en el lenguaje de programación Python, utilizando una estructura basada en objetos, clases y funciones para gestionar tanto la activación de alarmas como el sistema de videovigilancia. La monitorización de las cámaras fue realizado a través de una página web, permitiendo una supervisión eficiente del entorno [7].

Chen et al., en el 2021, desarrollan un “Sistema de seguridad comunitaria basada en blockchain con dispositivos seguros de IoT”, enfocados para proteger a los residentes de edificios en varios peligros, como robos, incendios, inundaciones, terremotos y emergencias médicas. Los dispositivos IoT se colocaron en áreas públicas y privadas, y generan alertas en caso de incidentes. El sistema procesa estas alertas en múltiples fases y utiliza blockchain para almacenar los registros, garantizando que los datos sean seguros y no puedan ser alterados. Además, implementaron medidas de seguridad para

prevenir la manipulación de mensajes, interceptaciones y ataques. El esquema busca mejorar la seguridad comunitaria y prevenir conflictos innecesarios [8].

En el año 2022, Yanchatipán, desarrolla un “sistema de seguridad integral para la monitorización de alarmas y alerta de emergencia en la parroquia San Andrés del cantón Píllaro de la provincia de Tungurahua”, con el fin de combatir el creciente índice de delincuencia en la zona. Este sistema incorporó un módulo de geolocalización que permitió a los usuarios enviar alertas de emergencia o robo mediante dispositivos IoT conectados de manera inalámbrica a un servidor en la nube. Además, incluyó un botón de pánico inalámbrico para cada usuario, una base de datos en MariaDB administrada por una Raspberry Pi 4 para registrar a los moradores, y una interfaz en Python para la visualización en tiempo real del sistema y su funcionamiento [9].

El proyecto “Sistema de monitoreo de hogares inteligentes que utiliza microcontroladores ESP32” de IntechOpen es un claro ejemplo de la integración de diversos componentes electrónicos con el objetivo común de garantizar la seguridad de las personas. El sistema propuesto asegura la protección del hogar al detectar intrusos en el edificio, activar una alarma y capturar imágenes mediante una cámara. Posteriormente, emplean el estándar de comunicación MQTT para interactuar con la base de datos y enviar el informe correspondiente al propietario del inmueble [10].

Estas investigaciones y proyectos evidencian la aplicación de sistemas de control y monitoreo de alarmas comunitarias en diversos entornos sociales, destacando su capacidad para adaptarse a las necesidades específicas de cada comunidad. Los sistemas no solo garantizan una respuesta rápida ante incidentes, sino que también fomentan la participación ciudadana en la vigilancia de su entorno. Además, se mantiene una sólida integración de la tecnología mediante el uso de lenguajes y estándares, como SQL, para el almacenamiento, manipulación y recuperación de bases de datos, con el fin de gestionar la información de manera eficiente y oportuna.

1.3 Fundamentación teórica

1.3.1 Seguridad electrónica

Un sistema de seguridad electrónica tiene como objetivo proteger a las personas, instalaciones y bienes materiales tanto en entornos empresariales como en hogares. Este tipo de sistema integra diversos dispositivos tecnológicos que trabajan de manera interconectada, siendo gestionados desde una central de alarmas. Cuando se detecta una situación de riesgo, como una intrusión, incendio o emergencia, el sistema activa automáticamente diferentes mecanismos de respuesta. Entre estos elementos se incluyen sirenas, cámaras de vigilancia, luces de emergencia, y sistemas de notificación [11].

En la Figura 1 se muestra el esquema general de este tipo de sistema.



Figura 1. Estructura de un sistema de seguridad electrónica [11]

1.3.2 Alarmas

Las alarmas son dispositivos esenciales de seguridad que funcionan como herramientas de protección y disuasión en el entorno urbano. Originalmente diseñadas para la defensa de los hogares, su función se ha expandido al espacio público, convirtiéndose en una parte integral de la infraestructura de la seguridad comunitaria [12].

1.3.3 Sistemas de alarmas

Los sistemas de alarma están compuestos por diversos dispositivos eléctricos que, de manera conjunta, tienen la capacidad de detectar y registrar eventos relacionados con situaciones anormales o de riesgo. Estos sistemas se utilizan para alertar a estudiantes, pasajeros y empleados en estaciones de tren, aeropuertos, centros logísticos, así como en edificios administrativos y educativos, sobre posibles peligros. De esta forma, se les indica que deben tomar medidas de seguridad y ponerse a salvo [13].

1.3.4 Sistema de alarmas comunitarias

El sistema de alarmas comunitarias es una innovación tecnológica diseñada para mejorar la seguridad en áreas urbanas y rurales, proporcionando una herramienta efectiva tanto para las fuerzas del orden como para los ciudadanos. Estos sistemas permiten una vigilancia activa y colaborativa entre los residentes y las autoridades, facilitando la detección temprana de incidentes como robos, vandalismo, incendios o emergencias médicas. Los principales componentes que forman parte en la mayoría de estos sistema son [12]:

a. Sensores de movimiento

Dispositivos diseñados para detectar cambios en el entorno físico que indican la presencia o movimiento de objetos o personas.

b. Sensores de humo

Conocidos como detectores de humo, son dispositivos esenciales en la prevención de incendios, diseñados para detectar la presencia de humo, que es una indicación temprana de fuego. Juegan un papel crucial en sistemas de seguridad y protección contra incendios [12].

c. Cámaras de seguridad

Dispositivos utilizados para monitorear, grabar y, en algunos casos, transmitir en tiempo real eventos que ocurren en un área específica.

d. Sirenas y Alarmas Sonoras

Dispositivos diseñados para emitir sonidos fuertes y penetrantes con el objetivo de alertar a las personas sobre una situación de emergencia o peligro.

e. Botones de pánico

Dispositivos de seguridad diseñados para permitir a una persona activar rápidamente una alarma o notificación de emergencia en situaciones de peligro o crisis. Su principal objetivo es proporcionar una forma sencilla y rápida de alertar a las fuerzas de seguridad, a los servicios de emergencia o a otros individuos cercanos sobre una situación que requiera atención inmediata [12].

f. Aplicaciones móviles y sistemas de notificaciones

Herramientas tecnológicas que permiten a los usuarios gestionar y recibir alertas sobre situaciones de emergencia o eventos relevantes en tiempo real.

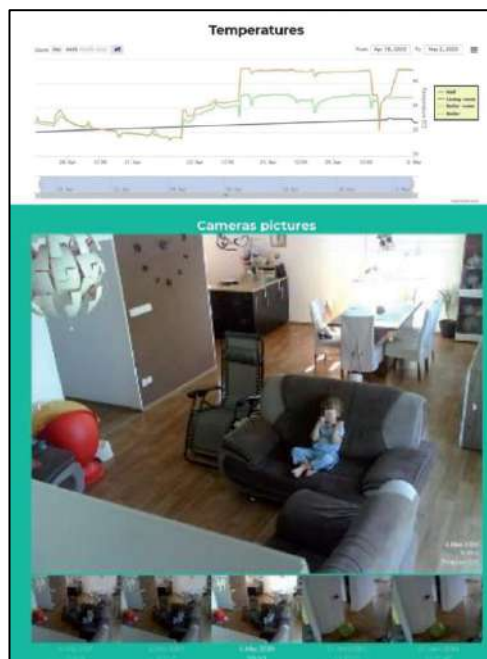


Figura 2. Sistema de alarma comunitaria en hogares [10].

1.3.5 Tipos de alarmas comunitarias

Las alarmas comunitarias pueden clasificarse en varios tipos, dependiendo de su función, la forma de comunicación a la central de monitoreo, la tecnología empleada y su propósito, a continuación se presentan diferentes tipos:

a. Con conexión a la central

Estos sistemas transmiten una señal de alarma a una instalación externa que proporciona servicios de seguridad, permitiendo que se envíen notificaciones pertinentes a las autoridades, como la policía, bomberos, emergencias, entre otros [14].

b. Sin conexión a la central

En este caso, la alarma se difunde a toda la comunidad a través de dispositivos sonoros y visuales, utilizando servicios de mensajería móvil para alertar a los residentes.

1.3.6 Especificaciones funcionales de un sistema de alarmas comunitarias

En los entornos donde se implementan alarmas, se identifican diversas características funcionales necesarias para alcanzar su propósito. Estas características son las siguientes:

a. Disuasiva

La función disuasiva tiene como objetivo prevenir o interrumpir actividades delictivas al informar a los delincuentes que el área cuenta con un sistema de seguridad. Esto provoca una reacción por parte de las autoridades y la comunidad, generando un impacto psicológico que desanima la conducta delictiva, empleando dispositivos como sirenas, reflectores y señales de advertencia [14].

b. Accesibilidad

La accesibilidad se refiere a los métodos disponibles para activar el sistema de alarma, facilitando así una respuesta rápida ante la detección de una amenaza. Esta accesibilidad puede variar según si la activación se realiza directamente en el panel de

control o mediante botones inalámbricos asignados a los miembros del equipo de seguridad [14].

c. De Difusión

La difusión puede ocurrir en dos niveles: el interno, que informa a la comunidad sobre un evento de inseguridad, y el externo, que comunica la alarma a una autoridad competente para que intervenga y apoye a la comunidad. El objetivo es coordinar acciones conjuntas que ayuden a eliminar cualquier peligro o amenaza identificada [14].

1.3.7 Propósito de los sistemas de alarmas comunitarias

Los sistemas de alarmas comunitarias están diseñados para fortalecer la seguridad de una comunidad, un edificio o lugar, facilitando la comunicación y respuesta rápida ante eventos de riesgo como robos, emergencias médicas, incendios o desastres naturales. Su propósito principal es alertar y movilizar a las personas y a las autoridades para una respuesta coordinada y efectiva [15].

1.3.8 Protocolos de comunicación de los sistemas de alarmas comunitarias

Los protocolos de comunicación definen los estándares y reglas para el envío y recepción de datos, con el fin de asegurar que todos los elementos del sistema trabajen en conjunto. A continuación, se presentan algunos de los principales protocolos utilizados en estos sistemas:

a. Protocolos de Red Local (LAN)

- **Wi-Fi (IEEE 802.11):** Facilita la comunicación entre dispositivos en la misma red local y permite la transmisión rápida de datos. Es adecuado para áreas donde hay disponibilidad de routers Wi-Fi y se requiere un alto ancho de banda para transmitir información de audio, video o datos en tiempo real [16].
- **Zigbee:** Utilizado para la comunicación en redes de bajo consumo de energía, ideal para sensores y dispositivos IoT. Tiene un alcance limitado pero es confiable para la transmisión de datos en interiores [16].

- **Bluetooth:** Se emplea en algunos sistemas de alarmas de corto alcance y bajo consumo. Puede ser útil para la comunicación directa entre dispositivos personales (como teléfonos móviles) y dispositivos de alarma [16].

b. Protocolos de Red de Área Amplia (WAN)

- **GSM/3G/4G/5G:** Permite la comunicación remota de los dispositivos de alarma con servidores en la nube o aplicaciones móviles a través de redes celulares. Es adecuado para áreas rurales o sitios donde no hay cobertura Wi-Fi [16].
- **LoRaWAN:** Utilizado para comunicación de largo alcance y bajo consumo de energía. Ideal para sensores que no necesitan transmitir grandes cantidades de datos, como dispositivos de monitoreo ambiental o de estado [16].

c. Protocolos de Mensajería y Notificación

- **MQTT (Message Queuing Telemetry Transport):** Un protocolo de comunicación ligero y eficiente en redes inestables o de bajo ancho de banda. Es común en IoT para enviar y recibir datos de sensores y dispositivos de alarma a un servidor central o sistema de monitoreo [16].
- **HTTP / HTTPS:** Es utilizado en aplicaciones de notificación que envían alertas o reportes a través de servidores web. Con HTTPS, se garantiza una transmisión de datos segura y encriptada, que protege la información [16].
- **WebSockets:** Permite comunicación en tiempo real entre aplicaciones web y servidores. Es útil para notificaciones instantáneas en interfaces de usuario “dashboards de monitoreo” [16].

d. Protocolos de Integración de Dispositivos y Sensores

- **Modbus y BACnet:** Utilizados en la automatización de edificios para la comunicación entre dispositivos y sensores de seguridad. Son protocolos industriales que garantizan confiabilidad y compatibilidad entre distintos dispositivos en una red [16].

- **RESTful APIs:** Las APIs RESTful permiten la integración de sistemas de alarma comunitaria con aplicaciones móviles, plataformas web y otros sistemas de control. A través de estos, se pueden consultar estados, configurar alarmas y gestionar notificaciones de manera remota [16].

e. Protocolos de Seguridad en Comunicación

- **SSL/TLS:** Para la encriptación de datos transmitidos a través de redes públicas. Protege contra accesos no autorizados y asegura que solo los usuarios y dispositivos autorizados puedan acceder a la información [16].
- **VPN:** Utilizado para proteger las comunicaciones en redes de áreas amplias o cuando se accede remotamente a la infraestructura de monitoreo, garantizando la privacidad y seguridad de los datos [16].

En la Tabla 1 se muestra un resumen de los protocolos de comunicación de los sistemas de alarmas comunitarias, en conjunto con las ventajas y desventajas de los mismos.

Tabla 1. Protocolos de comunicación.

Tipo de Protocolo	Protocolo	Descripción	Ventajas	Desventajas	Aplicación Principal
Red Local (LAN)	Wi-Fi (IEEE 802.11)	Transmisión de datos en redes locales con routers Wi-Fi.	Alta velocidad, ancho de banda suficiente.	Consumo elevado de energía, limitado por el alcance.	Conexión de sensores y cámaras a corta distancia.
	Zigbee	Red de bajo consumo para dispositivos IoT con alcance limitado.	Bajo consumo de energía, ideal para interiores.	Alcance reducido, menor ancho de banda.	Sensores y dispositivos IoT en áreas cercanas.
	Bluetooth	Comunicación directa de corto alcance entre dispositivos.	Bajo consumo, conexión rápida.	Corto alcance, menor capacidad de conexión simultánea.	Comunicación entre dispositivos móviles y alarmas.
Red de Área Amplia (WAN)	GSM/3G/4G/5G	Comunicación remota a través de redes celulares.	Cobertura amplia, adecuada para áreas sin Wi-Fi.	Dependencia de la red celular, costos de datos.	Conexión remota en áreas rurales.
	LoRaWAN	Comunicación de largo alcance con bajo consumo.	Larga distancia, bajo consumo energético.	Baja velocidad, adecuado solo para	Sensores de monitoreo ambiental en exteriores.

Tipo de Protocolo	Protocolo	Descripción	Ventajas	Desventajas	Aplicación Principal
		ideal para pequeños datos.		pequeños paquetes.	
Mensajería y Notificación	MQTT	Protocolo ligero para IoT, ideal para transmisión eficiente en redes inestables.	Uso mínimo de ancho de banda, confiable.	No es ideal para grandes volúmenes de datos.	Transmisión de datos de sensores y alertas.
	HTTP/HTTPS	Protocolo web para notificaciones y reportes; HTTPS agrega seguridad.	Amplia compatibilidad, datos seguros con HTTPS.	Requiere más recursos que MQTT, susceptible a retrasos.	Envío de reportes y acceso a dashboards.
	WebSockets	Permite comunicación en tiempo real entre servidor y cliente web.	Comunicación instantánea, ideal para aplicaciones web.	Requiere conexión constante, uso de más ancho de banda.	Notificaciones en dashboards web en tiempo real.
Integración de Dispositivos	Modbus/BACnet	Protocolos industriales para comunicación entre sensores y dispositivos en redes de automatización.	Alta compatibilidad y confiabilidad en entornos industriales.	Más complejos, requieren configuración específica.	Integración de sensores en edificios.
	RESTful APIs	APIs para la integración con plataformas y aplicaciones web/móviles.	Flexibilidad y escalabilidad, fácil de implementar.	Requiere una configuración de seguridad adicional.	Control y monitoreo remoto de alarmas.
Seguridad en Comunicación	SSL/TLS	Protocolos de encriptación para proteger los datos en transmisión.	Alta seguridad en redes públicas.	Puede aumentar el tiempo de transmisión de datos.	Protección de datos en internet.
	VPN	Protege la comunicación en redes amplias, proporcionando acceso remoto seguro.	Alta privacidad y seguridad de datos.	Mayor consumo de recursos y configuración compleja.	Acceso seguro a redes de monitoreo remoto.

1.3.9 Características de los protocolos de comunicación

Los protocolos de comunicación en un sistema de alarmas comunitarias se basan en la necesidad de garantizar transmisión confiable, baja latencia, cobertura amplia y eficiencia energética, optimizando la respuesta ante eventos de emergencia [16]. A continuación, se presentan algunas de las principales características de los protocolos utilizados en estos sistemas:

a. Ethernet (IEEE 802.3): Empleado comúnmente por su alta estabilidad y velocidad, lo que lo hace ideal para conectar servidores de monitoreo y sistemas de control en sistemas de alarmas comunitarias [16].

- **Características técnicas:**

- Ofrece velocidades desde 10 Mbps hasta 100 Gbps.
- Utiliza cables UTP, STP o fibra óptica, dependiendo de la infraestructura.
- Baja latencia y alta confiabilidad en la transmisión de datos.

b. Wi-Fi (IEEE 802.11): Empleado para la interconexión inalámbrica de dispositivos dentro de la red local, como sensores, cámaras y paneles de monitoreo, sin necesidad de cableado [16].

- **Características técnicas:**

- Velocidades de hasta 9.6 Gbps en Wi-Fi 6 (802.11).
- Opera en las bandas de 2.4 GHz y 5 GHz, ofreciendo un equilibrio entre alcance y velocidad.
- Soporta mecanismos de seguridad avanzada (WPA3) para proteger la red contra accesos no autorizados.

c. TCP/IP (Transmission Control Protocol / Internet Protocol): Implementados para la transmisión y direccionamiento de datos dentro de la red LAN, asegurando una comunicación estructurada y organizada entre dispositivos [16].

- ***Características técnicas:***

- Modelo de capa 4, permite la segmentación y reenvío eficiente de datos.
- Utiliza direcciones IPv4 (32 bits) o IPv6 (128 bits) para identificar dispositivos en la red.
- Garantiza la entrega confiable de paquetes mediante mecanismos de control de flujo y corrección de errores.

d. MQTT (Message Queuing Telemetry Transport): Se emplea para la comunicación entre dispositivos IoT dentro de la red LAN, como sensores y actuadores, permitiendo la transmisión eficiente de datos con bajo consumo de ancho de banda [16].

- ***Características técnicas:***

- Basado en el protocolo TCP/IP, permitiendo conexiones persistentes a lo largo del tiempo.
- Usa un esquema publicador/suscriptor, optimizando la comunicación en entornos de alarmas.
- Requiere bajo ancho de banda, adecuado para dispositivos con recursos limitados.

1.3.10 Redes inalámbricas

Las redes inalámbricas se han convertido en una tecnología dominante en la actualidad, permitiendo a los usuarios conectarse en cualquier momento y lugar del mundo para acceder a información y diversos recursos. Estas redes se comunican mediante medios no guiados, utilizando ondas electromagnéticas para llevar a cabo la transmisión y recepción a través de antenas [17].

a. Clasificación de redes inalámbricas

Las redes inalámbricas se pueden clasificar en función de su aplicación y el alcance de la señal en redes de área personal, redes de área local, redes metropolitanas y redes de área amplia. A continuación se describen 3 tipos de redes inalámbricas:

- **WPAN.** Estas redes están basadas en el estándar 802.15 y ofrecen un rango de cobertura de hasta 10 metros. Se destacan por su bajo consumo de energía y su velocidad de transmisión reducida. Entre las tecnologías más relevantes en este ámbito se encuentran Bluetooth, IrDA, ZigBee y RFID [17].
- **WLAN.** Redes diseñadas para comunicaciones a distancias de hasta 100 metros y basadas en el estándar 802.11, se utilizan principalmente en hogares e instituciones educativas. Ofrecen escalabilidad y facilitan la expansión, lo que permite crear más puntos de conexión en áreas de difícil acceso [17].
- **WMAN.** Las redes conocidas como WiMAX están basadas en el estándar 802.16 y utilizan una arquitectura de punto a multipunto, lo que les permite ofrecer altas velocidades de transmisión de datos. WiMAX opera en bandas licenciadas y no licenciadas que van de 2 GHz a 11 GHz y de 10 GHz a 66 GHz. Esta tecnología puede alcanzar velocidades de hasta 70 Mbps a distancias de 50 km, permitiendo la conexión de miles de usuarios [17].

1.3.11 IoT (Internet de las Cosas)

El concepto de IoT se refiere a la capacidad de conectar dispositivos y objetos físicos a través de internet, permitiéndoles comunicarse entre sí y con sistemas externos. Esta interconexión no se limita solo a dispositivos como computadoras o teléfonos móviles, sino que incluye objetos cotidianos como electrodomésticos, vehículos, sensores de monitoreo, maquinaria industrial, entre otros [18]. En la siguiente figura se puede apreciar de manera general la interacción que el IoT permite con los diferentes equipos.



Figura 3. Internet de las cosas [18].

1.3.12 Arquitectura IoT

La arquitectura de un sistema IoT se basa en la orientación de eventos, las capas se estructuran progresivamente, integrando tecnologías y componentes específicos en cada nivel [18]. En la Figura 4 se puede apreciar de manera gráfica como se debería estructurar una arquitectura IoT.

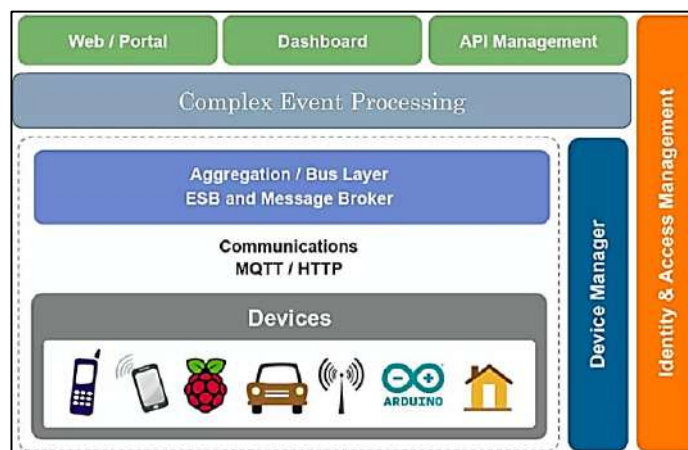


Figura 4. Arquitectura IoT [18].

La capa inicial incluye aplicaciones de usuario y portales web, desde donde se generan datos que luego se procesan y analizan en niveles superiores. A partir de ahí, los datos se envían mediante protocolos HTTP y MQTT hacia diversos dispositivos, entre ellos teléfonos, Raspberry Pi, Arduino, y sensores, etc [18].

1.3.13 Desafíos del IoT

Como toda tecnología, el IoT presenta diversos desafíos en múltiples ámbitos, incluyendo la necesidad de estandarización, la capacidad de escalabilidad, la heterogeneidad de los dispositivos, la interoperabilidad entre sistemas, la disponibilidad continua de servicios, la eficiencia energética y el impacto ambiental [18].

1.3.14 LoRa

LoRa (Long Range) es una tecnología de comunicación inalámbrica de amplia cobertura y baja potencia, utilizada principalmente por su eficiencia energética, alcance y robustez en entornos IoT. Su modulación basada en espectro ensanchado de chirp (CSS - Chirp Spread Spectrum) permite una alta sensibilidad del receptor, logrando transmisión de datos a largas distancias (hasta 15 km en zonas rurales y 2-5 km en entornos urbanos) con baja potencia. Siendo ideal para aplicaciones IoT donde los dispositivos funcionan con baterías durante años sin necesidad de mantenimiento constante. Por otra parte, su capacidad para operar en bandas ISM (Industrial, Scientific and Medical) sin licencia (868 MHz en Europa y 915 MHz en América) reduce costos operativos y facilita su implementación en redes LPWAN (Low Power Wide Area Network). Su robustez frente a interferencias y su capacidad de penetración en obstáculos urbanos garantizan una comunicación estable y confiable, lo que la convierte en una opción óptima para sistemas de monitoreo remoto, sensores ambientales, alarmas comunitarias y otros dispositivos distribuidos geográficamente [19].

a. Aspectos claves de LoRa

- **Bajo consumo de energía:** Ideal para dispositivos alimentados por baterías, que pueden operar durante meses sin recarga [19].
- **Largo alcance:** Puede cubrir grandes áreas, lo cual es útil en aplicaciones rurales y urbanas [19].
- **Bajo ancho de banda:** Transmite datos a velocidades bajas, adecuado para aplicaciones donde la velocidad de datos no es crucial [19].
- **LoRaWAN:** LoRa se usa a menudo en redes LoRaWAN, que permiten comunicaciones bidireccionales y ofrecen un protocolo de red más completo para gestionar y autenticar dispositivos [19].

1.4 Objetivos

1.4.1 Objetivo general

Diseñar un sistema de control y monitorización de alarmas comunitarias para el Taller de Innovación y Desarrollo Tecnológico FISEI.

1.4.2 Objetivos específicos

- Investigar los sistemas de control y monitorización de alarmas comunitarias.
- Analizar los equipos para el sistema de control y monitorización de alarmas comunitarias.
- Implementar el sistema de control y monitorización de alarmas comunitarias para el Taller de Innovación y Desarrollo Tecnológico FISEI de la Universidad Técnica de Ambato.

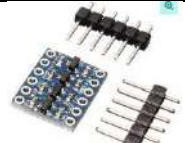
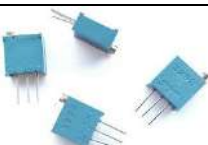
CAPÍTULO II. METODOLOGÍA

2.1 Materiales

En la Tabla 2 se detallan los recursos materiales que se utilizaron para la elaboración de proyecto de titulación.

Tabla 2. Recursos materiales empleados en el sistema.

Materiales	Descripción	Imagen
Modulo LILYGO TTGO LoRa32 915Mhz V1.6.1	La ESP32 es un microcontrolador versátil con Wi-Fi, Bluetooth y un módulo LoRa 868 / 915 MHz, ideal para IoT y automatización. Ofrece transmisión bidireccional de datos a largas distancias, siendo una solución flexible para proyectos que requieren conectividad inalámbrica eficiente.	
Botón de pánico con llave	Dispositivo que activa o cierra un circuito y solo puede desactivarse mediante una llave de seguridad.	
Sirena con diferentes tonos	Dispositivo de alerta sonora capaz de emitir diversos tonos o frecuencias, facilitando la diferenciación entre tipos de alarmas o niveles de emergencia.	
Batería recargable	Dispositivos que almacenan energía, pueden ser recargadas y reutilizadas múltiples veces.	
Cables	Conductores eléctricos diseñados para transportar electricidad o señales entre distintos puntos.	
Resistencias	Componentes eléctricos que regulan el flujo de corriente en un circuito.	

Materiales	Descripción	Imagen
Transistores	Dispositivos semiconductores utilizados para amplificar o conmutar señales eléctricas y energía.	
Convertidor de nivel lógico	Circuito que permite adaptar señales eléctricas entre diferentes niveles de voltaje, asegurando que la comunicación entre dispositivos con diferentes especificaciones de voltaje sea posible y segura.	
Switch	Dispositivo eléctrico que permite abrir o cerrar un circuito, controlando así el flujo de corriente eléctrica.	
Potenciómetro de precisión	Componente eléctrico que se utiliza para ajustar y controlar niveles de resistencia en un circuito con un alto grado de exactitud y estabilidad	
Baquelita	La baquelita es un tipo de resina plástica termoestable que se caracteriza por su alta resistencia térmica y eléctrica.	
Estaño	Material que permite la unión de diferentes elementos o componentes electrónicos, usado principalmente en la industria electrónica.	
Borneras	También llamados bloques de terminales o terminales de conexión, son dispositivos eléctricos utilizados para conectar varios conductores en un circuito eléctrico de manera rápida y segura.	
Conector DC hembra	Componente que se utiliza para recibir corriente continua (DC) a través de una conexión con un conector macho, comúnmente de adaptadores de corriente.	
Tornillos	Elementos de fijación mecánica que se utilizan para unir o asegurar dos o más objetos.	
Panel de control de alarma contra incendios Bosch	Dispositivo centralizado que supervisa y controla el sistema de detección y alarma de incendios en un edificio o instalación.	

En la Tabla 3 se describen los softwares que se emplearon para la programación y recolección de la información.

Tabla 3. Softwares empleados para el sistema.

Software	Descripción	Imagen
Visual Studio Code	Visual Studio Code (VS Code) es un editor de código fuente ligero, gratuito y multiplataforma, desarrollado por Microsoft. Está diseñado para escribir y depurar código, con soporte para múltiples lenguajes de programación y una amplia variedad de extensiones para ampliar sus funcionalidades.	
Ubuntu	Es un sistema operativo de código abierto basado en Linux, que sirve para diversas tareas y entornos, tanto en computadoras personales como en servidores.	
Nginx	Es un servidor web y proxy inverso de código abierto, diseñado para manejar una gran cantidad de conexiones simultáneas con eficiencia y velocidad.	
MySQL WorkBench	Es una herramienta visual para el diseño y administración de bases de datos MySQL, que ofrece un entorno gráfico para interactuar con bases de datos sin necesidad de utilizar comandos.	
Mosquitto	Mosquitto es un broker MQTT de código abierto que implementa el protocolo MQTT, diseñado para transmitir mensajes ligeros en redes de bajo ancho de banda o dispositivos de recursos limitados, comúnmente utilizados en el Internet de las Cosas (IoT).	
Node JS	Es un entorno de ejecución de código abierto que permite ejecutar JavaScript en el lado del servidor.	
Python	Es un lenguaje de programación de alto nivel, interpretado y de propósito general, conocido por su simplicidad y legibilidad al momento de realizar la codificación.	
Proteus	Es un software de simulación y diseño de circuitos electrónicos.	

Software	Descripción	Imagen
Cloudinary	Es un servicio basado en la nube que permite gestionar, almacenar, optimizar y entregar medios (imágenes, videos y otros archivos) de manera eficiente en aplicaciones web y móviles.	
Fusion 360	Es un software de diseño asistido por computador (CAD) que permite la creación y la manipulación de modelos tridimensionales.	

2.2 Métodos

2.2.1 Modalidad de la investigación

a. Investigación bibliográfica documental

Se realizó una investigación bibliográfica para recopilar información de diversas fuentes, como artículos científicos, libros y revistas especializadas, sobre el desarrollo y gestión de sistemas de alarmas comunitarias utilizando tecnologías modernas como el Internet de las Cosas (IoT) y redes inalámbricas. Asimismo, se consultó información en línea para identificar equipos que cumplan con los requisitos necesarios para la monitorización y control eficiente del sistema, permitiendo su gestión remota en tiempo real.

- **Metodología prisma.** La metodología PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) es una herramienta ampliamente utilizada para llevar a cabo revisiones sistemáticas de la literatura en diversas bases de datos científicas. De manera resumida, consta de cinco pasos principales a seguir:

1. Formulación de las preguntas de investigación
2. Definir los términos adecuados de búsqueda
3. Selección de la documentación adecuada
4. Extracción de la documentación

5. Determinación de resultados

• Preguntas de investigación

Las preguntas de investigación son fundamentales para orientar el enfoque de la revisión de la literatura, en este caso se basaron en explorar el internet de las cosas IoT, redes de comunicación y su aplicación en sistemas de alarmas comunitarias. Para ello, se consideran tres preguntas detalladas en la Tabla 4.

Tabla 4. Preguntas de investigación.

Número	Pregunta de investigación (RQ)	Motivación
RQ1	¿De qué manera el IoT y las redes de comunicación influye en los sistema de alarmas comunitarias?	Identificar el impacto que el IoT y las redes de comunicación ha tenido en los sistemas de alarmas comunitarias actuales.
RQ2	¿Qué aspectos fundamentales han sido considerados para el uso del IoT y las redes de comunicación en los sistemas de alarmas comunitarias?	Conocer los aspectos fundamentales que las empresas de seguridad consideran para el uso del IoT en los sistemas de alarmas comunitarias
RQ3	¿Cuáles son los beneficios de aplicar el IoT a un sistema de alarmas comunitarias?	Aplicar el IoT en sistemas de control y monitorización de alarmas comunitarias.

• Términos de búsqueda

La revisión documental se llevó a cabo utilizando bases de datos como SciELO, SCOPUS, IEEE Xplore, Web of Science, DOAJ y Google Scholar, asegurando que los artículos científicos seleccionados tuvieran una fecha de publicación no mayor a cinco años. Esta decisión se basó en la recomendación expuesta en el artículo “Norma editorial: ¿es adecuado no citar los artículos de más de cinco años de antigüedad?” [20], con el fin de garantizar que la información obtenida sea lo más actual y relevante posible. Los términos empleados para la búsqueda se detallan en la Tabla 5.

Tabla 5. Términos de búsqueda.

Pregunta de investigación	Términos de búsqueda	Punto de vista
RQ1	“Systems” & “Alarms” & “Communication Networks” & “IoT Alarms”	VP1
	“Systems” & “Alarms” & “Communication Networks” & “IoT Alarms” & “Influences” & “Applications”	VP2

Pregunta de investigación	Términos de búsqueda	Punto de vista
RQ2	“Factors” & “Systems” & “Alarms” & “Communication Networks”	VP3
	“Uses” & “Systems” & “Alarms” & “Communication Networks”	VP4
RQ3	“Applications” & “Systems” & “Alarms” & “Communication Networks” & “IoT Alarms”	VP5
	“Benefits” & “Systems” & “Alarms” & “Communication Networks” & “IoT Alarms”	VP6

- **Selección de la documentación**

En esta fase de la revisión se aplicaron diversos criterios de inclusión y exclusión que permitieron seleccionar los documentos más relevantes para el tema de investigación. Los criterios utilizados se presentan en la Tabla 6.

Tabla 6. Criterios de búsqueda.

Número	Inclusión	Exclusión
C1	Artículos relacionados con la temática	Artículos repetidos
C2	Artículos en inglés y español	Páginas y sitios web que no presente una fuente confiable de la información.
C3	Artículos relacionados con los sistemas de alarmas y la seguridad ciudadana.	Investigaciones realizadas con enfoques distintos a la seguridad ciudadana.
C4	Artículos relacionados con las redes de comunicación que existen entre los dispositivos electrónicos.	Artículos no relacionados al tema
C5	Artículos relacionados con el IoT y su apoyo a los sistemas de seguridad ciudadana.	Artículos no relacionados al tema
C6	Artículos publicados en los últimos cinco años.	Artículos con más de 5 años de antigüedad
C7	Artículos disponibles, libre acceso	Artículos con restricción de acceso

Tras realizar una revisión sistemática de la documentación seleccionada en base a los criterios de inclusión y exclusión, se identificaron 15 artículos que aportaron información relevante y de gran valor para el tema de investigación. La selección de los documentos que conforman este estudio se detalla en la Figura 5, destacando los criterios utilizados para seleccionar o descartar material relevante.

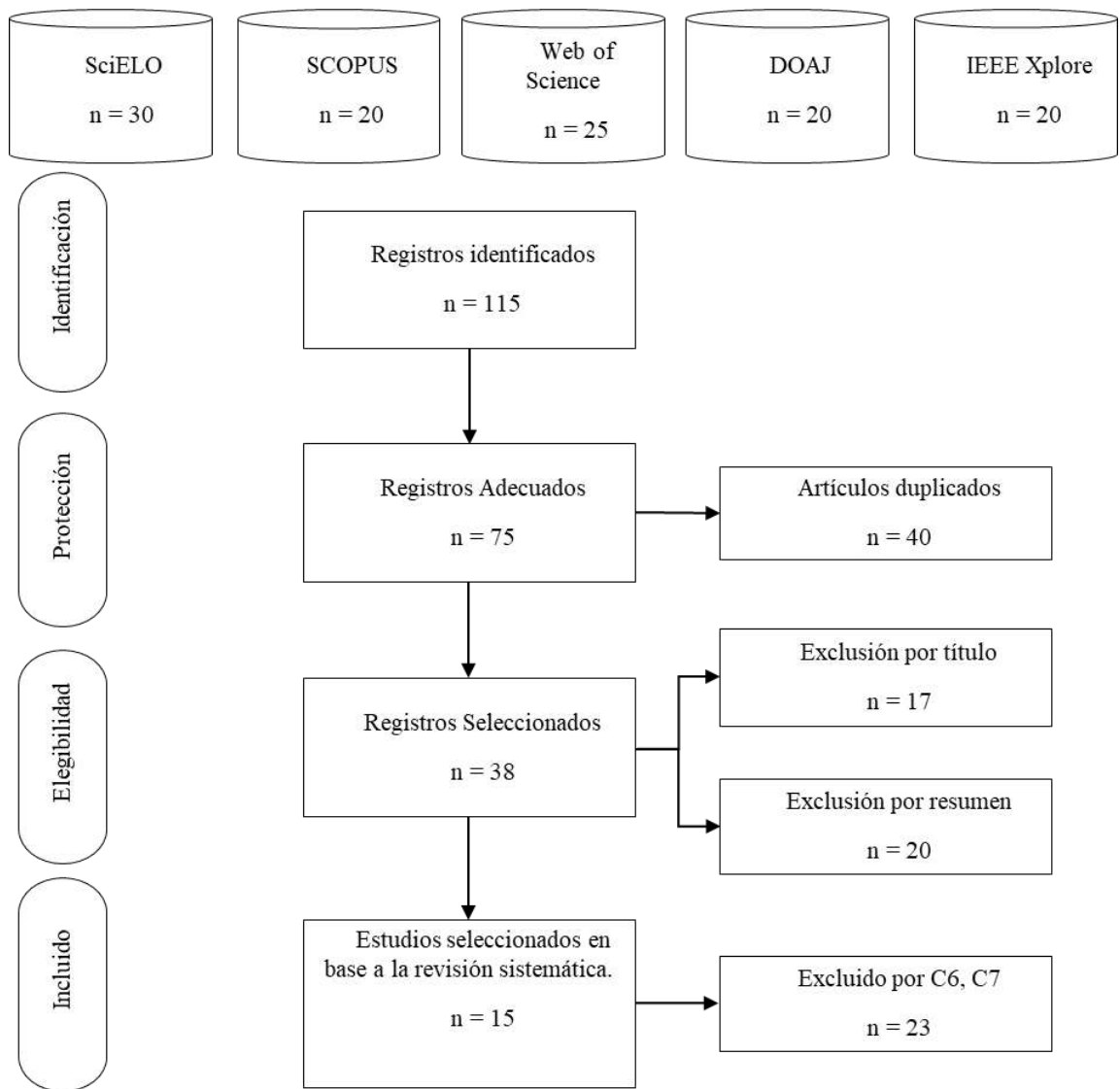


Figura 5. Diagrama de flujo de la Metodología PRISMA.

b. Investigación aplicada

La investigación fue aplicada ya que permitió desarrollar nuevas técnicas de procesamiento de datos y análisis en tiempo real, enfocadas en mejorar la precisión y rapidez de la detección y clasificación de eventos dentro de un sistema de alarmas comunitarias. Asimismo, se centró en la experimentación con algoritmos capaces de detectar situaciones de riesgo de manera eficiente.

c. Investigación de campo

Se utilizó esta investigación debido a que se llevó a cabo en el Taller de Innovación y Desarrollo Tecnológico FISEI, un espacio donde pueden surgir diversas situaciones relacionadas con la seguridad personal. Lo cual permitió recopilar información valiosa, fundamental para el desarrollo del proyecto.

d. Investigación experimental

Se llevó a cabo una investigación experimental enfocada en las pruebas de funcionamiento del sistema de alarmas comunitarias para comprobar la eficacia de los dispositivos de detección y comunicación, así como la integración de la aplicación móvil diseñada para la activación del mismo. El propósito fue garantizar una comunicación efectiva entre los diversos componentes del sistema.

2.2.2 Recolección de información

La Tabla 7, presenta la recopilación de información de acuerdo con los objetivos establecidos para el desarrollo del trabajo de investigación.

Tabla 7. Recolección de la información.

Objetivos	Técnica/ Método	Herramienta/ instrumento
Investigar los sistemas de control y monitorización de alarmas comunitarias.	Recopilación de información	Bases de datos académicas, libros especializados, artículos científicos.
	Recopilación de información	Bases de datos académicas, libros especializados, artículos científicos.
Analizar los equipos para el sistema de control y monitorización de alarmas comunitarias.	Recopilación de información	Datasheet de los componentes que conformarán el sistema de control y monitorización.
	Recopilación de información.	Datasheet de los componentes que conformarán el sistema de control y monitorización.
	Método de ponderación.	Tablas de ponderación de selección de componentes del sistema de control y monitorización.
Implementar el sistema de control y monitorización de alarmas comunitarias para el Taller de Innovación y Desarrollo Tecnológico	Construcción del Sistema de alarma.	Diagramas de conexión eléctrico. Datasheet de los equipos.
	Construcción del Sistema de alarma.	1.Simulación del sistema de control y monitorización, 2. Fichas de resultados sobre las salidas sistema.

Objetivos	Técnica/ Método	Herramienta/ instrumento
FISEI de la Universidad Técnica de Ambato.	Pruebas de funcionamiento.	1. Tablas de registro de datos. 2. Tablas de comparativa de salidas manual y automático.

2.2.3 Procesamiento y análisis de datos

El procesamiento de datos se determinó bajo los siguientes parámetros:

- Se llevó a cabo un análisis exhaustivo de artículos científicos y trabajos de titulación que proporcionen información relevante sobre el diseño de sistemas de alarmas comunitarias con IoT.
- Se elaboró planos detallados del cableado para adaptar el sistema a las necesidades específicas, asegurándose de la correcta conexión y configuración.
- Se comparó las distintas características de los componentes disponibles en el mercado, considerando cualidades como el rendimiento, calidad y compatibilidad, para tomar decisiones informadas en cuanto a la adquisición de nuevos equipos.
- Se seleccionó cuidadosamente los componentes y controladores que mejor se ajusten a las necesidades, tomando en consideración varios factores como el precio, beneficios, protocolos de comunicación compatibles y aplicaciones necesarias.
- Se llevó a cabo la adquisición de los equipos seleccionados, siguiendo un proceso de compra que garantice la calidad y compatibilidad con las necesidades.
- Se realizó la instalación y configuración precisa de los componentes y controladores, utilizando protocolos de comunicación y softwares correspondientes, para asegurar un funcionamiento óptimo del sistema.

- Se identificó los diferentes softwares apropiados que permitan controlar y monitorizar de manera eficiente el sistema automatizado, facilitando así la gestión del mismo.
- Se realizó pruebas para verificar el funcionamiento adecuado de todo el sistema, asegurándose de que responda correctamente y cumpla con los requisitos necesarios.
- Se elaboró el informe final del proyecto que respalde de manera escrita todas las etapas y resultados obtenidos a lo largo del desarrollo del proyecto, así como las conclusiones y recomendaciones para futuras mejoras o actualizaciones del sistema de alarmas comunitarias.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de resultados

En este capítulo se presenta un análisis detallado de los resultados obtenidos en la implementación del "Sistema de Control y Monitoreo de Alarmas Comunitarias para el Taller de Innovación y Desarrollo Tecnológico FISEI de la Universidad Técnica de Ambato". Para contextualizar adecuadamente el proyecto, es importante recalcar que la investigación abarcó una revisión detallada de la literatura bajo la metodología PRISMA, seguida por el diseño y desarrollo de un sistema completo de alarmas, la selección de equipos adecuados, la configuración del servidor junto con la base de datos en línea y la programación de los dispositivos “tarjetas ESP32” para la recolección de los datos. Todo esto se integró con una aplicación basada en IoT, permitiendo su funcionamiento optimizado.

3.1.1 Análisis de los sistemas de alarmas comunitarias

Para analizar los sistemas de alarmas comunitarias, que abarcan el control y monitoreo a través de dispositivos conectados e integrados mediante IoT, se aplicó la metodología PRISMA, siguiendo los pasos de una revisión sistemática de la bibliografía relacionada. Como resultado, se recopilaron 15 artículos científicos pertinentes, los cuales proporcionan respuestas detalladas a las preguntas planteadas, las mismas se muestran en la Tabla 8.

Tabla 8. Resultados de investigación.

Ref.	¿De qué manera el IoT y las redes de comunicación influye en los sistema de alarmas comunitarias?	¿Qué aspectos fundamentales han sido considerados para el uso del IoT y las redes de comunicación en los sistemas de alarmas comunitarias?	¿Cuáles son los beneficios de aplicar el IoT a un sistema de alarmas comunitarias?
[21]	El IoT, combinado con redes de comunicación como LoRa, permite una vigilancia eficiente y de largo alcance en sistemas de alarmas, facilitando la detección y notificación automática de emergencias en tiempo real a través de dispositivos interconectados.	Los aspectos que han sido considerados en este trabajo son: 1. La facilidad de uso (especialmente personas mayores). 2. La sostenibilidad económica. 3. La eficiencia en el manejo de energía y costo de implementación.	Los beneficios de aplicar IoT a un sistema de alarmas comunitarias, incluyen una mayor seguridad mediante notificaciones automática de emergencia, comunicación en tiempo real con la comunidad, y un entorno más seguro para personas de avanzadas edades.

Ref.	¿De qué manera el IoT y las redes de comunicación influye en los sistemas de alarmas comunitarias?	¿Qué aspectos fundamentales han sido considerados para el uso del IoT y las redes de comunicación en los sistemas de alarmas comunitarias?	¿Cuáles son los beneficios de aplicar el IoT a un sistema de alarmas comunitarias?
[22]	Estas nuevas tecnologías conectada a los sistemas tradicionales de seguridad permiten la detección de fallos o anomalías en tiempo real y ofrecen funciones de autodiagnóstico, lo que reduce la necesidad de personal especializado en sitio.	Los aspectos que han sido considerados en este trabajo son: 1. Ciberseguridad, dado la necesidad de mantener altos niveles de protección en los datos recolectados. 2. La confiabilidad de las conexiones de comunicación. 3. La disponibilidad de datos en tiempo real para permitir la rápida detección y gestión de emergencias. 4. Estrategias de mantenimiento preventivo para mejorar la calidad y disponibilidad del sistema.	Entre los múltiples beneficios que aporta el IoT, dentro de un sistema de alarmas permite la optimización en el tiempo de respuesta al detectar incidentes y notificar a los servicios de emergencia de manera inmediata.
[23]	A nivel de industrial las redes de comunicación permite el control continuo de las operaciones dentro y fuera del lugar de trabajo, enfocadas a la detección temprana de incidentes o peligros inminentes que podrían afectar de una u otra manera a la comunidad de trabajo.	Los aspectos que han sido considerados en este trabajo son: 1. Mantenimiento, todos los dispositivos deben ser revisados y calibrados a lo largo de su vida útil. 2. Identificación, cada componente del sistema debe estar etiquetado o numerado. 3. Operación, aplicaciones móviles que permitan el control remoto de la activación del sistema.	Sistemas que pueden operar en conjunto con diversos módulos “enfocados a una situación de riesgo en particular, incendios, inundaciones, robos, etc” que los fabricantes ofrecen en un mercado cada vez más cambiante.
[24]	Tecnologías como el IoT Ayudan a consolidar información en tiempo real, permitiendo que los datos sean analizados y que se envíen notificaciones instantáneas a los ciudadanos y a las fuerzas de seguridad. Facilitan la toma de medidas colectivas de protección o prevención.	Los aspectos que han sido considerados en este trabajo son: 1. Las tecnologías existentes y su capacidad de funcionamiento en diferentes dispositivos. 2. Viabilidad económica, dispositivos de bajo costo que cumplan con su función. 3. Escalabilidad del sistema.	Aplicar el IoT a un sistema de alarmas comunitarias permite diseñar y probar un sistema de alerta con características básicas que cumpla con los requisitos técnicos, manteniendo un bajo costo en implementación, operación y mantenimiento.
[25]	En ciudades inteligentes permite la creación de redes de sensores inalámbricos que recopilan datos en tiempo real sobre el ambiente y la estructura de los edificios, facilitando una respuesta inmediata ante eventos críticos como incendios o terremotos, gestionando la evacuación de las personas y alertando a los residentes en tiempo real sobre el estado de seguridad.	Los aspectos que han sido considerados en este trabajo son: 1. Redes de sensores de bajo consumo energético para maximizar la vida útil de los dispositivos. 2. Protocolos eficientes de transmisión de datos, que aseguren comunicación confiable y en tiempo real. 4. Estrategias de redundancia y respaldo, el uso de NFC para intercambiar información cuando la conectividad se interrumpe.	Automatización de procesos de alerta y evacuación, disminución de la dependencia en intervención humana. Bases de datos con históricos que permiten identificar patrones en cuestiones de emergencias. Distribución optimizada de los recursos durante situaciones críticas.

Ref.	¿De qué manera el IoT y las redes de comunicación influye en los sistemas de alarmas comunitarias?	¿Qué aspectos fundamentales han sido considerados para el uso del IoT y las redes de comunicación en los sistemas de alarmas comunitarias?	¿Cuáles son los beneficios de aplicar el IoT a un sistema de alarmas comunitarias?
		5. Algoritmos de análisis de datos.	
[26]	Transforman los sistemas de alarmas comunitarias al permitir la automatización y el control remoto mediante dispositivos interconectados, posibilitan monitorear áreas y detectar intrusiones en tiempo real.	Los aspectos que han sido considerados en este trabajo son: 1. Protocolos de comunicación seguros (TCP/IP y MQTT). 2. El consumo energético de los componentes y del sistema en general. 3. La sensibilidad de los equipos ante condiciones extremas de ambiente “llovizna, viento, calor”.	Reducción de costos en equipos y optimización de la eficiencia energética gracias a la interconectividad que ofrece el IoT con diversos dispositivos. Mejora la capacidad de respuesta ante situaciones de riesgo, permitiendo una gestión más rápida y eficiente de las emergencias.
[27]	Permiten la implementación de sistemas altamente integrados y adaptativos, capaces de detectar y responder a incidentes críticos en tiempo real, optimizan la transmisión de alertas y la activación de mecanismos automatizados de mitigación.	Los aspectos que han sido considerados en este trabajo son: 1. Redundancia y precisión: uso de múltiples sensores para minimizar falsas alarmas. 2. Algoritmos de lógica difusa para evaluar la gravedad de las situaciones y activar respuestas adecuadas. 3. Eficiencia energética y bajos costos.	En el ámbito de la lucha contra incendios, el IoT facilita la recopilación de datos en tiempo real sobre las condiciones del entorno y la evolución del incidente, permitiendo un análisis inmediato que optimiza la asignación de recursos y la ejecución de respuestas específicas para cada situación de riesgo inminente.
[28]	Al integrar sensores y dispositivos de comunicación de baja potencia con nuevas tecnologías como el IoT, el sistema de alarmas se vuelve más eficiente y capaz de monitorear un área amplia con menos consumo de energía.	Los sistemas de alarmas comunitarias habilitados con IoT suelen considerar: 1. Protocolos de comunicación de bajo consumo. 2. La confiabilidad en la entrega de mensajes. 3. La seguridad en la transmisión de datos. 4. La capacidad de integración con dispositivos de diferentes proveedores.	Permite un monitoreo constante y detallado de las condiciones del entorno, con base en la reducción de los costos operativos, el bajo consumo energético y la capacidad de respuesta en tiempo real.
[29]	Facilitan una respuesta rápida y precisa en situaciones de emergencia, ya que los sensores detectan eventos y transmiten datos automáticamente a través de redes como LoRaWAN, MQTT y Wi-Fi, mejorando la efectividad al reducir los tiempos de respuesta y permitir una vigilancia continua del entorno.	En los sistemas de alarmas comunitarias habilitados con IoT, se consideran aspectos como el bajo consumo de energía, la confiabilidad de los datos y la interoperabilidad entre diferentes dispositivos.	La arquitectura IoT ofrece una gran flexibilidad permitiendo la incorporación de nuevos dispositivos y sensores de manera modular, lo que facilita la expansión del sistema de alarmas comunitarias conforme a las necesidades específicas de cada entorno. Esta adaptabilidad permite que el sistema crezca y se ajuste a cambios como el aumento de la población, la aparición de nuevas áreas de riesgo o la implementación de tecnologías adicionales.
[30]	La tecnología IoT y las redes de comunicación desempeñan un rol crucial en la mejora de los	Los aspectos que han sido considerados en este trabajo son:	Los dispositivos con IoT suelen utilizar tecnologías de bajo consumo energético, como

Ref.	¿De qué manera el IoT y las redes de comunicación influye en los sistemas de alarmas comunitarias?	¿Qué aspectos fundamentales han sido considerados para el uso del IoT y las redes de comunicación en los sistemas de alarmas comunitarias?	¿Cuáles son los beneficios de aplicar el IoT a un sistema de alarmas comunitarias?
	sistemas de alarmas comunitarias al facilitar una comunicación constante entre sensores y plataformas de monitoreo.	1. Arquitecturas escalables que integran procesamiento de datos en la nube. 2. La interoperabilidad de los dispositivos. 3. La eficiencia energética y la seguridad en la transmisión de datos.	sensores y redes optimizadas, que requieren una cantidad mínima de energía para su funcionamiento continuo. Beneficioso en sistemas de alarmas comunitarias, ya que permite que los dispositivos se mantengan activos durante períodos prolongados sin necesidad de reemplazo frecuente de baterías o de una infraestructura de energía compleja.
[31]	Gracias a arquitecturas IoT avanzadas y sistemas inteligentes, es posible conectar dispositivos como asistentes personales y sensores en una red integral que detecta amenazas en tiempo real. Esto permite que las alertas se envíen automáticamente a los residentes o autoridades pertinentes.	En este trabajo se han considerado aspectos fundamentales como una infraestructura robusta y principios de diseño orientados a la innovación. Estos principios permiten implementar ecosistemas IoT flexibles y escalables que no solo se adaptan a las necesidades cambiantes de las comunidades, sino que también favorecen condiciones límite menos restrictivas.	Constituyen una amplia gama de soluciones tecnológicas destinadas a mejorar los sistemas entorno a los dispositivos que los conforman, potenciando su características de funcionamiento y rendimiento.
[32]	El IoT y las redes de comunicación transforman los sistemas de alarmas comunitarias al permitir la integración de sensores, microcontroladores y redes de comunicación que envían alertas en tiempo real. Permite que las alarmas no solo sean activadas localmente, sino que se comuniquen a través de plataformas en la nube para alertar a personas en ubicaciones remotas.	Los aspectos que han sido considerados en este trabajo son: 1. La confiabilidad del sistema. 2. La asequibilidad. 3. La capacidad de notificación en tiempo real. Adicionalmente, el sistema integra autenticación de usuario mediante contraseñas para evitar falsas alarmas.	Al utilizar servicios en la nube y comunicación Wi-Fi en lugar de sistemas de seguridad físicos o servicios de vigilancia, el sistema es más económico y accesible. Facilita la visualización de la ubicación exacta del incidente en tiempo real.
[33]	A través de sensores y tecnologías de comunicación inalámbrica, como Wi-Fi o GSM, se recopilan y envían datos ambientales o de eventos críticos a una base de datos central o directamente a dispositivos móviles de los residentes y autoridades. Garantizando una respuesta inmediata ante situaciones de emergencia.	Los principales aspectos considerados en este trabajo son la precisión y velocidad de transmisión de datos, la confiabilidad de los dispositivos de detección y la capacidad de integrar estos datos en plataformas de comunicación accesibles.	Posibilita un monitoreo constante y en tiempo real, enviando alertas automáticas y precisas a múltiples dispositivos, lo que facilita una respuesta inmediata y coordinada ante emergencias, reduce la necesidad de vigilancia manual y optimiza la seguridad de manera económica y eficiente.
[34]	En un sistema de alarma basado en IoT, los sensores detectan emergencias como humo o fuego y envían los datos a una unidad	Los aspectos que han sido considerados en este trabajo son: 1. Confiabilidad y precisión en la detección de humo o incendios.	La automatización de un sistema de alarmas basado en IoT permite una vigilancia constante y confiable en áreas extensas,

Ref.	¿De qué manera el IoT y las redes de comunicación influye en los sistema de alarmas comunitarias?	¿Qué aspectos fundamentales han sido considerados para el uso del IoT y las redes de comunicación en los sistemas de alarmas comunitarias?	¿Cuáles son los beneficios de aplicar el IoT a un sistema de alarmas comunitarias?
	central, que notifica de inmediato a usuarios y autoridades mediante aplicaciones móviles y SMS a través de redes Wi-Fi, 4G o 5G.	2. Conectividad y latencia. 3. Escalabilidad del sistema para mejorar en un futuro. 4. Eficiencia energética.	enviando alertas remotas y en tiempo real a residentes y servicios de emergencia, optimizan la capacidad de respuesta, reducen costos operativos y eliminan la necesidad de intervención manual.
[35]	El IoT y las redes de comunicación revolucionan los sistemas de alarmas comunitarias al permitir la conectividad y monitoreo remoto en tiempo real, facilitando la automatización y el control inteligente de dispositivos de seguridad.	Los aspectos que han sido considerados en este trabajo son: 1. Flujo de datos. 2. Seguridad en la base de datos mediante cadenas de bloques. 3. Eficiencia energética.	La integración con redes de última generación, como 5G, maximiza el potencial del IoT al permitir conexiones rápidas y estables entre dispositivos, lo que facilita el envío de alertas instantáneas en diferentes dispositivos móviles.

A partir de la información recopilada en la revisión sistemática de cada fuente, se procede a dar respuesta a las tres preguntas de investigación formuladas, con el fin de analizar las innovaciones tecnológicas en el ámbito de los sistemas de control y monitoreo de alarmas comunitarias. Este análisis se centró en la integración del IoT y las redes de comunicación, considerando su capacidad para optimizar la gestión de la seguridad y mejorar la respuesta ante incidentes en tiempo real.

Pregunta 1: ¿De qué manera el IoT y las redes de comunicación influye en los sistema de alarmas comunitarias?

El IoT y las redes de comunicación transforman significativamente los sistemas de alarmas comunitarias al mejorar su conectividad, rapidez de respuesta y capacidad de monitoreo en tiempo real. Con el IoT, los sensores y dispositivos de alarma pueden enviar datos constantemente a una plataforma central, lo que permite detectar incidentes de forma inmediata y sin intervención humana, facilitando una vigilancia continua, ya que cualquier anomalía es reportada al instante a través de notificaciones a usuarios, servicios de seguridad o plataformas de monitoreo.

Las redes de comunicación juegan un papel crucial, debido a que proporcionan la infraestructura necesaria para que los datos se transmitan de manera eficiente. Esto no solo acelera el tiempo de respuesta ante un evento, sino que también permite a los

sistemas recopilar y analizar datos históricos. Gracias a esta información, se pueden identificar patrones de seguridad, optimizar recursos y tomar decisiones basadas en datos para mejorar la seguridad. En conjunto, el IoT y las redes de comunicación convierten a los sistemas de alarmas comunitarias en soluciones más integradas, económicas y efectivas.

Pregunta 2: ¿Qué aspectos fundamentales han sido considerados para el uso del IoT y las redes de comunicación en los sistemas de alarmas comunitarias?

En el uso del IoT y las redes de comunicación para sistemas de alarmas comunitarias, se consideran diversos factores clave para maximizar su eficacia y utilidad. La exactitud y precisión son fundamentales para asegurar que el sistema detecte incidentes y situaciones de riesgo con alta fiabilidad, permitiendo alertas inmediatas y específicas. La confiabilidad se convierte en otro factor esencial, ya que garantiza que las alarmas y notificaciones sean consistentes y estén disponibles en todo momento.

La adaptabilidad de la infraestructura IoT es igualmente crucial, dado que los entornos comunitarios pueden experimentar cambios, como la adición de nuevos dispositivos o la ampliación de áreas de cobertura. La eficiencia económica es otro aspecto importante, debido a que estos sistemas buscan optimizar los recursos disponibles, minimizando costos operativos y energéticos.

En términos de seguridad, tanto la protección de datos personales como la prevención de accesos no autorizados son prioritarias para mantener la privacidad y el control del sistema. La posibilidad de operación autónoma de los dispositivos IoT permite una vigilancia continua sin intervención humana constante, aumentando la capacidad de respuesta ante situaciones de emergencia. Finalmente, el control de errores es esencial para garantizar la precisión de las alertas y reducir falsas alarmas, asegurando así un sistema de monitoreo confiable y eficaz.

Pregunta 3: ¿Cuáles son los beneficios de aplicar el IoT a un sistema de alarmas comunitarias?

La aplicación del IoT a un sistema de alarmas comunitarias proporciona beneficios significativos, como la mejora en la detección y respuesta ante incidentes en tiempo

real, la optimización de recursos, el monitoreo constante de la seguridad, la reducción de falsas alarmas y la capacidad de análisis de datos para identificar patrones de riesgo, lo cual contribuye a un entorno más seguro y eficiente.

Sin duda, las nuevas tecnologías emergentes están redefiniendo la manera en que se aborda la seguridad, al mejorar la eficiencia y efectividad de los sistemas de alarmas, y al mismo tiempo impulsar la participación activa de las personas que confirman su comunidad.

3.2 Desarrollo e implementación del sistema de alarmas

Para el desarrollo e implementación del sistema de alarmas comunitarias, se integraron cuatro fases:

- Diseño y construcción de placas electrónicas para el sistema.
- Conexión eléctrico de los componentes físicos del sistema
- Programación y aplicación web
- Pruebas de desempeño y validación del sistema de alarmas.

Este sistema emplea tecnología de comunicación inalámbrica basada en conexión Wi-Fi, lo que garantiza una transmisión eficiente y confiable de alertas de seguridad en tiempo real dentro de su área de cobertura. Además, gracias a las características de las tarjetas de desarrollo seleccionadas, el sistema también está preparado para incorporar comunicación inalámbrica basada en LoRa. Esta tecnología, reconocida por su bajo consumo de energía y su capacidad para transmitir datos a largas distancias, representa una opción ideal para ampliar la cobertura del sistema y asegurar su operatividad en entornos donde la infraestructura Wi-Fi pueda ser limitada o inexistente.

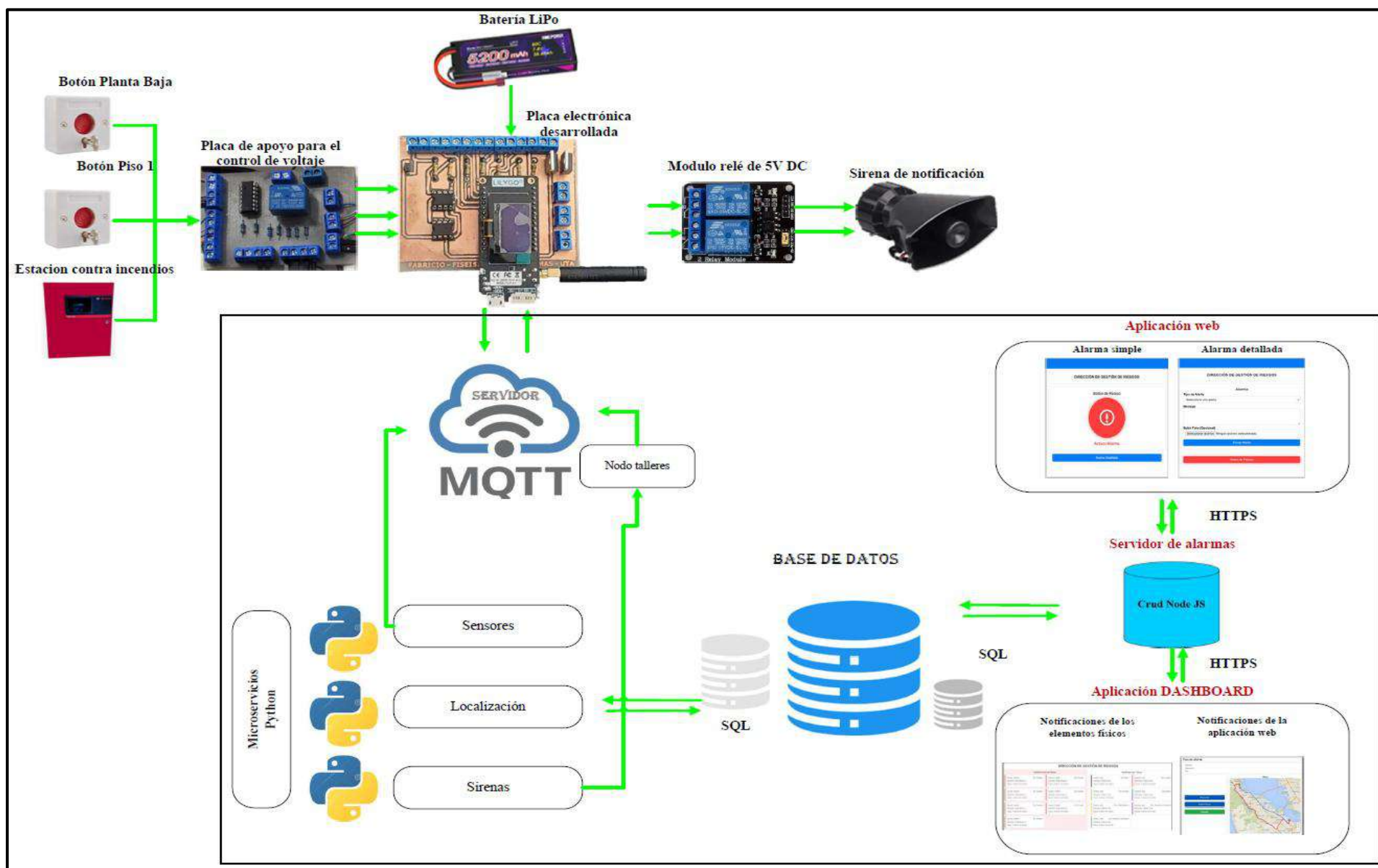


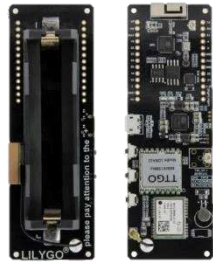
Figura 6. Esquema general del sistema.

3.3 Primera fase

3.3.1 Diseño y construcción de placas electrónicas para el sistema.

Para iniciar el desarrollo del piloto se realizó la identificación de las principales características técnicas entre dos tarjetas de desarrollo basadas en ESP 32 que incorporan módulos LoRa 868 y 915 MHz como se muestra en la Tabla 9.

Tabla 9. Características principales de las placas ESP 32.

Características	LILYGO LoRa32 V1.6.1	ESP32 LORA 923 MHz con GPS NEO-M8N y soporte para batería 18650 V1.1
Representación		
Voltaje de alimentación	3.3V ~ 5V	1,8 ~ 3,7 V
Soporte de batería	No dispone	Soporte 18650
Temperatura de funcionamiento	-40 °C ~ +85 °C	-40°C ~ +85°C
Pantalla: <ol style="list-style-type: none"> 1. Display 2. Tamaño 3. Resolución 4. Controlador 	OLED 0.96 pulgadas 128×64 píxeles SSD1306 dirección I2C 0x3C	No dispone No dispone No dispone No dispone
Lora 32: Frecuencia de funcionamiento Potencia de transmisión Error de frecuencia Espacio FIFO Modo de modulación Forma de interfaz	868 / 915 MHz + 20dBm +/- 15 KHz 64 bytes FSK, GFSK, MSK, GMSK, LoRa TM, OOK SPI	433 / 470 MHz +20dBm +/-15 KHz 64 bytes FSK,GFSK,MSK,GMSK,LoRa TM,OOK SPI

Características	LILYGO LoRa32 V1.6.1	ESP32 LORA 923 MHz con GPS NEO-M8N y soporte para batería 18650 v1.1
Alcance de transmisión de datos: 1. Exteriores con obstrucciones	Alcance de 1 a 3 km.	Alcance 100-200 m
Comunicaciones inalámbricas: 1. Wifi 2. Bluetooth 3. LoRa	Si Si Si	Si Si Si
Lenguajes de programación más comunes empleados: 1. C 2. C++ 3. MicroPython 4. JavaScrip	Si Si Si Si	Si Si Si Si

Para la selección de la placa óptima se emplearon los criterios más relevantes y de mayor impacto en el diseño del sistema de alarmas, donde se describió cada factor en la Tabla 10.

Tabla 10. Descripción de los factores para la selección de la placa de desarrollo.

Factor	Denominación	Descripción
F1	Lenguajes de programación	Indica los lenguajes o métodos utilizados para escribir las instrucciones que debe seguir los dispositivos, la información que debe enviar o recibir.
F2	Tipos de datos	Detalla las diferentes formas en que los dispositivos pueden manejar la información, números reales, booleanos , enteros, cadenas de texto plano. Dependiendo el tipo de comunicación que exista entre los dispositivos también se puede enviar videos, imágenes, mensajes de audio, entre otros.
F3	Alcance de transmisión de datos	Define la distancia máxima en la que el dispositivo o placa puede enviar y recibir datos de forma efectiva.
F4	Consumo de voltaje	Señala de manera puntual el rango de voltaje necesario para el funcionamiento adecuado de los dispositivos, así como la optimización que los componentes poseen para su desempeño normal.
F5	Compatibilidad	Indica la capacidad de la placa de desarrollo para integrarse con otros sistemas y tecnologías, asegurando su funcionamiento óptimo y permitiendo su escalabilidad en diversos entornos.
F6	Precio	Especifica el costo de adquisición de la placa de desarrollo, factor clave en la selección del dispositivo.

Factor	Denominación	Descripción
F7	Disponibilidad	Evalúa la facilidad y accesibilidad para obtener el dispositivo en el mercado, teniendo en cuenta adquisición inmediata, tiempos de entrega y la existencia en inventario.

Al analizar cada factor clave de una placa de desarrollo ESP 32, se elaboró una matriz de enfrentamiento que vincula los criterios en función de su importancia relativa, evaluándolos con una ponderación de 0,5 a 5 como se presenta en la Tabla 11 (la misma escala valorativa se aplica a todas las matrices realizadas).

Tabla 11. Escala de relevancia para la evaluación de criterios.

Descripción	Valor
Importancia máxima	5
Muy alta importancia	4
Alta importancia	3
Importancia moderada	2
Igual importancia	1
Baja importancia	0,5

Tabla 12. Ponderación de factores para placas ESP 32.

Ponderación de factores									
	F1	F2	F3	F4	F5	F6	F7	Total	Peso ponderado
F1		0,5	0,5	0,5	0,5	0,5	0,5	3	0,03
F2	3		0,5	1	1	0,5	1	7	0,07
F3	5	5		1	1	1	0,5	13,5	0,14
F4	4	4	3		1	1	0,5	13,5	0,14
F5	3	3	4	5		4	1	20	0,20
F6	3	5	3	3	0,5		1	15,5	0,16
F7	5	5	3	5	5	4		27	0,27
Total								99,5	1

Utilizando los pesos ponderados obtenidos en la Tabla 12, se evaluaron los factores en la escala de la Tabla 13, que se aplica a todos los dispositivos seleccionados. En este caso, se analizó y valoró el mejor dispositivo para su implementación en el sistema, tal como se muestra en la siguiente Tabla 14.

Tabla 13. Escala de evaluación para clasificar los factores de ponderación.

Descripción	Valores numéricos
Excelente	10
Sobresaliente	9
Muy notable	8
Notable	7
Bueno	6
Aprobado	5
Suficiente	4
Insuficiente	3
Deficiente	2
Muy deficiente	1

La Tabla 14 presenta una comparación entre las placas de desarrollo LILYGO LoRa32 V1.6.1 y la ESP32 LORA 923 MHz con GPS NEO-M8N y soporte para batería 18650 V1.1, basada en una evaluación ponderada de factores.

Tabla 14. Comparación de las placas de desarrollo ESP 32.

Factor	Peso Ponderado	LILYGO LoRa32 V1.6.1		ESP32 LORA 923 MHz con GPS NEO-M8N y soporte para batería 18650 V1.1	
		Calificación	Total	Calificación	Total
Lenguajes de programación	0,03	10	0,3	10	0,3
Tipos de datos	0,07	6	0,42	6	0,42
Alcance de transmisión de datos	0,14	10	1,4	2	0,28
Consumo de voltaje	0,14	10	1,4	3	0,42
Compatibilidad	0,2	8	1,6	6	1,2
Precio	0,16	10	1,6	10	1,6
Disponibilidad	0,27	1	0,27	1	0,27
Total			6,99		4,49

En primer lugar, el alcance de transmisión de datos, que es fundamental para aplicaciones de comunicación a distancia, refleja una ventaja para la LILYGO LoRa32 V1.6.1. En cuanto a los lenguajes de programación, ambas placas obtienen la máxima puntuación, lo que demuestra que tienen la misma flexibilidad en este ámbito. Respecto al tipo de datos, esencial para el manejo y procesamiento de información, ambas placas tienen un desempeño similar. En relación con el consumo de voltaje, los resultados reflejan que la placa LILYGO LoRa32 V1.6.1 tiene mayor optimización en sus componentes que la ESP32 LORA 923 MHz con GPS NEO-M8N y soporte para batería 18650 V1.1, traduciéndose en un menor consumo de voltaje a lo largo del tiempo. Por otra parte, en términos de compatibilidad, la LILYGO LoRa32 V1.6.1 destaca por su integración más amplia con diferentes sistemas. En cuanto al precio, ambas placas de desarrollo tienen la misma ponderación, lo que indica que no hay diferencias significativas en términos de costo entre las dos opciones. De manera similar, en el aspecto de disponibilidad, ambas unidades obtienen una calificación baja, lo que refleja que son de difícil adquisición o venta en el mercado.

3.3.2 Botón físico de activación

Se cuenta con dos botones físicos instalados en la planta baja y el piso 1 del taller tal como se muestra en la Figura 7, los mismos permiten la activación del sistema, debido a que están conectados a las entradas del circuito desarrollado.



Figura 7. Botones físicos de activación.

La Tabla 15 presenta un análisis detallado de las características técnicas de los botones empleados frente a otros modelos disponibles en el mercado, resaltando sus

puntos en común como sus diferencias. La selección de un modelo específico se realizó en función de los requerimientos del sistema y su aplicación.

Tabla 15. Características de los botones de emergencia.

Características	Botón de emergencia con llave S-BE01 alarma incendio	Botón de paro de emergencia con caja amarilla
Representación		
Dimensiones	(5 x 5.2 x 3.2) cm	(6.5 x 7.8 x 6.5) cm
Soporte de anclaje: 1. Dos tornillos 2. Cuatro tornillos	Si No dispone	Si No dispone
Peso:	0.038 kg	0.124 kg
Mecanismos de retorno a la posición inicial:	Una vez pulsado es necesario insertar la llave para poder volver a su estado inicial.	Una vez pulsado es necesario girar el botón para volver a su estado inicial.
Conexiones físicas disponibles: 1. Dos conexiones 2. Tres conexiones 3. Cuatro conexiones	Si Si No	Si No No
Estados operativos del botón	Dispone de 2 estados posibles: Posición normal Posición activada	Dispone de 3 estados posible: Posición normal Posición activada Retorno a la posición inicial
Rango de voltaje soportado	1 V ~ 24 V	1 V ~ 24 V

La Tabla 16 detalla los factores claves que se tuvieron en cuenta para evaluar los diferentes botones de emergencia y su funcionalidad. Cada factor proporciona una visión completa para seleccionar el componente más adecuada para el sistema.

Tabla 16. Descripción de los factores para la selección del botón.

Factor	Denominación	Descripción
F1	Tipos de mecanismos de retorno a la posición inicial	Define los sistemas o métodos que permiten que un botón, interruptor o dispositivo de control vuelva a su estado original después de haber sido activado.
F2	Número de conexiones físicas	Detalla la cantidad de terminales o puntos de conexión disponibles en el dispositivo.
F3	Estados operativos del botón	Señala los diferentes modos o condiciones en los que puede encontrarse el botón durante su funcionamiento.
F4	Voltaje soportado	Indica el rango de voltaje que el dispositivo puede soportar sin sufrir daños o afectar su funcionamiento.
F5	Precio	Especifica el costo de adquisición, factor clave en la selección del componente.
F6	Disponibilidad	Evalúa la facilidad y accesibilidad para obtener el dispositivo en el mercado, teniendo en cuenta adquisición inmediata, tiempos de entrega y la existencia en inventario.

Considerando los criterios expuestos en la Tabla 16, se obtuvo sus respectivos pesos ponderados en torno a la matriz de enfrentamiento como se puede apreciar en la Tabla 17.

Tabla 17. Ponderación de factores para el botón.

Ponderation de factores								
	F1	F2	F3	F4	F5	F6	Total	Peso ponderado
F1		3	4	2	3	0,5	12,5	0,17
F2	0,5		3	0,5	3	0,5	7,5	0,10
F3	0,5	0,5		0,5	3	0,5	5	0,07
F4	5	5	5		5	0,5	20,5	0,28
F5	0,5	0,5	0,5	0,5		0,5	2,5	0,03
F6	5	5	5	5	5		25	0,34
Total							73	1

Una vez obtenidos los valores ponderados de cada factor en la Tabla 17, se procedió a clasificar los botones de emergencia con llave S-BE01 para alarma de incendio y el

botón de paro de emergencia con caja amarilla según las características presentadas en la Tabla 16. A continuación, se compararon ambos dispositivos y se seleccionó el de mayor calificación, como se muestra en la Tabla 18.

Tabla 18. Comparación de los botones de emergencia.

Factor	Peso Ponderado	Botón de emergencia con llave S-BE01 alarma incendio		Botón de paro de emergencia con caja amarilla	
		Calificación	Total	Calificación	Total
Tipos de mecanismos de retorno a la posición inicial	0,17	10	1,7	8	1,36
Número de conexiones físicas	0,1	6	0,6	5	0,5
Estados operativos del botón	0,07	8	0,56	10	0,7
Voltaje soportado	0,28	6	1,68	6	1,68
Precio	0,03	10	0,3	10	0,3
Disponibilidad	0,34	9	3,06	6	2,04
Total			7,9		6,58

En Tabla 18 se comparó cuantitativamente el botón de emergencia con llave S-BE01 para alarma de incendio y el botón de paro de emergencia con caja amarilla, destacando diferencias clave en varios factores evaluados. En cuanto a los mecanismos de retorno a la posición inicial, el botón con llave S-BE01 obtuvo una calificación superior debido a su sistema de seguridad con llave, que ofrece mayor control y confiabilidad en comparación con un dispositivo de diferente accionamiento. Respecto al número de conexiones físicas, ambos dispositivos presentan valores similares, sin diferencias significativas en este aspecto. En el factor de estados operativos, el botón de paro de emergencia con caja amarilla se destaca, ya que sus características están diseñadas específicamente para satisfacer las demandas de entornos industriales. Por otro lado, en términos de voltaje soportado, ambos dispositivos muestran un rendimiento equivalente, dado que este parámetro está relacionado con las condiciones de

operación específicas de cada aplicación. En lo referente al precio, ambos botones presentan precios similares, lo que los hace igualmente accesibles desde el punto de vista económico. Finalmente, en cuanto a la disponibilidad, el botón S-BE01 para alarma de incendio obtiene una mejor puntuación, siendo más fácil de encontrar en el mercado en comparación con el botón de paro de emergencia. Este análisis demostró que el botón con llave ofrece ventajas moderadas en ciertos aspectos clave frente al botón de paro de emergencia, considerándose como la opción más adecuada.

3.3.3 Batería LiPo para respaldo de alimentación del sistema

El sistema cuenta con una batería LiPo instalada como respaldo, la cual garantiza la alimentación de la placa mas no la alimentación directa de la sirena, todo ello en caso de interrupciones en el suministro eléctrico principal. Esta batería está conectada de manera directa con el circuito, pues depende del mismo para mantener su nivel de voltaje tal como se muestra en la siguiente Figura 8.



Figura 8. Batería para el respaldo de energía.

La Tabla 19 muestra un análisis detallado de las características técnicas de las baterías LiPo consideradas para el respaldo en caso de interrupciones eléctricas, destacando sus similitudes como sus diferencias. La elección de un modelo se llevó a cabo considerando las necesidades del sistema y su aplicación particular.

Tabla 19. Características de las baterías LiPo.

Características	Batería Lipo de 7.4V 2S 80C 5200mAh	Batería LiPo 7.4V 3500 mAh 25C
Representación		
Dimensiones	(140 x 47 x 25) mm	(136 x 42 x 16) mm
Enfunches: 1. De carga 2. De descarga	Si Si	Si Si
Peso:	283,5 g	215 g
Atributos: 1. Tipo de batería 2. Voltaje 3. Capacidad 4. # de celdas 5. Tensión máxima por celda 6. Velocidad de carga	Batería LiPo 7,4 V 5200 mAh 2 3,7V ~ 4,2V 1C (3C max)	Batería LiPo 7,4 V 3500 mAh 2 4,2V 1C (3C max)
Capacidad máxima de descarga continua	80C significa que puede entregar una corriente máxima de descarga de hasta 80 veces su capacidad nominal.	25C significa que puede entregar una corriente máxima de descarga de hasta 25 veces su capacidad nominal.

Para la selección de la batería se evaluaron los criterios más relevantes y de mayor impacto en el diseño del sistema de alarmas, los cuales se detallan en la Tabla 20, considerando aspectos clave que afectan el rendimiento y la eficiencia del mismo.

Tabla 20. Descripción de los factores para la selección de la batería.

Factor	Denominación	Descripción
F1	Capacidad de carga	Define la cantidad de energía que la batería puede almacenar, crucial para determinar el tiempo de duración.
F2	Tiempo de carga	Especifica el tiempo necesario para cargar completamente la batería. Este factor influye en la eficiencia operativa y la disponibilidad del sistema.
F3	Peso y Tamaño	Señala la adecuación física de la batería para la instalación en el sistema.
F4	Ciclo de vida y confiabilidad	Indica la vida útil de la batería, incluyendo la cantidad de ciclos de carga y descarga antes de que la capacidad se vea significativamente reducida.

Factor	Denominación	Descripción
F5	Precio	Especifica el costo de adquisición, factor clave en la selección del componente.
F6	Disponibilidad	Evalúa la facilidad y accesibilidad para obtener el dispositivo en el mercado, teniendo en cuenta adquisición inmediata, tiempos de entrega y la existencia en inventario.

En base a los criterios establecidos en la Tabla 20, se calcularon los pesos ponderados correspondientes a través de la matriz de enfrentamiento, como se muestra en la Tabla 21.

Tabla 21. Ponderación de factores para la batería de respaldo.

Ponderación de factores								
	F1	F2	F3	F4	F5	F6	Total	Peso ponderado
F1		1	3	0,5	5	0,5	10	0,14
F2	1		5	0,5	3	0,5	10	0,14
F3	0,5	0,5		0,5	0,5	0,5	2,5	0,04
F4	5	5	5		3	0,5	18,5	0,26
F5	0,5	0,5	3	0,5		0,5	5	0,07
F6	5	5	5	5	5		25	0,35
Total							71	1

Una vez calculado los valores ponderados de cada factor en la Tabla 21, se procedió a evaluar las baterías LiPo conforme a los factores descritos en la Tabla 20, Permitiendo realizar una comparación entre los distintos modelos y se eligió el de mayor puntuación, tal como se muestra en la Tabla 22.

Tabla 22. Comparación cuantitativa entre las baterías de respaldo.

Factor	Peso Ponderado	Batería Lipo de 7.4V 2S 80C 5200mAh		Batería LiPo 7.4V 3500mAh 25C	
		Calificación	Total	Calificación	Total
Capacidad de carga	0,14	10	1,4	6	0,84
Tiempo de carga	0,14	7	0,98	7	0,98
Peso y Tamaño	0,04	6	0,24	6	0,24
Ciclo de vida y confiabilidad	0,26	10	2,6	10	2,6
Precio	0,07	6	0,42	6	0,42
Disponibilidad	0,35	4	1,4	4	1,4
Total			7,04		6,48

En la Tabla 22 se muestra una comparación cuantitativa entre las baterías LiPo de 7.4V 2S 80C 5200mAh y LiPo de 7.4V 3500mAh 25C, resaltando las diferencias y similitudes clave en varios factores críticos. En cuanto a la capacidad de carga, la batería LiPo de 7.4V 2S 80C 5200mAh muestra una ventaja significativa, lo que implica un mayor almacenamiento de energía para aplicaciones de mayor demanda. En lo referente al tiempo de carga, ambas baterías muestran un rendimiento comparable, sin diferencias notables que puedan influir en la selección. En cuanto al peso y tamaño, las dos baterías presentan calificaciones similares, dado que este factor depende directamente de los requisitos específicos del sistema y su entorno operativo. En términos de ciclo de vida y confiabilidad, ambos modelos presentan un rendimiento equivalente, lo que indica que ambos son igualmente confiables bajo condiciones de uso similares. En lo que respecta al precio, las dos baterías tienen valores aproximados, lo que las hace opciones viables desde el punto de vista económico. Finalmente, en cuanto a la disponibilidad, ambos modelos muestran escalas similares de accesibilidad en el mercado, lo que sugiere que ambos están igualmente disponibles para su adquisición. En este análisis se evidenció que, aunque ambas baterías presentan

características y capacidades comparables, la batería LiPo de 7.4V 2S 80C 5200mAh ofrece ventajas notables en términos de capacidad de carga, lo que la posiciona como la opción más adecuada para su incorporación al sistema.

3.3.4 Dispositivo de notificación “Sirena”

La divulgación de notificaciones y alarmas se realiza a través de una bocina integrada al sistema para emitir señales acústicas claras y de alto alcance. Con lo que se busca garantizar la comunicación inmediata y efectiva del evento. Tal y como se lo puede apreciar en la Figura 9.



Figura 9. Sirena física de notificaciones.

La Tabla 23 presenta un análisis detallado de las especificaciones técnicas de las bocinas sirenas consideradas para el sistema, destacando tanto sus puntos en común como sus diferencias clave. La selección del modelo más adecuado se realizó tomando en cuenta los requisitos específicos del sistema y en entorno de aplicación.

Tabla 23. Características de las sirenas.

Características	Megáfono Bocina Sirena 5 tonos 100 w	Altavoz de sirena 3 tonos 50 w
Representación		

Características	Megáfono Bocina Sirena 5 tonos 100 w	Altavoz de sirena 3 tonos 50 w
Dimensiones	(14 x 14 x 12) cm	(15 x 13 x 13) cm
Números de sonidos de sirena	5 Sonidos de sirena: Hooter Ambulance Traffic police Fire alarm	3 Sonidos de sirena: Wail Horn Siren
Alimentación	12 V DC	12 V DC
Potencia	100 W Produce sonidos muy altos, cercano a los 300 db a todo volumen.	50 W Produce sonidos altos, cercano a los 112 db a todo volumen.
Disponibilidad de micrófono	Si	Si

Para la selección de la sirena se evaluaron los criterios más relevantes y de mayor impacto en el diseño del sistema de alarmas, los cuales se detallan en la Tabla 24, considerando aspectos clave que afectan el rendimiento y la eficiencia del mismo.

Tabla 24. Descripción de los factores para la selección de la sirena.

Factor	Denominación	Descripción
F1	Alimentación	Define de manera puntual el voltaje necesario para el funcionamiento adecuado de los dispositivos.
F2	Tonos	Detalla los diferentes patrones sonoros o tonos para distintos tipos de emergencias o notificaciones.
F3	Peso y Tamaño	Señala la adecuación física del componente para la instalación en el sistema.
F4	Potencia	Indica el nivel de presión sonora que puede alcanzar, asegurando que sea audible en el rango deseado o la distancia contemplada.
F5	Materiales de construcción	Relaciona los componentes con los cuales fue fabricado y ensamblado en dispositivo (plástico, metal, etc).
F6	Precio	Especifica el costo de adquisición, factor clave en la selección del componente.
F7	Disponibilidad	Evalúa la facilidad y accesibilidad para obtener el dispositivo en el mercado, teniendo en cuenta adquisición inmediata, tiempos de entrega y la existencia en inventario.

En base a los criterios establecidos en la Tabla 24, se calcularon los pesos ponderados correspondientes a través de la matriz de enfrentamiento, como se muestra en la Tabla 25.

Tabla 25. Ponderación de factores para la sirena.

Ponderación de factores									
	F1	F2	F3	F4	F5	F6	F7	Total	Peso ponderado
F1		3	2	0,5	2	3	0,5	11	0,12
F2	0,5		2	0,5	2	2	0,5	7,5	0,08
F3	1	0,5		0,5	0,5	3	0,5	6	0,07
F4	5	5	5		5	5	0,5	25,5	0,28
F5	0,5	0,5	3	0,5		3	0,5	8	0,09
F6	0,5	0,5	0,5	0,5	0,5		0,5	3	0,03
F7	5	5	5	5	5	5		30	0,33
Total								91	1

Una vez calculado los valores ponderados de cada factor en la Tabla 25, se procedió a evaluar los diferentes modelos de sirenas conforme a los factores descritos en la Tabla 24. Se compararon detalladamente las características y el rendimiento de cada modelo frente a los criterios establecidos. Para ello, se utilizó un enfoque basado en la asignación de puntuaciones a cada modelo, las cuales reflejan su desempeño relativo en cada factor, tal como se muestra en la Tabla 26.

Tabla 26. Comparación cuantitativa entre los tipos de sirenas.

Factor	Peso Ponderado	Megáfono Bocina Sirena 5 tonos 100 w		Altavoz de sirena 3 tonos 50 w	
		Calificación	Total	Calificación	Total
Alimentación	0,12	10	1,2	10	1,2
Tonos	0,08	7	0,56	4	0,32
Peso y Tamaño	0,07	6	0,42	6	0,42
Potencia	0,28	10	2,8	6	1,68
Materiales de construcción	0,09	5	0,45	5	0,45
Precio	0,03	10	0,3	10	0,3
Disponibilidad	0,33	5	1,65	5	1,65
Total			7,38		6,02

En la Tabla 26 se muestra una comparación cuantitativa entre el megáfono bocina sirena 5 tonos 100 w y el altavoz de sirena 3 tonos 50 w, resaltando las diferencias y similitudes clave en varios factores críticos. En cuanto al factor de alimentación, ambos modelos de sirena obtienen calificaciones similares, favoreciendo igualitariamente para cada dispositivo en este factor. Respecto a los tonos, el megáfono bocina sirena de 5 tonos y 100 W muestra una ventaja, ofreciendo mayor variedad de sonidos, lo que lo posiciona por encima del altavoz de sirena de 3 tonos y 50 W. En lo referente a peso y tamaño, ambos dispositivos tienen un desempeño comparable, ya que ambos obtienen calificaciones iguales, planteando que este factor no es determinante para la selección entre ellos. En torno a potencia, el megáfono bocina destaca significativamente con una calificación mucho mayor, lo que implica un rendimiento superior en términos de alcance del sonido. En relación a los materiales de construcción, ambos dispositivos muestran una confiabilidad similar, con

puntuaciones idénticas que indican resistencia adecuada bajo condiciones normales de uso. Desde el punto de vista del precio, ambos dispositivos son igualmente atractivos económicamente, con valores similares que no influyen considerablemente en la decisión de compra. Finalmente, en el factor de disponibilidad, ambos dispositivos presentan una accesibilidad equivalente en el mercado, reflejada en sus calificaciones iguales. En base al análisis se evidenció que, si bien es cierto que los dispositivos discrepan en varios factores, la funcionalidad y la selección de cada uno depende de la aplicación que se esté desarrollando, motivo por el cual se seleccionó el megáfono bocina sirena 5 tonos 100 w.

3.3.5 Placa electrónica

Se diseñó el circuito utilizando el software Proteus, donde se llevaron a cabo simulaciones para verificar y validar las conexiones eléctricas. Este proceso de simulación fue crucial, ya que permitió anticipar y resolver posibles problemas asegurando así un funcionamiento óptimo del sistema bajo condiciones reales de operación.

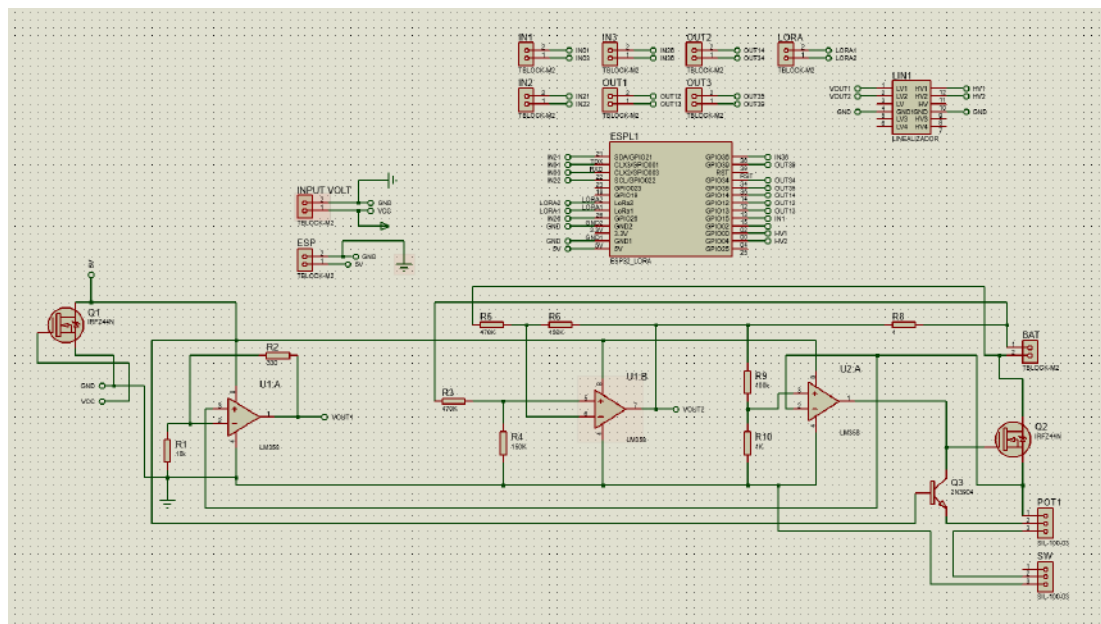


Figura 10. Esquema del conexionado de la placa base.

El diseño elaborado en Proteus, que se presenta en la Figura10, facilitó la realización de simulaciones que permitieron evaluar el rendimiento del sistema en diversas

condiciones operativas. Posteriormente, se procedió a la elaboración del diseño de la placa de circuito impreso (PCB), que se presenta en la Figura 11. Este diseño fue optimizado para garantizar una disposición adecuada de los componentes, con el fin de minimizar las interferencias y maximizar la eficiencia funcional del sistema. La cuidadosa disposición de los elementos en la PCB es fundamental para asegurar la integridad de las señales y la estabilidad del sistema. Al existir múltiples cruces en las líneas de conexión en los componentes de la placa, se optó por un modelo a doble cara, donde las líneas en color azul representan un lado de la placa y las líneas en color rojo muestran el otro lado, tal como se puede apreciar en la siguiente figura.

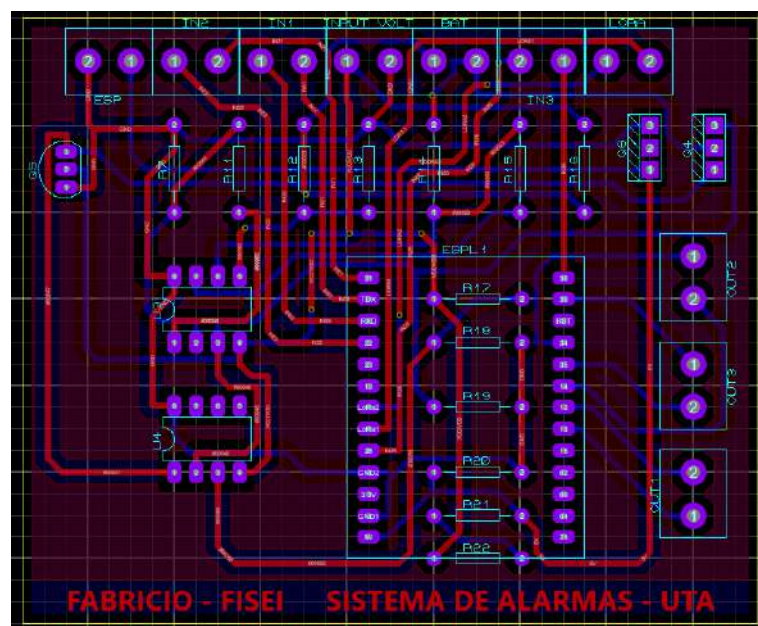


Figura 11. Diseño PCB de la placa.

Para el diseño presentado en la Figura 11, se han tomado en cuenta los lineamientos establecidos por la normas IPC-2221 “Norma general sobre diseños PBC” e IPC-2222 “Diseño de PCBs de doble cara y multicapa”, especialmente en lo que respecta a la estructura y construcción de las capas. Con el fin de optimizar el rendimiento y reducir costos, se ha minimizado la cantidad de vías a través de las capas, tal como lo recomienda la norma. Además, se ha seguido lo indicado por la norma en cuanto a la especificación de las dimensiones mínimas de los componentes, asegurando que sean compatibles con las capacidades de fabricación. Se ha prestado especial atención a la selección adecuada de materiales para las capas, considerando su idoneidad en términos de propiedades eléctricas y térmicas. De la misma forma se ha considerado

el tamaño de la pista en función de la corriente que debe soportar, la cual menciona que para una intensidad de 2 amperios, se puede aplicar un ancho de pista de 0,8 mm. En cuanto, al voltaje que se está suministrando, la normativa recomienda una separación de pistas de 0,317 mm para voltajes de 0 a 30 v. Finalmente, el tratamiento de las vías de conexión se ha realizado según las directrices de la norma para garantizar la integridad del diseño y la fiabilidad [36][37].

Para mejorar la apreciación final de la placa, el software Proteus ofrece una opción de visualización en 3D, que representa de manera realista el producto final que se está diseñando. La figura a continuación muestra el resultado terminado, integrado con los componentes previamente mencionados, lo que permite una evaluación más efectiva del diseño y su funcionalidad. Esta visualización facilita la identificación de posibles ajustes y optimizaciones antes de la fabricación del prototipo.

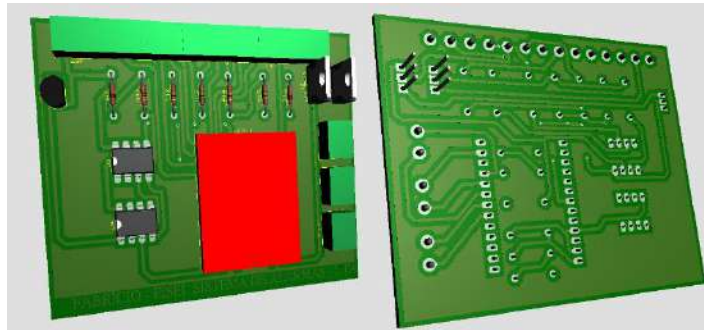


Figura 12. Visualización 3D de la placa.

3.3.6 Construcción de la placa a partir del diseño PCB

En este apartado se presenta el resultado final del diseño PCB, desde la conceptualización inicial hasta la fabricación de una placa funcional de baquelita a doble cara. Este material fue seleccionado por sus propiedades de resistencia mecánica y durabilidad, lo que lo convierte en una base ideal para la integración de los componentes electrónicos necesarios para el correcto funcionamiento del circuito. Un diseño bien estructurado no solo define el esquema eléctrico, sino que también traduce esta planificación en un modelo físico que será replicado en la fabricación de la placa.

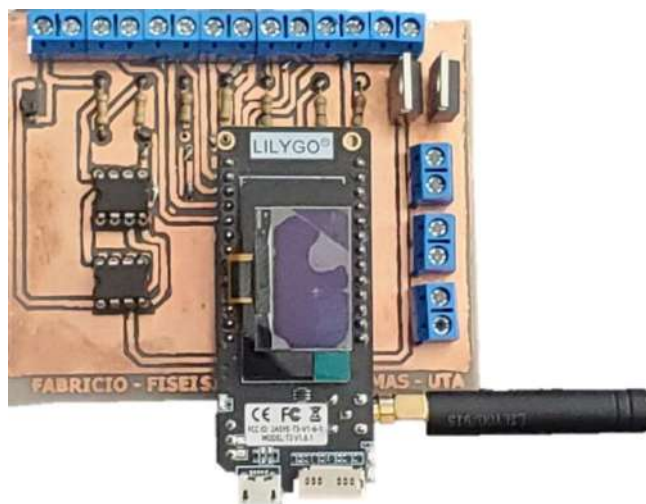


Figura 13. Placa final elaborada.

La Figura 13 presenta el diseño que se obtuvo al finalizar la construcción del circuito, donde se puede apreciar de manera real la integración de los componentes, las múltiples comunicaciones que existen entre ellos y sus interactividad con la placa de desarrollo. Cada pista posibilita la comunicación con elementos específicos: resistencias de diferentes valores, aplicadores operacionales, placa de desarrollo y borneras de comunicación para las entradas y salidas del circuito. El mismo fue valorado en base a la normativa IPC-A-610 “*Estándar para la inspección de ensamblajes electrónicos*”, la cual se aplicó para garantizar que las uniones soldadas sean uniformes y estén libres de grietas, excesos o puentes que puedan generar cortocircuitos. Asimismo, se consideró la sección dedicada a componentes montados, verificando que estén alineados correctamente, sin daños visibles y firmemente fijados a sus vías, asegurando un rendimiento óptimo durante la operación [38].

3.3.7 Carcasa de alojamiento

Para el alojamiento de las placas electrónicas, se diseñó y fabricó una base utilizando TPU (poliuretano termoplástico), todo ello en base a las herramientas de modelado que ofrece el software Fusión 360. Esta estructura no solo proporciona un soporte sólido para los componentes electrónicos, sino que también ofrece una protección eficaz contra factores ambientales, como la humedad, el polvo y la temperatura. Su diseño asegura que los componentes permanezcan aislados de la intemperie, garantizando una funcionalidad continua y estable

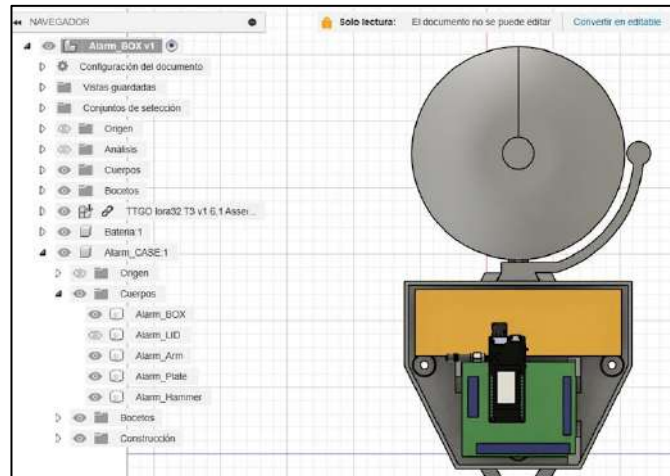


Figura 14. Carcasa de alojamiento de las placas.

El mismo fue desarrollado en base a diferentes componentes que deben ir integrados de manera conjunta dentro del sistema de alarmas, para ello cada parte fue generada de manera única, permitiendo la flexibilidad del ensamble final.

3.3.8 Señalética de seguridad

Para relacionar de mejor manera los botones físicos con el sistema de alarmas, se implementó una señalética de seguridad diseñada conforme al reglamento interno de seguridad y salud de los trabajadores y mejoramiento del medio ambiente del trabajo de la secretaria de derechos humanos. La Figura 15 presenta el diseño empleado.



Figura 15. Señalética de seguridad.

La figura antes presentada sigue los parámetros establecidos en el Art.19. Señalización de seguridad, apartado e, del presente reglamento. Donde se menciona que “*señales de salvamento o socorro. Son de forma cuadrada o rectangular, con pictograma blanco sobre fondo verde, legibles a 3 metros de distancia*” [39].

3.4 Segunda fase

3.4.1 Conexionado eléctrico de los componentes físicos del sistema

Para garantizar la correcta implementación de las conexiones eléctricas del sistema, se utilizó el software CadeSimu, una herramienta que permitió diseñar y simular el diagrama eléctrico. Este diagrama sirvió no solo como una guía precisa para las conexiones físicas del sistema, sino también como un mecanismo preventivo para evitar errores comunes, como uniones incorrectas o posibles cortocircuitos que podrían comprometer la funcionalidad del sistema. El diseño incluyó los principales componentes como la estación contra incendios y detección de humo, botones de activación manual, la sirena para alertas sonoras, y la batería de respaldo para asegurar la operatividad en caso de fallos en el suministro eléctrico principal.

A continuación, en la Figura 16, se presenta el diagrama eléctrico que detalla las conexiones realizadas entre los diferentes componente y la placa elaborada, ofreciendo una representación visual clara y comprensible.

a. Entradas físicas

La Figura 17 presenta el esquema eléctrico que detalla la alimentación de la placa, la conexión de las entradas físicas provenientes de los pulsadores y la estación contra incendios. También se destaca la fuente de respaldo proporcionada por la batería LiPo, que asegura el funcionamiento en caso de interrupciones del suministro eléctrico.

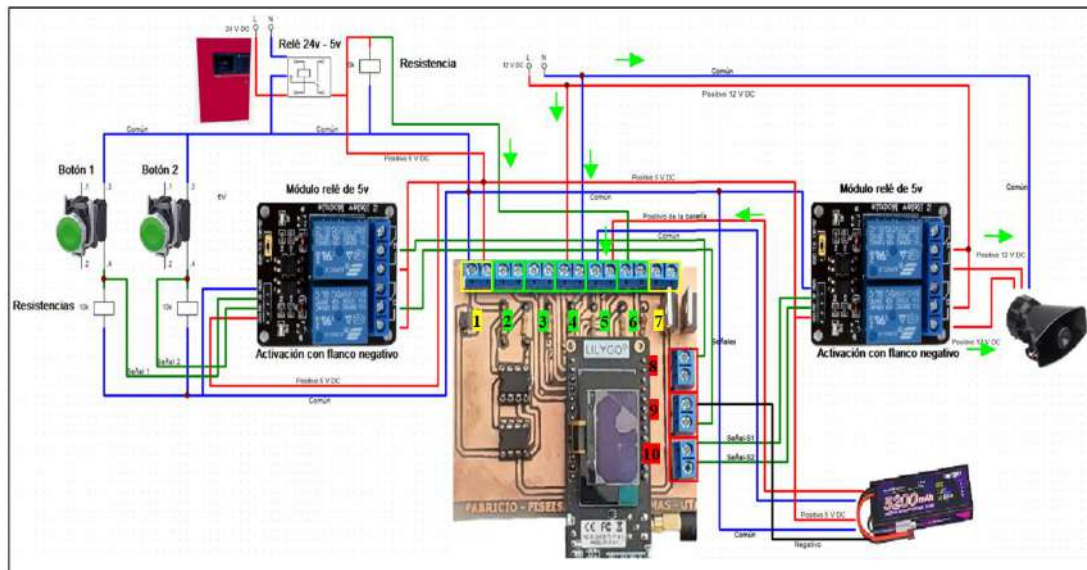


Figura 17. Diagrama de conexionado entradas físicas.

Las entradas al circuito están conformadas por las siguientes borneras, los cuales se detallan en la Tabla 27.

Tabla 27. Descripción de las entradas físicas de la placa.

Número de bornera	Componente	Descripción
2	Bornera de respaldo	Entrada adicional predispuesta para el sistema en caso de faltar conexiones entre la placa y sus componentes.
3	Bornera de respaldo	Entrada adicional predispuesta para el sistema en caso de faltar conexiones entre la placa y sus componentes.
4	Fuente de alimentación de 12V DC	Suministra la energía necesaria para la alimentación del circuito, así también la alimentación de la sirena de manera directa.
5	Batería de respaldo para alimentación	Dispositivo que proporciona energía al circuito electrónico durante interrupciones en el suministro eléctrico.

Número de bornera	Componente	Descripción
6	Estación contra incendios y detección de humo, señal de activación	Equipo que permite la detección de incendios, activa una de sus salidas de 24 V DC, la cual es acondicionada a través instrumentos de 24 V DC a 5V DC, la misma que es enviada a una entra física del sistema, para su activación.

b. Salidas físicas

La Figura 18 muestra el esquema de conexión de las salidas físicas del circuito electrónico a través de las borneras. Estas borneras actúan como puntos de conexión para distribuir las señales y voltajes provenientes de la placa electrónica hacia los diferentes dispositivos externos, como la sirena, los botones de control y la batería de respaldo.

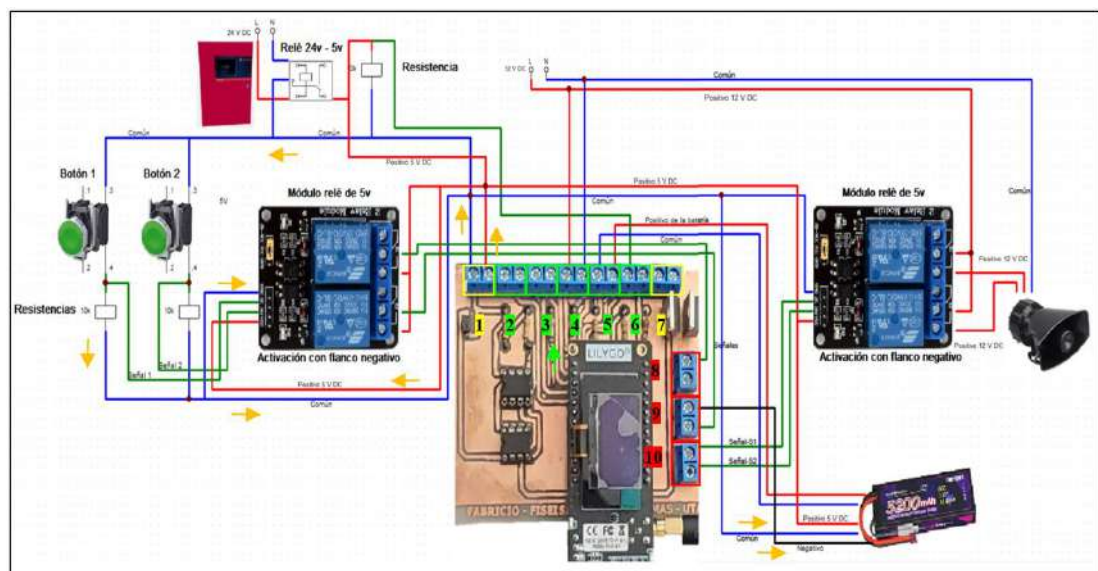


Figura 18. Diagrama de conexiones de salidas físicas.

Las salidas del circuito están conformadas por las siguientes borneras, los cuales se detallan en la Tabla 28.

Tabla 28. Descripción de las salidas físicas de la placa.

Número de bornera	Salida	Descripción
1	5 V DC, y GND desde la placa	Habilita la alimentación de los botones y los módulos relé, teniendo en cuenta el rango de voltaje con el cual se está trabajando.

Número de bornera	Salida	Descripción
7	Bornera de respaldo	Salida adicional predispuesta para el sistema en caso de faltar conexiones entre la placa y sus componentes.

c. Pines que trabajan como entradas y salidas

Denominados con la abreviatura GPIO son pines configurables que se pueden usar como entradas o salidas digitales según las necesidades del usuario. Sirven para interactuar con sensores, actuadores, displays y otros dispositivos externos, que requieran diversas modalidades de conexionado. La Figura 19 muestra el esquema de conexión de los pines GPIO con sus respectivos elementos.

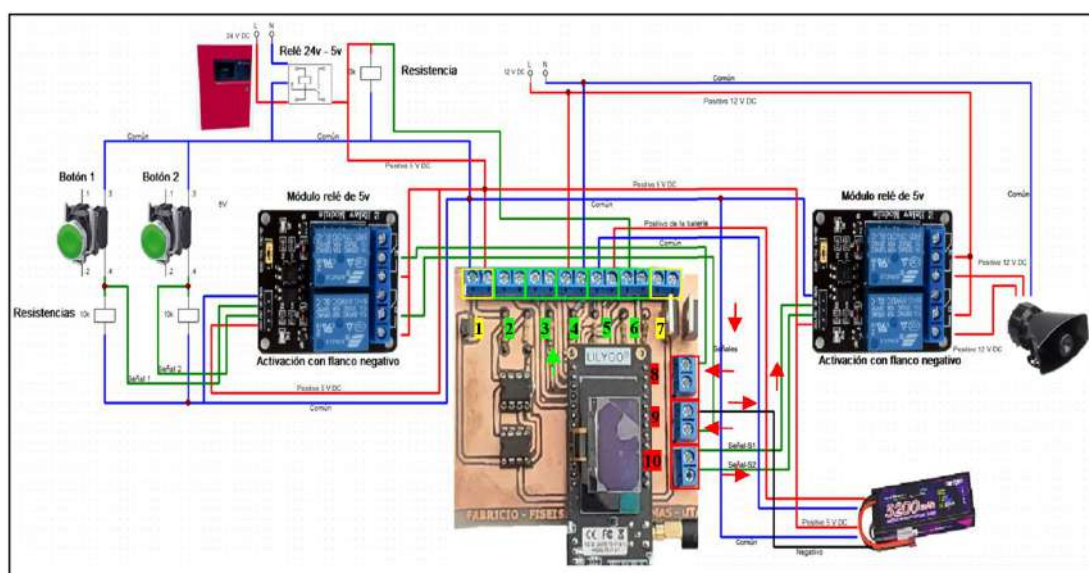


Figura 19. Diagrama de conexionado de los pines GPIO.

Los pines GPIO están conformadas por las siguientes borneras, los cuales se detallan en la Tabla 29.

Tabla 29. Descripción de los pines GPIO.

Número de bornera	Entrada/Salida	Descripción
8	Entrada de la señal del botón 1, primera ranura de la bornera.	Permite el ingreso de la señal que envía el botón 1, instalado en la planta baja del taller.
9	Salida empleada como negativo de la placa, la cual se conecta con la batería de respaldo, primera entrada de la bornera. Entrada de la señal del botón 2, segunda ranura de la bornera.	La primera conexión permite cerrar el circuito de carga de la batería LiPo en conjunto con las demás líneas requeridas.

Número de bornera	Entrada/Salida	Descripción
		La segunda conexión permite el ingreso de la señal que envía el botón 2, instalado en la piso 1 de taller.
10	Salidas empleadas para la activación de la sirena “tono 1 y tono 2”, en base al relé de conexión.	Permite la activación del relé en flanco negativo cerrando el circuito de alimentación de la sirena de 12 V DC, mediante el cual se controla el tono empleado.

3.4.2 Flujograma del sistema

Una vez que se seleccionó los componentes necesarios y se estableció las conexiones entre los diversos dispositivos y la placa diseñada, se procedió a desarrollar un diagrama de flujo. Este diagrama tiene como propósito principal representar de manera clara y ordenada el funcionamiento del sistema. En él se detallaron las interacciones de las etapas del proceso y la lógica operativa que definió el comportamiento del sistema.

La Figura 20 presentada a continuación permitió comprender la secuencia de operaciones, identificar posibles puntos de mejora y garantizar que todos los elementos trabajen de manera coordinada para cumplir los objetivos del diseño.

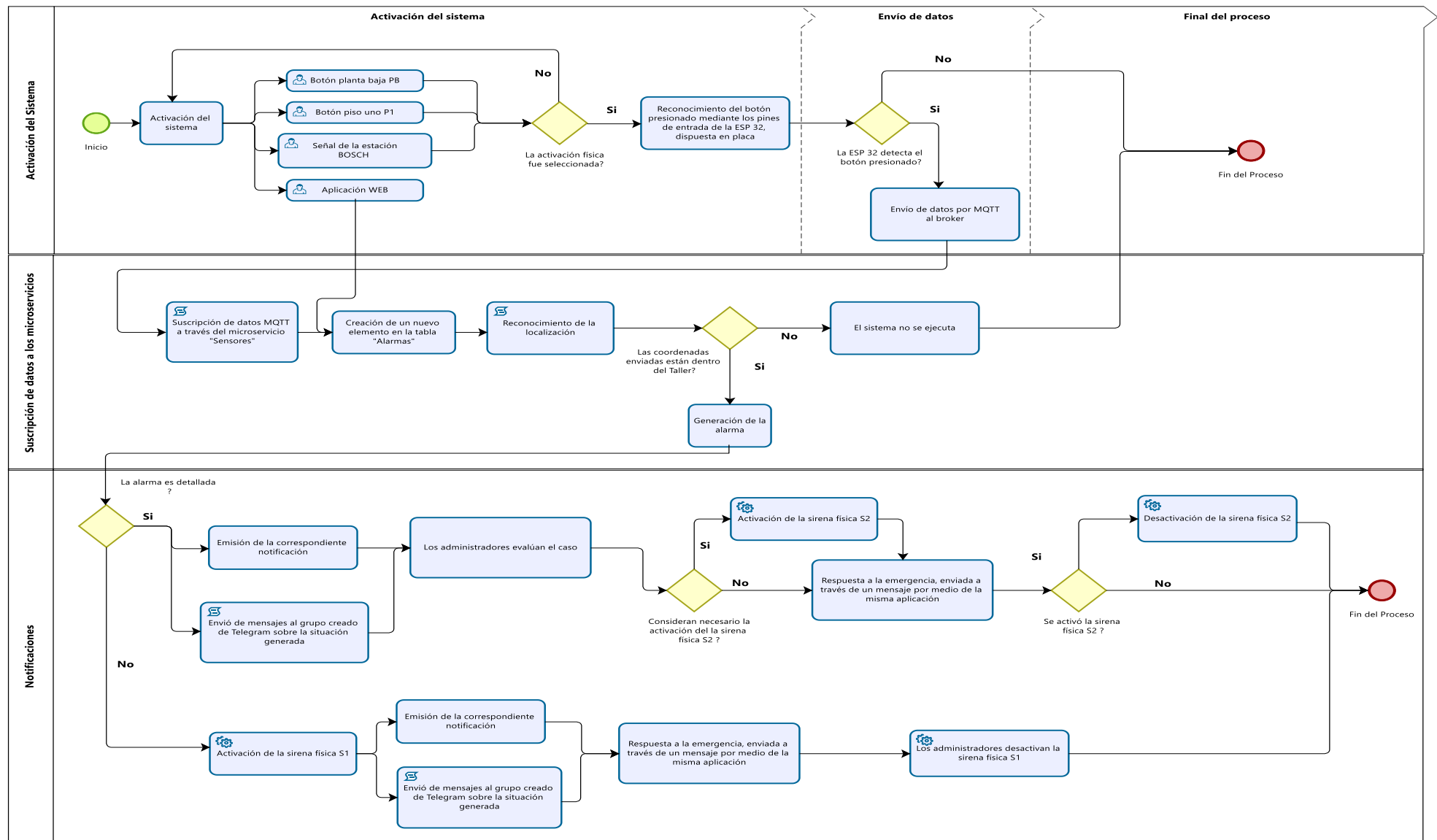


Figura 20. Diagrama de flujo del sistema.

3.5 Tercera fase

3.5.1 Programación y aplicación web

a. Detección de las alarmas generadas

Para iniciar con la detección de la alarma generada de manera física “botón de emergencia” o remota a través de la aplicación móvil, se desarrolló un programa en Visual Studio Code empleando C++, el mismo que fue cargado a la tarjeta de desarrollo Esp 32, quien a través de la conexión Wifi envía la información a la base de datos. En las figuras siguientes se presentan y explican secciones relacionadas con el Anexo A, el cual contiene la programación completa del sistema.

```
1  #include <Wifi.h>
2  #include <PubSubClient.h>
3  #include <Wire.h>
4  #include <Adafruit_GFX.h>
5  #include <Adafruit_SSD1306.h>
6
7  // Configuración Wifi
8  const char* ssidList[] = {"alarmaTT", "FISEI-Wifi", "Lizbeth"};
9  const char* passwordList[] = {"12345678", "", "VeritoCaiza2001"};
10
11 // Configuración del servidor MQTT
12 const char* mqtt_server = "74.***.***.***";
13 const char* mqtt_user = "p***";
14 const char* mqtt_password = "*****";
```

Librerías empleadas en la programación del código en función de las necesidades para la detección de la alarma generada.

Redes Wifi disponibles para la conexión.

Configuración de las credenciales del servidor MQTT a emplearse

Figura 21. Programa para la detección de alarmas.

```
16 // Tópicos MQTT
17 const char* topic_P1_S1 = "huachi/fisei/talleres/P1_S1";
18 const char* topic_P1_S2 = "huachi/fisei/talleres/P1_S2";
19 const char* topic_PB_BP = "huachi/fisei/talleres/PB_BP";
20 const char* topic_P1_BP = "huachi/fisei/talleres/P1_BP";
21 const char* topic_P1_AI = "huachi/fisei/talleres/P1_AI";
22
```

Tópicos que se encarga de la monitorización del estado de cada alarma.

Figura 22. Definición de los tópicos para la comunicación MQTT.

Los tópicos definidos en el código son utilizados para organizar la comunicación entre diferentes dispositivos a través del protocolo MQTT. Cada uno de estos tópicos corresponde a una ubicación dentro del sistema de monitoreo. La Tabla 30 presenta una descripción detalla de la simbología utilizada.

Tabla 30. Simbología utilizada.

Simbología empleada en los tópicos	
Acrónimo	Significado
P1_S1	Hace referencia al primer tono de la sirena “S1”, la misma que está instalada en el piso número uno del taller “P1”
P1_S2	Relacionado con el segundo tono “S2” que la sirena debe activar en caso de una emergencia en el piso uno del taller.
PB_BP	Vinculado directamente con el botón físico “BP” de activación del sistema instalado en la planta baja “PB” del taller.
P1_BP	Asociado con el botón físico “BP” de activación del sistema instalado en el piso uno del taller “P1”.
P1_AI	Enlazado con la estación contra incendios y detección de humo “AI” en el primer piso del taller “P1”.

Para la poder visualizar el estado de los botones y la sirena, se imprimen el pantalla OLED texto relacionado con el funcionamiento de cada dispositivo. En la Figura 23, se presenta la configuración que debe tener la pantalla para su correcto funcionamiento.

```

35 // Configuración OLED
36 #define SCREEN_WIDTH 128 } Configuración para el funcionamiento y visualización de
37 #define SCREEN_HEIGHT 64 } los mensajes en la pantalla OLED
38 #define OLED_RESET -1
39 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &wire, OLED_RESET);
40

```

Figura 23. Configuración de la pantalla OLED.

Una vez activada la situación o emergencia, solo los administradores pueden desactivar la sirena de notificación a través del servidor, idea por la cual se agregó las siguientes líneas en el código principal mostrado en la Figura 24.

```

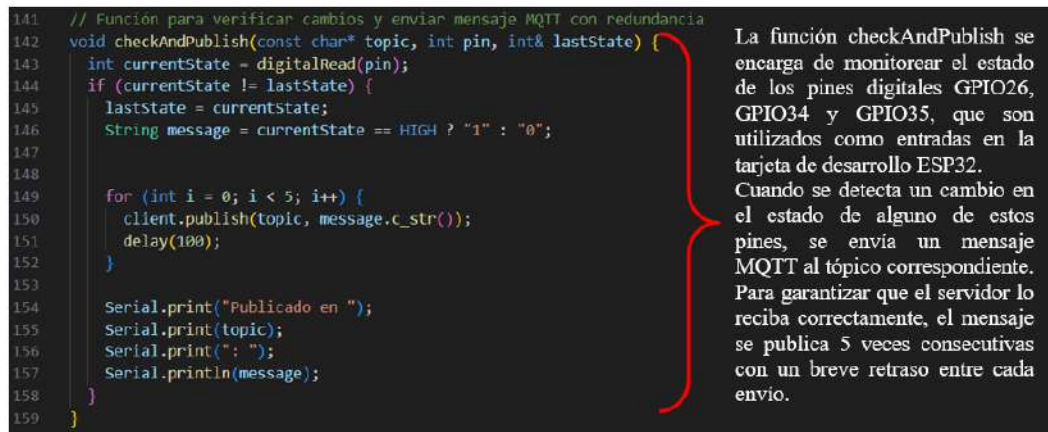
122 void callback(char* topic, byte* payload, unsigned int length) {
123     String message;
124     for (int i = 0; i < length; i++) {
125         message += (char)payload[i];
126     }
127
128     if (String(topic) == topic_P1_S1) {
129         digitalWrite(GPIO12, message == "1" ? LOW : HIGH);
130         Serial.print("GPIO12: ");
131         Serial.println(message == "1" ? "ENCENDIDO" : "APAGADO");
132     }
133
134     if (String(topic) == topic_P1_S2) {
135         digitalWrite(GPIO13, message == "1" ? LOW : HIGH);
136         Serial.print("GPIO13: ");
137         Serial.println(message == "1" ? "ENCENDIDO" : "APAGADO");
138     }
139 }

```

La función callback permite recibir mensajes MQTT y, según el tipo de mensaje, controlar los pines GPIO12 y GPIO13, los cuales se utilizan para desactivar la alarma. Útil cuando la sirena se activa mediante los botones físicos, la estación contra incendios o la aplicación móvil. El mismo emplea lógica inversa donde “1” activa el pin en “Low” y “0” desactiva el pin en “High”.

Figura 24. Desactivación remota de los pines para el control de la sirena.

El cambio de estado de los pines digitales empleados como entradas al sistema son controlados y monitorizados a través de las siguientes líneas de programación, las cuales se pueden apreciar en la Figura 25.



```

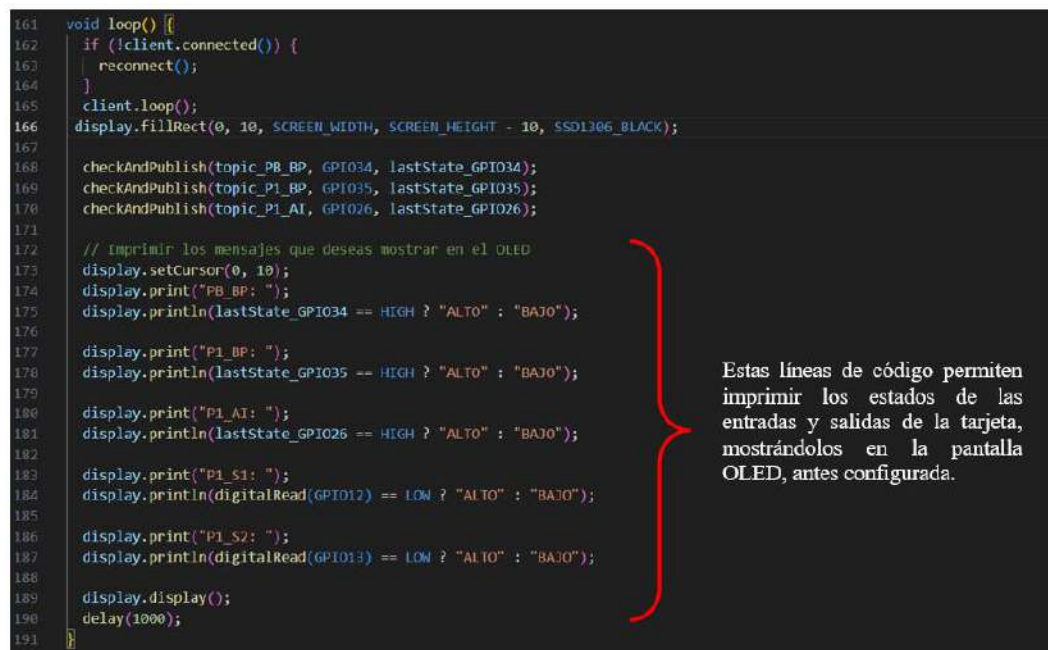
141 // Funcion para verificar cambios y enviar mensaje MQTT con redundancia
142 void checkAndPublish(const char* topic, int pin, int& lastState) {
143     int currentState = digitalRead(pin);
144     if (currentState != lastState) {
145         lastState = currentState;
146         String message = currentState == HIGH ? "1" : "0";
147
148         for (int i = 0; i < 5; i++) {
149             client.publish(topic, message.c_str());
150             delay(100);
151         }
152
153         Serial.print("Publicado en ");
154         Serial.print(topic);
155         Serial.print(": ");
156         Serial.println(message);
157     }
158 }
159

```

La función `checkAndPublish` se encarga de monitorear el estado de los pines digitales GPIO26, GPIO34 y GPIO35, que son utilizados como entradas en la tarjeta de desarrollo ESP32. Cuando se detecta un cambio en el estado de alguno de estos pines, se envía un mensaje MQTT al tópico correspondiente. Para garantizar que el servidor lo reciba correctamente, el mensaje se publica 5 veces consecutivas con un breve retraso entre cada envío.

Figura 25. Control en los cambios de las entradas digitales.

Finalmente la actualización de la pantalla OLED se realizada cada segundo, teniendo en cuenta el estado actual de las entradas y salidas de la ESP32, todo ello presentado en texto a través de la pantalla. En la siguiente figura se puede apreciar el código dispuesto para este apartado.



```

161 void loop() {
162     if (!client.connected()) {
163         reconnect();
164     }
165     client.loop();
166     display.fillRect(0, 10, SCREEN_WIDTH, SCREEN_HEIGHT - 10, SSD1306_BLACK);
167
168     checkAndPublish(topic_PB_BP, GPIO34, lastState_GPIO34);
169     checkAndPublish(topic_P1_BP, GPIO35, lastState_GPIO35);
170     checkAndPublish(topic_P1_AI, GPIO26, lastState_GPIO26);
171
172     // Imprimir los mensajes que deseas mostrar en el OLED
173     display.setCursor(0, 10);
174     display.print("PB_BP: ");
175     display.println(lastState_GPIO34 == HIGH ? "ALTO" : "BAJO");
176
177     display.print("P1_BP: ");
178     display.println(lastState_GPIO35 == HIGH ? "ALTO" : "BAJO");
179
180     display.print("P1_AI: ");
181     display.println(lastState_GPIO26 == HIGH ? "ALTO" : "BAJO");
182
183     display.print("P1_S1: ");
184     display.println(digitalRead(GPIO12) == LOW ? "ALTO" : "BAJO");
185
186     display.print("P1_S2: ");
187     display.println(digitalRead(GPIO13) == LOW ? "ALTO" : "BAJO");
188
189     display.display();
190     delay(1000);
191 }

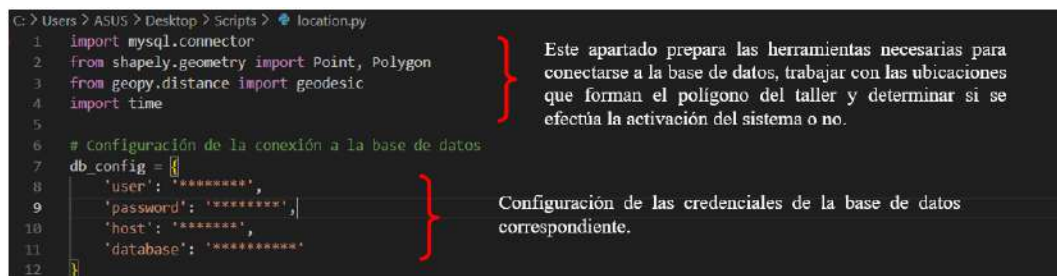
```

Estas líneas de código permiten imprimir los estados de las entradas y salidas de la tarjeta, mostrándolos en la pantalla OLED, antes configurada.

Figura 26. Impresión de los estados en la pantalla OLED.

- **Script de localización y georreferencia**

El siguiente código, pensado para controlar la activación del sistema a través de la aplicación web, implementa una lógica que permite identificar si las coordenadas del punto de activación de la emergencia se encuentran dentro del área delimitada del taller. La aplicación web está configurada para operar solo dentro de esta zona geográfica, asegurando que la activación de la alarma sea posible si las coordenadas están dentro de los límites establecidos. En caso de que las coordenadas no estén dentro del área del taller, el sistema no procesará la solicitud, garantizando así un funcionamiento controlado y seguro. En las siguientes figuras se detallan las secciones más relevantes del Anexo B, el cual contiene el código completo.



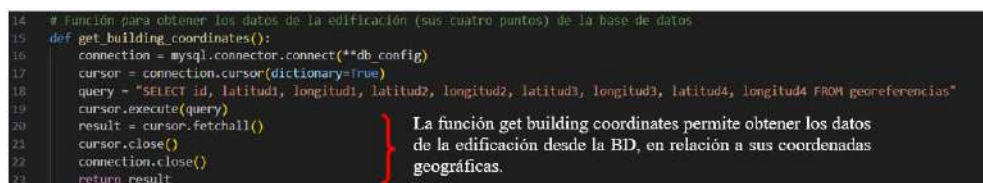
```
C:\Users\ASUS\Desktop\Scripts> location.py
1 import mysql.connector
2 from shapely.geometry import Point, Polygon
3 from geopy.distance import geodesic
4 import time
5
6 # Configuración de la conexión a la base de datos
7 db_config = {
8     'user': '*****',
9     'password': '*****',
10    'host': '*****',
11    'database': '*****'
12 }
```

Este apartado prepara las herramientas necesarias para conectarse a la base de datos, trabajar con las ubicaciones que forman el polígono del taller y determinar si se efectúa la activación del sistema o no.

Configuración de las credenciales de la base de datos correspondiente.

Figura 27. Configuraciones necesarias para conectarse con la base de datos.

Para visualizar de manera precisa la ubicación, se llevó a cabo el mapeo detallado del Taller de Innovación y Desarrollo Tecnológico FISEI, utilizando la normativa ISO 19107 “Información geográfica – Modelo Espacial” como referencia. Esta normativa establece que los polígonos deben estar contenidos dentro de áreas controladas y estructurados de forma correcta, asegurando que no existan errores de delimitación, solapamientos o inconsistencias en las áreas geográficas [40]. Las distribución de los 4 puntos que conforma el área delimitada para el taller se pueden visualizar en el Anexo C. En la siguiente figura se detalla el código empleado para acceder a las coordenadas del polígono cargado en la base de datos.



```
14 # Función para obtener los datos de la edificación (sus cuatro puntos) de la base de datos
15 def get_building_coordinates():
16     connection = mysql.connector.connect(**db_config)
17     cursor = connection.cursor(dictionary=True)
18     query = "SELECT id, latitud1, longitud1, latitud2, longitud2, latitud3, longitud3, latitud4, longitud4 FROM georeferencias"
19     cursor.execute(query)
20     result = cursor.fetchall()
21     cursor.close()
22     connection.close()
23     return result
```

La función get building coordinates permite obtener los datos de la edificación desde la BD, en relación a sus coordenadas geográficas.

Figura 28. Obtención de las coordenadas desde la BD.

En torno a lo mencionado anteriormente, la Figura 29 muestra el código para proceder con la identificación del edificio determinando si se efectúa la activación del sistema o no, teniendo en cuenta que el polígono está conformado de 4 puntos de longitud y latitud respectivamente para el taller, los mismos que se pueden apreciar en el Anexo D.

```

25 # Función para determinar el punto de activación (latitud y longitud)
26 def find_closest_building_id(user_latitude, user_longitude):
27     buildings = get_building_coordinates()
28     user_point = Point(user_longitude, user_latitude)
29
30     closest_building_id = None
31     min_distance = float('inf') # inicialmente una distancia infinita
32
33     for building in buildings:
34         # Definir el polígono de la edificación con sus cuatro puntos
35         building_polygon = Polygon([
36             (building['longitud1'], building['latitud1']),
37             (building['longitud2'], building['latitud2']),
38             (building['longitud3'], building['latitud3']),
39             (building['longitud4'], building['latitud4'])
40         ])
41
42         # Encontrar las coordenadas del usuario
43         closest_point_in_polygon = building_polygon.exterior.interpolate(building_polygon.exterior.project(user_point))
44
45         # Calcular la distancia geodésica entre el punto del usuario y el punto más cercano del polígono
46         distance = geodesic((user_latitude, user_longitude), (closest_point_in_polygon.y, closest_point_in_polygon.x)).meters
47
48         # verificar si las coordenadas están dentro del taller
49         if distance < min_distance:
50             min_distance = distance
51             closest_building_id = building['id']
52
53     return closest_building_id

```

La función find closest building, permite interactuar con las coordenadas que conforman el taller, teniendo en cuenta los 4 puntos que lo conforman, tanto en longitud como en latitud.

La función geodesic permite controlar las coordenadas del usuario y en función de ello activar o no el sistema.

Figura 29. Control de las coordenadas de activación.

- **Script para los sensores**

Al igual que en casos anteriores, el Anexo E contiene el código completo. El primer paso del siguiente script, es la identificación de las credenciales que se deben validar para permitir la comunicación entre el sistema y el servidor en línea. La Figura 30, muestra el apartado en las líneas de código referente a lo mencionado.

```

1  import time
2  import mysql.connector
3  import paho.mqtt.client as mqtt
4
5  # Configuración de la base de datos
6  db_config = {
7      'host': '*****',
8      'user': '*****',
9      'password': '*****',
10     'database': '*****'
11 }
12
13 # Configuración del servidor MQTT
14 mqtt_server = "74.***.***.***"
15 mqtt_user = "D**"
16 mqtt_password = "*****"

```

Herramientas necesarias para poder conectarse con la base de datos y el servidor MQTT

Credenciales de acceso para la base de datos

Credenciales de acceso el servidor en línea

Figura 30. Credenciales de acceso para a comunicación de los sensores.

```

93 def on_message(client, userdata, msg):
94     topic = msg.topic
95     payload = msg.payload.decode('utf-8')
96
97     # Mapear tópicos a IDs de sensores
98     topic_map = {
99         "huachi/fisei/talleres/PB_BP": 1,
100         "huachi/fisei/talleres/P1_BP": 2,
101         "huachi/fisei/talleres/P1_AI": 3
102     }
103
104     # Verificar si el tópicos es relevante
105     if topic in topic_map:
106         id_sensor = topic_map[topic]
107         conn = mysql.connector.connect(**db_config)
108         try:
109             if payload == "1":
110                 actualizar_estado(id_sensor, 1, conn)
111             elif payload == "0":
112                 actualizar_estado(id_sensor, 0, conn)
113         finally:
114             conn.close()

```

El apartado on message permite ejecutar una función en respuesta a un evento, para este caso se extrae el tópicos en formato binario en el cual se publicó el mensaje a través de utf-8

Representan canales o categorías específicas en los que los dispositivos o servicios pueden enviar mensajes.

Este código escucha los mensajes MQTT en los tópicos definidos. Si el mensaje proviene de un tópicos relevante extrae el identificador del sensor y actualiza su estado en la BD. Se marca "1" como activo y "0" como inactivo.

Figura 31. Actualización de los tópicos como activo o inactivo.

Posteriormente a la actualización de los tópicos, se genera una alarma en la base de datos que contiene información relevante sobre la emergencia; usuario, tipo de alarma, mensajes, latitud y longitud. Las líneas de código asociadas a esta partes se presentan a continuación en la Figura 32.

```

58 # Generar una nueva alarma en la base de datos
59 def generar_alarma(id_sensor, conn):
60     try:
61         cursor = conn.cursor()
62
63         # Obtener información del sensor
64         query_sensor = """
65         SELECT id_usuario, observacion, latitud, longitud, estado
66         FROM sensores
67         WHERE id = %s
68         """
69         cursor.execute(query_sensor, (id_sensor,))
70         sensor_data = cursor.fetchone()
71
72         if sensor_data:
73             id_usuario, observacion, latitud, longitud, estado = sensor_data
74             id_tipo = 8 # Tipo fijo
75
76             # Insertar nueva alarma
77             query_alarma = """
78             INSERT INTO alarmas (id_usuario, id_tipo, mensaje, latitud, longitud, estado)
79             VALUES (%s, %s, %s, %s, %s, %s)
80             """
81             cursor.execute(query_alarma, (id_usuario, id_tipo, observacion, latitud, longitud, estado))
82             conn.commit()
83             print(f"Alarma generada para id_sensor={id_sensor}")
84         else:
85             print(f"No se encontró información para el sensor con id={id_sensor}")
86
87     except mysql.connector.Error as err:
88         print(f"Error al generar alarma: {err}")
89     finally:
90         cursor.close()

```

Figura 32. Generar nuevas alarmas en la base de datos.

- *Script para las sirenas*

El objetivo de este código fue gestionar y evaluar los estados de las sirenas, de igual forma se procede a describir las secciones más relevantes del script, teniendo en cuenta que el mismo se encuentra en su totalidad en el Anexo F. Las figuras a continuación muestran una breve de explicación.

```

1  import time
2  import mysql.connector
3  import paho.mqtt.client as mqtt
4
5  # Configuración de la base de datos
6  db_config = {
7      'host': '*****',
8      'user': '*****',
9      'password': '*****',
10     'database': '*****'
11 }
12
13 # Configuración del servidor MQTT
14 mqtt_server = "74.***.***.***"
15 mqtt_user = "D**"
16 mqtt_password = "*****"

```

Herramientas necesarias para poder conectarse con la base de datos y el servidor MQTT

Credenciales de acceso para la base de datos

Credenciales de acceso al servidor en línea

Figura 33. Credenciales de acceso para a comunicación de las sirenas.

```

18 # Conectar al servidor MQTT
19 def conectar_mqtt():
20     client = mqtt.Client()
21     client.username_pw_set(mqtt_user, mqtt_password)
22     client.connect(mqtt_server, 1883, 60)
23     return client
24
25 # Enviar mensaje MQTT
26 def enviar_mensaje_mqtt(topico, mensaje):
27     try:
28         client = conectar_mqtt()
29         client.publish(topico, mensaje)
30         client.disconnect()
31     except Exception as e:
32         print(f"Error al enviar mensaje MQTT: {e}")
33

```

El apartado establece la conexión con el servidor MQTT utilizando los parámetros de configuración.

La función enviar mensaje mqtt publica mensajes en los tópicos correspondientes de MQTT. Se conecta al servidor MQTT, publica un mensaje y luego se desconecta. En caso de un error el código lo captura y lo muestra.

Figura 34. Publicación de mensajes en los tópicos.

Si se encuentran alarmas activas, se actualiza el estado de la sirena a "encendida" (estado = 1). Si no se encuentran alarmas activas, se evalúa si todas las alarmas están inactivas, y si es así, se apaga la sirena (estado = 0).

```

46     if alarmas_activas:
47         cursor.execute("UPDATE sirenas SET estado = 1 WHERE id = 1")
48         cursor.execute("UPDATE alarmas SET id_sirena = 1 WHERE id_tipo = 8 AND estado = 1")
49     else:
50         cursor.execute("""
51             SELECT COUNT(*) FROM alarmas WHERE id_tipo = 8 AND estado = 0
52             """)
53         inactivas_count = cursor.fetchone()[0]
54
55         cursor.execute("""
56             SELECT COUNT(*) FROM alarmas WHERE id_tipo = 8
57             """)
58         total_count = cursor.fetchone()[0]
59
60         if inactivas_count == total_count:
61             cursor.execute("UPDATE sirenas SET estado = 0 WHERE id = 1")
62

```

Figura 35. Actualización de los estados de la sirena.

Después de actualizar el estado de las sirenas, se consulta el estado de todas las sirenas registradas y se envía un mensaje NUTT con su estado (1 o 0) a los tópicos correspondientes.

```

64
65     # Evaluar y enviar mensajes MQTT para todas las sirenas
66     cursor.execute("SELECT id, topico, estado FROM sirenas")
67     sirenas = cursor.fetchall()
68
69     for sirena in sirenas:
70         id_sirena, topico, estado = sirena
71         enviar_mensaje_mqtt(topico, str(estado))

```

Figura 36. Envío de los estados a los tópicos.

b. Servidor lógico

El código detallado a continuación permitió la configuración de un servidor lógico diseñado para administrar de manera eficiente las interacciones entre los usuarios, la base de datos MySQL, y el sistema de almacenamiento de imágenes en la nube con Cloudinary, además de implementar notificaciones en tiempo real mediante Socket.IO. Las figuras a continuación presentan las secciones más relevantes del Anexo G, el mismo que contiene el código completo.

```

13  const app = express();
14  const server = http.createServer(app);
15  const io = socketIo(server, {
16    cors: {
17      origin: '*',
18      methods: ['GET', 'POST']
19    }
20  });

```

Herramientas primordiales para la funcionalidad del servidor

Apartado para habilitar comunicación en tiempo real entre clientes y el servidor.

Figura 37. Herramientas para habilitar el servidor.

La Figura 38 se enfoca en aspectos clave relacionados con la protección de la aplicación, así como el nivel de flexibilidad que el servidor ofrece para interactuar con otros dominios.

```

25  app.use(helmet({
26    contentSecurityPolicy: {
27      directives: {
28        defaultSrc: ["'self'"],
29        connectSrc: ["'self'", ...allowedOrigins.map(origin => origin.replace(/^https?/, 'w'))],
30      },
31    },
32  }));
33
34  app.use(cors({
35    origin: '*',
36    methods: ['GET', 'POST', 'PUT', 'OPTIONS'],
37    allowedHeaders: ['Content-Type', 'Authorization']
38  }));

```

Helmet es un paquete que ayuda a mejorar la seguridad de aplicaciones Express configurando cabeceras HTTP.

Este apartado permite que el servidor acepte solicitudes desde otros dominios.

Figura 38. Seguridad de la aplicación y solicitudes de otros dominios.

```

50  const db = mysql.createConnection({
51    host: process.env.DB_HOST,
52    user: process.env.DB_USER,
53    password: process.env.DB_PASSWORD,
54    database: process.env.DB_DATABASE
55  });

```

La conexión con MySQL utiliza credenciales de variables de entorno asegurando que no se expongan en el código

Figura 39. Credenciales de la conexión MySQL.

Las líneas de código presentadas en la Figura 39 permitieron mantener las credenciales utilizadas para la conexión con MySQL protegidas, evitando su exposición directa en el código. Este enfoque refuerza las prácticas de seguridad al almacenar las credenciales en un archivo de configuración.

```

83  app.get('/tipo-alerta', (req, res) => {
84    const query = 'SELECT id, descripcion FROM tipo_alerta';
85    db.query(query, (err, results) => {
86      if (err) return res.status(500).end();
87      res.json(results);
88    });
89  });

```

Devuelve una lista de tipos de alerta desde la tabla tipo_alerta.

Figura 40. Lista de alertas generadas en base a la tabla mencionada.

La Figura 41 señala el registro de una alerta en la tabla alarmas con datos mínimos como usuario e información de ubicación.

```
91 app.post('/trigger-panic-button', (req, res) => {
92   const { id_usuario, latitud, longitud } = req.body;
93   if (!id_usuario) return res.status(400).end();
94
95   const id_tipo = 8;
96   const estado = 1;
97
98   const insertQuery = `
99     INSERT INTO alarmas (id_usuario, id_tipo, mensaje, latitud, longitud, foto_url, id_georeferencia, id_sirena, estado, feedback)
100     VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
101   `;
102   const values = [id_usuario, id_tipo, null, latitud, longitud, null, null, null, estado, null];
```

Este apartado permite recopilar información clave sobre el usuario, así como la ubicación desde el lugar donde se activó o generó la alerta. Para realizar pruebas se tomó como referencia al usuario 8, con un estado activo de 1.

Figura 41. Registro de una alerta en la tabla de alarmas.

Finalmente, la generación de alertas permite incluir imágenes opcionales. Estas fotos se cargan previamente en la plataforma Cloudinary, y una vez procesadas, su URL se guarda en la base de datos para su posterior referencia. La Figura 42 presenta el código con el cual funciona dicha función.

```
110 app.post('/send-alert', upload.single('foto'), (req, res) => {
111   try {
112     const { id_usuario, id_tipo, mensaje, latitud, longitud } = req.body;
113     const foto = req.file ? req.file : null;
114
115     if (foto) {
116       cloudinary.uploader.upload_stream(
117         {
118           resource_type: 'image',
119           transformation: [
120             { width: 800, height: 600, crop: 'limit' },
121             { quality: 'auto' },
122             { fetch_format: 'auto' }
123           ]
124         },
125         (error, result) => {
126           if (error) return res.status(500).end();
```

Este apartado procesa alertas con imágenes opcionales. Si se incluye una foto, se sube a Cloudinary antes de guardarla en la base de datos.

Figura 42. Procesamiento de imágenes para guardarlas en la BD.

c. Aplicación “botón de pánico”

• Alarma simple

Para permitir la activación del sistema desde cualquier ubicación dentro del Taller de Innovación y Desarrollo Tecnológico FISEI, se diseñó una aplicación multiplataforma que funciona tanto en ordenadores como en teléfonos móviles con conexión a internet. Esta flexibilidad asegura que el sistema pueda ser utilizado de manera rápida y eficiente en cualquier momento. La Figura 43 muestra la interfaz inicial presentada al usuario, enfocada en la activación del botón de pánico.

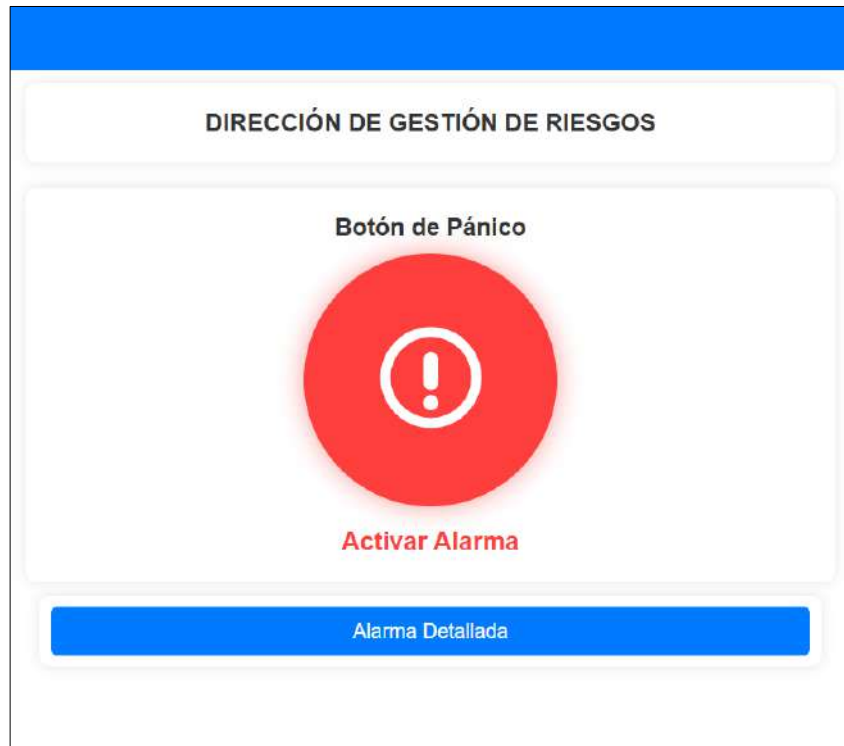


Figura 43. Aplicación botón de pánico.

La imagen muestra una interfaz intuitiva y funcional, diseñada específicamente para la gestión del sistema de alarmas. Al pulsar el botón de pánico, se genera una alarma que se envía a la base de datos, la cual activa automáticamente las notificaciones dirigidas a los miembros del taller a través de una mensaje de texto en Telegram. De manera simultánea, se activa una sirena física que emite una notificación sonora, garantizando así una alerta inmediata y una respuesta eficiente ante la situación generada.

- ***Alarma detallada***

En la versión mejorada de la interfaz, al presionar el botón azul con el texto "Alarma Detallada", el usuario es llevado a una nueva pantalla que conserva la estructura general de la interfaz inicial, pero con cambios visuales y funcionales que brindan mayor detalle. La nueva ventana se adapta al contexto de la alarma seleccionada, proporcionando información específica sobre el acontecimiento generado o presenciado.

The image shows a web interface for a risk management system. At the top is a blue header bar. Below it is a white box with the title 'DIRECCIÓN DE GESTIÓN DE RIESGOS'. Underneath is a section titled 'Alarma'. This section contains three main input areas: a dropdown menu labeled 'Tipo de Alerta' with the placeholder text 'Selecciona una alerta'; a large text area labeled 'Mensaje'; and a file upload section labeled 'Subir Foto (Opcional)' with a button 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'. Below these inputs are two prominent buttons: a blue 'Enviar Alerta' button and a red 'Boton de Pánico' button.

Figura 44. Alarma detallada.

Como se puede apreciar en la Figura 44, la nueva interfaz presenta una serie de apartados para mejorar la gestión y respuesta ante las alertas generadas. Estos apartados ofrecen una visión más detallada de la alarma, permitiendo al usuario enviar información adicional como; el tipo de alerta, mensaje de texto “plasmear ideas sobre el evento”, y fotos que permitan corroborar y validar la emergencia generada.

- ***Tipos de alarmas***

Dentro del sistema de alarmas comunitarias se consideró varias alertas que podrían ocurrir en el entorno educativo, las cuales se muestra en la Figura 45.

The image shows a web form titled 'Alarma'. It features a dropdown menu labeled 'Tipo de Alerta' with the placeholder text 'Selecciona una alerta'. The dropdown is open, showing a list of alert types: 'Personal no autorizado', 'Robo', 'Incendio', 'Emergencia Médica', 'Novedad', 'Falla Mecánica', and 'Vehículo no autorizado'. Below the dropdown is a blue button labeled 'Enviar Alerta'. At the bottom of the form is a large red button labeled 'Boton de Pánico'.

Figura 45. Listado de las posibles alerta a generarse.

- ***Mensajes adicionales***

Se adicionó este apartado, con el fin de que el usuario puede enviar información relevante o crucial sobre la alerta generada o visualizada, facilitando la toma de decisiones ante dicho eventos. La Figura 46 muestra este apartado en la aplicación.

The image shows the same 'Alarma' form but with additional fields. Below the 'Tipo de Alerta' dropdown is a text area labeled 'Mensaje' containing the text: 'Buenos días, necesito ayuda en el taller de la FISEI, una persona se acaba de caer de las gradas.' Below this is a section labeled 'Subir Foto (Opcional)' which includes a button 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'. The blue 'Enviar Alerta' button and the red 'Boton de Pánico' button are still present at the bottom.

Figura 46. Apartado para enviar información adicional.

- ***Apartado para fotos***

En mucha situaciones de emergencia, una foto o imagen plasma el evento de mejor manera. Con eso en mente se adicionó este apartado, donde es opcional que el usuario

envíe fotos desde su celular o en el caso que esté en el ordenador. La Figura 47 muestra un ejemplo de su uso.

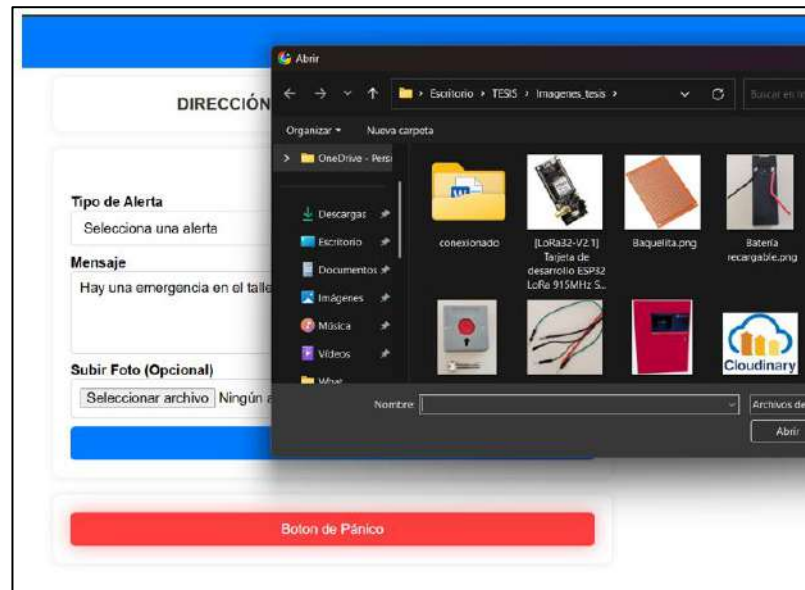


Figura 47. Apartado para enviar fotos o imágenes.

Finalmente, en la parte inferior de la interfaz, se encuentra un botón de color azul, diseñado específicamente para permitir el envío inmediato de la información recolectada en tiempo real. Este botón facilita la transmisión de datos esenciales relacionados con la alarma, ya sea a los responsables de seguridad, al equipo de mantenimiento o al sistema centralizado.

Además, para mejorar la experiencia del usuario, si el acceso a la alarma detallada se realizó por error, existe la opción de regresar fácilmente a la interfaz anterior. Esto se logra mediante un botón de color rojo denominado “Botón de pánico”.

d. Interfaz de monitoreo

Para supervisar de manera eficiente las alertas generadas por los usuarios, se desarrolló una interfaz intuitiva y funcional que recopila, organiza y presenta la información de forma detallada y estructurada. Esta herramienta permitió visualizar todos los eventos relevantes en tiempo real, la misma que se presenta en la Figura 48.

todos sus integrantes. El mensaje incluye información clara y detallada sobre la situación detectada, como el tipo de alerta, la ubicación específica, y otros datos relevantes, incluyendo imágenes o instrucciones específicas si es necesario. Este mecanismo asegura una difusión rápida y efectiva de la información, la Figura 50 muestra un ejemplo cuando el sistema esta activado.



Figura 50. Notificación de la situación generada.

3.6 Cuarta fase

3.6.1 Pruebas de desempeño y validación del sistema de alarmas.

a. Población y muestra

- Población

Para comprobar el funcionamiento del sistema aplicado, se definió como población de estudio a los docentes que forman parte del Taller de Innovación y Desarrollo Tecnológico FISEI. Además, se incluyó al personal que presta sus servicios en dicho

espacio y a los estudiantes que están desarrollando sus trabajos de titulación en el mismo lugar, garantizando así una evaluación integral y representativa en el contexto del taller. La Tabla 31 detalla la población de estudio.

Tabla 31. Población conformada por el taller.

Población	Número
Autoridades	5
Docentes	7
Estudiantes	27
Personal operativo	3
Total	42

Se tomó en cuenta la población mencionada previamente, considerando aspectos como su comprensión del tema de investigación, sus conocimientos previos y su nivel de familiarización con el sistema de control y monitorización de alarmas comunitarias.

- ***Muestra***

Dado que la población detallada en la Tabla 31 es inferior a 100 personas, se procederá a realizar el estudio con el total de la misma.

- b. Pruebas de funcionamiento***

Para las pruebas de funcionamiento inicial del sistema, se estableció como punto de partida la verificación integral de su correcta activación, asegurando que cada componente (botón, sirena, tarjeta de desarrollo) respondiera de manera adecuada.

De igual manera, se llevó a cabo una revisión del envío de notificaciones sobre los eventos registrados, garantizando que dichas alertas llegaran de manera efectiva y oportuna a todos los usuarios involucrados en las pruebas iniciales. Estas notificaciones se evaluaron tanto en términos de contenido, para asegurar que la información proporcionada fuera clara y precisa, como en términos de entrega, verificando su llegada a los usuarios.

El fin de estas pruebas fue confirmar el adecuado funcionamiento del sistema en condiciones controladas, asegurando su confiabilidad y capacidad para cumplir con los requerimientos establecidos, la cual se enfocó en los dispositivos y la aplicación

que activan el sistema, en la Tabla 32 se muestran las pruebas realizadas y los datos obtenidos.

Tabla 32. Pruebas de funcionamiento del sistema de alarmas.

Pruebas de activación del sistema de alarmas implementado						
#	Botón 1- Planta baja		Botón 2 – Piso uno		Aplicación - Botón de pánico	
	Fecha y Hora	Acción	Fecha y Hora	Acción	Fecha y Hora	Acción
1	12/12/2024 11:00	Falla	12/12/2024 11:05	Correcto	12/12/2024 11:10	Correcto
2	12/12/2024 11:15	Correcto	12/12/2024 11:20	Correcto	12/12/2024 11:25	Correcto
3	12/12/2024 11:30	Correcto	12/12/2024 11:35	Correcto	12/12/2024 11:40	Correcto
4	12/12/2024 11:45	Correcto	12/12/2024 11:50	Correcto	12/12/2024 11:55	Correcto
5	12/12/2024 12:00	Correcto	12/12/2024 12:05	Correcto	12/12/2024 12:10	Correcto
6	12/12/2024 12:15	Correcto	12/12/2024 12:20	Correcto	12/12/2024 12:25	Correcto
7	12/12/2024 12:30	Correcto	12/12/2024 12:35	Falla	12/12/2024 12:40	Correcto
8	12/12/2024 12:45	Correcto	12/12/2024 12:50	Correcto	12/12/2024 12:55	Correcto
9	12/12/2024 13:00	Correcto	12/12/2024 13:05	Correcto	12/12/2024 13:10	Correcto
10	12/12/2024 13:15	Correcto	12/12/2024 13:20	Correcto	12/12/2024 13:25	Correcto
11	13/12/2024 09:00	Correcto	04/12/2024 09:05	Correcto	13/12/2024 09:10	Correcto
12	13/12/2024 09:15	Correcto	13/12/2024 09:20	Correcto	13/12/2024 09:25	Correcto
13	13/12/2024 09:30	Falla	13/12/2024 09:35	Correcto	13/12/2024 09:40	Falla
14	13/12/2024 09:45	Correcto	13/12/2024 09:50	Correcto	13/12/2024 09:55	Correcto
15	13/12/2024 10:00	Correcto	13/12/2024 10:05	Correcto	13/12/2024 10:10	Correcto
16	13/12/2024 10:15	Correcto	13/12/2024 10:20	Correcto	13/12/2024 10:25	Correcto
17	13/12/2024 10:30	Correcto	13/12/2024 10:35	Correcto	13/12/2024 10:40	Correcto
18	13/12/2024 10:45	Correcto	13/12/2024 10:50	Falla	13/12/2024 10:55	Correcto
19	13/12/2024 11:00	Correcto	13/12/2024 11:05	Correcto	13/12/2024 11:10	Correcto
20	13/12/2024 11:15	Correcto	13/12/2024 11:20	Correcto	13/12/2024 11:25	Falla
21	16/12/2024 10:00	Falla	16/12/2024 10:05	Correcto	16/12/2024 10:10	Correcto
22	16/12/2024 10:15	Correcto	16/12/2024 10:20	Correcto	16/12/2024 10:25	Correcto
23	16/12/2024 10:30	Correcto	16/12/2024 10:35	Correcto	16/12/2024 10:40	Correcto
24	16/12/2024 10:45	Correcto	16/12/2024 10:50	Falla	16/12/2024 10:55	Falla
25	16/12/2024 11:00	Correcto	16/12/2024 11:05	Correcto	16/12/2024 11:10	Correcto
26	16/12/2024 11:15	Correcto	16/12/2024 11:20	Correcto	16/12/2024 11:25	Correcto
27	16/12/2024 11:30	Correcto	16/12/2024 11:35	Correcto	16/12/2024 11:40	Correcto
28	16/12/2024 11:45	Correcto	16/12/2024 11:50	Correcto	16/12/2024 11:55	Correcto
29	16/12/2024 12:00	Correcto	16/12/2024 12:05	Correcto	16/12/2024 12:10	Correcto
30	16/12/2024 12:15	Correcto	16/12/2024 12:20	Correcto	16/12/2024 12:25	Correcto
31	17/12/2024 14:00	Correcto	17/12/2024 14:05	Correcto	17/12/2024 14:10	Correcto
32	17/12/2024 14:15	Correcto	17/12/2024 14:20	Correcto	17/12/2024 14:25	Correcto
33	17/12/2024 14:30	Falla	17/12/2024 14:35	Correcto	17/12/2024 14:40	Correcto
34	17/12/2024 14:45	Correcto	17/12/2024 14:50	Correcto	17/12/2024 14:55	Falla
35	17/12/2024 15:00	Correcto	17/12/2024 15:05	Correcto	17/12/2024 15:10	Correcto
36	17/12/2024 15:15	Correcto	17/12/2024 15:20	Correcto	17/12/2024 15:25	Correcto
37	17/12/2024 15:30	Correcto	17/12/2024 15:35	Correcto	17/12/2024 15:40	Correcto

Pruebas de activación del sistema de alarmas implementado						
#	Botón 1- Planta baja		Botón 2 – Piso uno		Aplicación - Botón de pánico	
	Fecha y Hora	Acción	Fecha y Hora	Acción	Fecha y Hora	Acción
38	17/12/2024 15:45	Correcto	17/12/2024 15:50	Falla	17/12/2024 15:55	Correcto
39	17/12/2024 16:00	Correcto	17/12/2024 16:05	Correcto	17/12/2024 16:10	Correcto
40	17/12/2024 16:15	Correcto	17/12/2024 16:20	Correcto	17/12/2024 16:25	Correcto
41	18/12/2024 16:30	Correcto	18/12/2024 16:35	Correcto	18/12/2024 16:40	Correcto
42	18/12/2024 16:45	Correcto	18/12/2024 16:50	Correcto	18/12/2024 16:55	Correcto
43	18/12/2024 17:00	Correcto	18/12/2024 17:05	Correcto	18/12/2024 17:10	Correcto
44	18/12/2024 17:15	Correcto	18/12/2024 17:20	Correcto	18/12/2024 17:25	Correcto
45	18/12/2024 17:30	Correcto	18/12/2024 17:35	Correcto	18/12/2024 17:40	Correcto

En la Tabla 32 se presentan los resultados obtenidos durante las pruebas de activación del sistema, realizadas tanto a través de los botones físicos como mediante la aplicación del botón de pánico. Sin embargo, en lo que respecta a la activación mediante la señal del módulo contra incendios y detector de humo, solo se llevaron a cabo dos pruebas. Esto se debió a que, tras cada activación, el sistema requería realizar múltiples configuraciones para restablecer las condiciones normales de funcionamiento, lo que dificultó la realización de pruebas adicionales. Por este motivo, se decidió no incluir este método en el conjunto de pruebas evaluadas.

- **Error Absoluto**

El error absoluto (ε_a) se define como la diferencia positiva entre el valor real ($\bar{x}$) y un valor medido o estimado (x_i). Matemáticamente, se expresa mediante la siguiente fórmula:

$$\varepsilon_a = |\bar{x} - x_i| \quad (1)$$

Donde:

ε_a = Error absoluto

$\bar{x}$ = Valor real

x_i = Medición

- **Error Relativo**

El error relativo (ϵ_r) se calcula dividiendo el error absoluto (ϵ_a) entre el valor real ($\bar{x}$). Este resultado puede representarse en forma porcentual o como un valor comprendido entre 0 y 1. Su expresión matemática es:

$$\epsilon_r = \frac{|\bar{x} - x_i|}{\bar{x}} * 100 \% \quad (2)$$

Donde:

ϵ_r = Error relativo

$\bar{x}$ = Valor real

x_i = Medición

A partir de los datos recopilados en la Tabla 32, correspondientes a las pruebas realizadas, se aplicaron las fórmulas (1) y (2) para calcular el error absoluto y el error relativo. Estos cálculos fueron fundamentales para la evaluación del desempeño del sistema de alarmas implementado en el Taller de Innovación y Desarrollo Tecnológico FISEI.

Tabla 33. Cálculo del error absoluto y relativo de las pruebas.

Cálculo del error de las pruebas realizadas			
Error calculado	Dispositivo de activación		
	Botón 1 – Planta baja	Botón 2 – Piso uno	Aplicación – Botón de pánico
Error absoluto ϵ_a	$\epsilon_a = \bar{x} - x_i $ $\epsilon_a = 45 - 41 $ $\epsilon_a = 4$	$\epsilon_a = \bar{x} - x_i $ $\epsilon_a = 45 - 40 $ $\epsilon_a = 5$	$\epsilon_a = \bar{x} - x_i $ $\epsilon_a = 45 - 41 $ $\epsilon_a = 4$
Error relativo ϵ_r	$\epsilon_r = \frac{ \bar{x} - x_i }{\bar{x}} * 100 \%$ $\epsilon_r = \frac{ 45 - 41 }{45} * 100 \%$ $\epsilon_r = \frac{4}{45} * 100 \%$	$\epsilon_r = \frac{ \bar{x} - x_i }{\bar{x}} * 100 \%$ $\epsilon_r = \frac{ 45 - 40 }{45} * 100 \%$ $\epsilon_r = \frac{5}{45} * 100 \%$	$\epsilon_r = \frac{ \bar{x} - x_i }{\bar{x}} * 100 \%$ $\epsilon_r = \frac{ 45 - 41 }{45} * 100 \%$ $\epsilon_r = \frac{4}{45} * 100 \%$

Cálculo del error de las pruebas realizadas			
Error calculado	Dispositivo de activación		
	Botón 1 – Planta baja	Botón 2 – Piso uno	Aplicación – Botón de pánico
	$\epsilon_r = 8,889 \% \approx 9\%$	$\epsilon_r = 11,11 \% \approx 11\%$	$\epsilon_r = 8,889 \% \approx 9\%$

La Tabla 33 sirvió como base para evaluar el desempeño del sistema mediante un total de 45 mediciones realizadas, las cuales permitieron calcular los errores correspondientes a cada dispositivo de activación. En el caso del Botón 1 - Planta baja, se obtuvo un error absoluto de 4, lo que corresponde a un error relativo de aproximadamente 9%, indicando una eficiencia del 91% en su funcionamiento. Para el Botón 2 - Piso uno, el error absoluto fue de 5, con un error relativo de 11%, lo que refleja una eficiencia del 89%. Por último, para la Aplicación - Botón de pánico, el error absoluto fue también de 4, con un error relativo de 9 % alcanzando una eficiencia del 91%.

Estos resultados reflejan que, en promedio, el sistema de activación tiene un error relativo aproximado del 9%, lo que implica una eficiencia global del 91% en su desempeño. La Figura 51 muestra un representación gráfica de los resultados obtenidos.

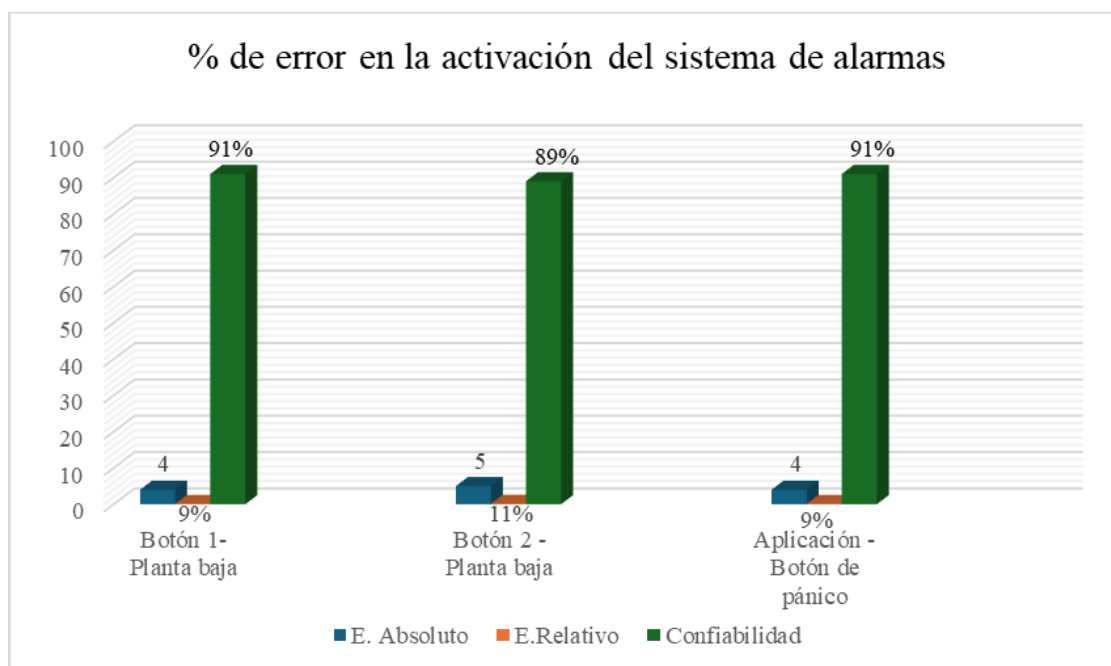


Figura 51. Porcentaje de error en la activación del sistema.

- **Tiempo de respuesta del sistema**

Teniendo en cuenta los datos presentados en la Tabla 32, a continuación se detalla el tiempo que el sistema tarda en responder a una activación generada, mediante los botones físicos o a través de la aplicación. Estos tiempos, expresados en segundos, representan el intervalo entre el evento que activa el sistema y la respuesta efectiva del mismo.

Tabla 34. Tiempo de retraso en la activación del sistema.

Pruebas de activación del sistema de alarmas implementado						
#	Botón 1- Planta baja		Botón 2- Planta uno		Aplicación - Botón de pánico	
	Tiempo en sg	Acción	Tiempo en sg	Acción	Tiempo en sg	Acción
1	00:00:00,00	Falla	00:00:06,86	Correcto	00:00:10,24	Correcto
2	00:00:06,12	Correcto	00:00:06,21	Correcto	00:00:10,70	Correcto
3	00:00:05,59	Correcto	00:00:06,73	Correcto	00:00:09,37	Correcto
4	00:00:05,77	Correcto	00:00:05,99	Correcto	00:00:10,45	Correcto
5	00:00:05,84	Correcto	00:00:06,52	Correcto	00:00:10,82	Correcto
6	00:00:06,09	Correcto	00:00:06,80	Correcto	00:00:09,54	Correcto
7	00:00:06,52	Correcto	00:00:00,00	Falla	00:00:09,08	Correcto
8	00:00:05,07	Correcto	00:00:06,68	Correcto	00:00:12,64	Correcto
9	00:00:06,82	Correcto	00:00:06,65	Correcto	00:00:12,61	Correcto
10	00:00:06,40	Correcto	00:00:06,01	Correcto	00:00:12,10	Correcto
11	00:00:05,65	Correcto	00:00:06,99	Correcto	00:00:09,24	Correcto
12	00:00:05,57	Correcto	00:00:06,20	Correcto	00:00:12,57	Correcto
13	00:00:00,00	Falla	00:00:05,11	Correcto	00:00:00,00	Falla
14	00:00:05,78	Correcto	00:00:05,14	Correcto	00:00:09,35	Correcto
15	00:00:06,74	Correcto	00:00:06,71	Correcto	00:00:09,28	Correcto
16	00:00:06,03	Correcto	00:00:05,08	Correcto	00:00:10,74	Correcto
17	00:00:05,81	Correcto	00:00:05,89	Correcto	00:00:12,56	Correcto
18	00:00:05,52	Correcto	00:00:00,00	Falla	00:00:09,12	Correcto
19	00:00:06,78	Correcto	00:00:06,81	Correcto	00:00:11,07	Correcto
20	00:00:05,24	Correcto	00:00:05,89	Correcto	00:00:00,00	Falla
21	00:00:00,00	Falla	00:00:06,55	Correcto	00:00:09,22	Correcto
22	00:00:06,80	Correcto	00:00:06,77	Correcto	00:00:10,72	Correcto
23	00:00:05,38	Correcto	00:00:05,46	Correcto	00:00:11,86	Correcto
24	00:00:06,98	Correcto	00:00:00,00	Falla	00:00:00,00	Falla
25	00:00:05,01	Correcto	00:00:06,48	Correcto	00:00:11,25	Correcto
26	00:00:05,95	Correcto	00:00:05,12	Correcto	00:00:11,59	Correcto
27	00:00:06,05	Correcto	00:00:05,49	Correcto	00:00:11,45	Correcto
28	00:00:06,98	Correcto	00:00:05,24	Correcto	00:00:11,47	Correcto
29	00:00:06,72	Correcto	00:00:05,78	Correcto	00:00:12,77	Correcto
30	00:00:06,95	Correcto	00:00:05,94	Correcto	00:00:12,28	Correcto
31	00:00:06,77	Correcto	00:00:05,22	Correcto	00:00:11,20	Correcto
32	00:00:06,22	Correcto	00:00:06,16	Correcto	00:00:09,45	Correcto
33	00:00:00,00	Falla	00:00:06,72	Correcto	00:00:12,74	Correcto

Pruebas de activación del sistema de alarmas implementado						
#	Botón 1- Planta baja		Botón 2- Planta uno		Aplicación - Botón de pánico	
	Tiempo en sg	Acción	Tiempo en sg	Acción	Tiempo en sg	Acción
34	00:00:05,25	Correcto	00:00:05,88	Correcto	00:00:00,00	Falla
35	00:00:05,38	Correcto	00:00:06,22	Correcto	00:00:10,00	Correcto
36	00:00:06,68	Correcto	00:00:06,54	Correcto	00:00:11,27	Correcto
37	00:00:06,64	Correcto	00:00:06,28	Correcto	00:00:09,36	Correcto
38	00:00:06,00	Correcto	00:00:00,00	Falla	00:00:10,64	Correcto
39	00:00:06,98	Correcto	00:00:06,00	Correcto	00:00:09,02	Correcto
40	00:00:05,20	Correcto	00:00:05,14	Correcto	00:00:11,53	Correcto
41	00:00:05,23	Correcto	00:00:05,48	Correcto	00:00:11,48	Correcto
42	00:00:05,08	Correcto	00:00:06,77	Correcto	00:00:10,18	Correcto
43	00:00:05,30	Correcto	00:00:06,18	Correcto	00:00:11,74	Correcto
44	00:00:05,17	Correcto	00:00:06,64	Correcto	00:00:09,88	Correcto
45	00:00:05,44	Correcto	00:00:06,76	Correcto	00:00:09,73	Correcto

En la Tabla 34 se muestran los resultados obtenidos del tiempo de retraso asociado a cada activación del sistema. Este tiempo de retraso corresponde al intervalo transcurrido entre el momento en que se genera una activación y el instante en que el sistema responde de manera efectiva.

El sistema propuesto se ajusta a los términos establecidos en la normativa UNE–EN 50131–1 , “*Requisitos Generales*”, la cual establece que el tiempo máximo de respuesta de un sistema de alarma no debe exceder los 10 segundos, considerando la distancia a la cual se genera la activación. Además, en situaciones de emergencia, la normativa recomienda que los tiempos de activación sean aún más rápidos, con un máximo de 2 a 4 segundos para garantizar una respuesta inmediata y eficaz. En cuanto a la gestión de los cables, sugiere el uso de cables blindados FPLR y FPLP (Fire Power Limited Plenum) recubiertos de poliéster de aluminio, los cuales deben ser instalados dentro de un conducto de acero o material similar que resista condiciones ambientales extremas, con especial atención a la protección contra incendios. Respecto a las carcasas y botones, estas deben ser de materiales resistentes que protejan los componentes internos de manipulaciones y condiciones adversas. Los botones de emergencia deben ser fácilmente accesibles, contruidos con materiales duraderos “acero, aluminio, entre otros” y capaces de resistir el desgaste, garantizando su fiabilidad y funcionamiento seguro en situaciones críticas [41].

- **Media aritmética o promedio**

El promedio ($\bar{X}$) se calcula sumando todos los valores de los datos ($x_1 + x_2 + x_3 + \dots + x_n$) y dividiéndolos por el total de valores (n). Su expresión matemática es:

$$\bar{X} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \quad (3)$$

Donde:

$\bar{X}$ = Media aritmética o promedio

$x_1 + x_2 + x_3 + \dots + x_n$ = Es la sumatoria de los valores o mediciones efectuadas.

n = Es la cantidad total de valores.

A partir de los datos recopilados en la Tabla 34, que corresponden a los tiempos de retraso registrados en cada activación del sistema, se procedió a realizar un análisis estadístico para obtener un valor representativo del comportamiento promedio del sistema. Para ello, se aplicó la fórmula (3), correspondiente al cálculo de la media aritmética, la cual permite determinar el tiempo promedio de retraso considerando todos los valores involucrados. Este cálculo es fundamental para determinar en donde se encuentra el cuello de botella que tiene el sistema.

Tabla 35. Cálculo del tiempo promedio en el retraso del sistema.

Cálculo del tiempo de retraso promedio en las pruebas realizadas			
Medida estadística	Dispositivo de activación		
	Botón 1 – Planta baja	Botón 2 – Piso uno	Aplicación – Botón de pánico
Promedio $\bar{X}$	$\bar{X} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$	$\bar{X} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$	$\bar{X} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$
	$\bar{X} = \frac{245,5 \text{ sg}}{45}$	$\bar{X} = \frac{251,10 \text{ sg}}{45}$	$\bar{X} = \frac{442,31 \text{ sg}}{45}$
	$\bar{X} = 5,46 \text{ sg}$	$\bar{X} = 5,58 \text{ sg}$	$\bar{X} = 9,83 \text{ sg}$

Con base en los datos obtenidos en la Tabla 34 y los cálculos realizados en la Tabla 35, se puede concluir que el sistema presenta un tiempo promedio de retraso de 5,46 segundos para la activación del botón 1 (Planta Baja). En cuanto al botón 2 (Piso Uno), su tiempo promedio de retraso es de 5,58 segundos, mientras que la aplicación móvil muestra un retraso de 9,83 segundos en su activación. Estos resultados permitieron identificar el cuello de botella en el sistema, el cual se atribuye a las características del servidor virtual.

UPC	RAM	NVMe
VPS Linux XS	1 núcleo virtual	1 GB
		10 GB

Figura 52. Características del servidor virtual.

La Figura 52 presenta las características técnicas del servidor virtual requeridas para su funcionamiento en línea. Estas características son las que generan los retrasos en la activación del sistema, ya que su rendimiento y capacidad de procesamiento influyen directamente en los tiempos de respuesta durante las interacciones con los botones y la aplicación móvil.

- ***Servidor VPS Linux XS empleado***

Este servidor, con 1 núcleo virtual, 1 GB de RAM y 10 GB SSD en Ubuntu 24.04, es adecuado para aplicaciones ligeras como servidores web, APIs pequeñas, broker MQTT para IoT, o bases de datos de bajo consumo. Su capacidad permite automatizar tareas, ejecutar scripts en Python o Bash, y servir como entorno de pruebas para el desarrollo de software. De igual forma, su uso en la nube ofrece flexibilidad y estabilidad para proyectos que no requieren alta demanda de procesamiento o almacenamiento, optimizando costos y recursos en soluciones escalables. Si bien, la capacidad del servidor no es adecuada para el sistema propuesto, puede considerarse como un punto de inicio para la selección de otras alternativas que se ajusten a las necesidades del sistema.

c. Validación del sistema

Para garantizar una validación efectiva del sistema, se identificaron y analizaron diferentes factores. Los mismos permitieron establecer una comparación equitativa, asegurando que las características y el desempeño de los dispositivos sean relacionados bajo las mismas condiciones. El enfoque utilizado se alineó con los métodos previamente manejados para comparar los componentes. La Tabla 36 presenta los factores que se consideraron en la validación.

Tabla 36. Descripción de los factores para la validación del sistema.

Factor	Denominación	Descripción
F1	Tiempo de respuesta	Indica el tiempo desde la activación de una alarma hasta que el usuario recibe la notificación.
F2	Precisión	Evalúa si las alarmas generadas corresponden a eventos reales y minimizan falsos positivos.
F3	Confiabilidad	Define la capacidad del sistema de operar sin interrupciones o fallas durante un período de tiempo.
F4	Escalabilidad	Señala el rendimiento del sistema al aumentar el número de dispositivos conectados o usuarios activos.
F5	Facilidad de Uso	Recolecta retroalimentación de los usuarios sobre la interfaz y operación del sistema.

Al analizar cada factor para la validación del sistema, se elaboró una matriz de enfrentamiento que vincula los criterios en función de su importancia relativa, evaluándolos con una ponderación de 1 a 5 como se presenta en la Tabla 37.

Tabla 37. Escala de relevancia para la evaluación de los criterios de validación.

Descripción	Valor
Importancia máxima	5
Muy alta importancia	4
Alta importancia	3
Importancia moderada	2
Igual importancia	1

Para la ponderación del sistema se tomó como referencia las características de un kit de alarma comunitaria con perifoneo celular vox 500 linseg. La Tabla 38 muestra las principales funcionalidades de los dos sistemas.

Tabla 38. Características principales de los sistemas de alarmas.

Características	Sistema implementado	Kit de alarma comunitaria
Representación		
Dispositivos: Botones de activación Sirena de notificación Señalética Panel Batería de respaldo Aplicación web	2 botones físicos 1 sirena de 100W 2 señaléticas Dos paneles para las placas Batería Lipo de 7.4V 5.2 A Si dispone	No dispone de botones 1 sirena de 30W, 1 sirena de 20W No dispone Carcasa integrada con las placas Batería TC 12 V 4.5 A Si dispone
Material de la caja	TPU (poliuretano termoplástico)	Plástico ABS - IP65
Tarjeta o controlador empleado.	Tarjeta LILYGO LoRa32 V1.6.1	MICRO CHIP “no especifica” con seguridad de transferencia de datos
Configuración de tipo de usuario	Escalable y ajustable	5 grupos (Admin0, Admin1, Admin2, policía y publico)
Confiabilidad	91% al tratarse de un piloto	100% al tratarse de un producto para el mercado

Tabla 39. Ponderación de los factores para la validación del sistema.

Ponderación de factores							
	F1	F2	F3	F4	F5	Total	Peso ponderado
F1		1	1	1	5	8	0,20
F2	1		3	1	1	6	0,15
F3	4	1		1	1	7	0,17
F4	4	4	1		1	10	0,24

Ponderación de factores							
F5	1	4	4	1		10	0,24
Total						41	1

Utilizando los pesos asignados en la Tabla 37, se evaluaron los factores en la escala de la Tabla 13, que se aplica a los sistemas seleccionados. En este caso, se comparó las características principales del piloto implementado contra un sistema disponible en el mercado, tal como se muestra en la siguiente tabla.

Tabla 40. Comparación de los sistemas de alarmas.

Factor	Peso Ponderado	Sistema implementado		Kit de alarma comunitaria	
		Calificación	Total	Calificación	Total
Tiempo de respuesta	0,2	6	1,2	10	2
Precisión	0,15	6	0,9	10	1,5
Confiabilidad	0,17	6	1,02	10	1,7
Escalabilidad	0,24	8	1,92	2	0,48
Facilidad de Uso	0,24	7	1,68	10	2,4
Total			6,72		8,08

La validación del sistema implementado se llevó a cabo mediante una comparación cuantitativa con un kit de alarma comunitaria disponible en el mercado, evaluando diversos factores clave. En términos de "Tiempo de respuesta", el sistema implementado mostró un rendimiento inferior debido a las características del servidor utilizado, lo que sugiere que el kit de alarma comunitaria tiene una mayor rapidez de respuesta. Sin embargo, este aspecto puede mejorarse optimizando la infraestructura del servidor. En cuanto a "Precisión", ambos sistemas ofrecen un buen desempeño, aunque el kit de alarma comunitaria tiene una ventaja en cuanto a la relevancia y control en la detección, dada su orientación al mercado. En "Confiabilidad", el sistema

implementado mostró un buen desempeño debido a las pruebas realizadas y el tiempo de funcionamiento, pero el kit de alarma comunitaria, respaldado por especificaciones del fabricante, superó al sistema implementado, al estar probado en diversos entornos operativos. En relación a "Escalabilidad", el sistema implementado obtuvo una calificación superior debido a su capacidad para integrar dispositivos de diferentes proveedores, lo que le otorga una mayor flexibilidad para su expansión. Por otro lado, el kit de alarma comunitaria trabaja en un lazo cerrado, lo que limita su capacidad de integración con dispositivos de otros fabricantes. Finalmente, en "Facilidad de Uso", aunque el kit de alarma comunitaria destaca por su aplicación descargable de fácil acceso, el sistema implementado ofrece una página web que puede ser adaptada a las necesidades específicas del entorno, lo que también facilita su uso. En general, la validación del sistema implementado demuestra que, si bien el kit de alarma comunitaria tiene algunas ventajas en términos de rapidez y control, el sistema implementado presenta un rendimiento sólido, con una capacidad de adaptación superior y un alto potencial de escalabilidad.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Los sistemas de control y monitorización de alarmas comunitarias se han consolidado como herramientas fundamentales para fortalecer la seguridad ciudadana y promover la colaboración entre los residentes y las autoridades. Estos sistemas integran tecnologías avanzadas, como IoT, comunicación inalámbrica y plataformas digitales, que centralizan y optimizan la gestión de alertas. Su implementación no solo reduce significativamente los tiempos de respuesta ante emergencias, sino que también facilita la toma de decisiones en tiempo real. El análisis de los equipos empleados en los sistemas de control y monitoreo de alarmas comunitarias destaca la importancia de una selección adecuada para garantizar un funcionamiento eficiente y confiable. Elementos como botones de alerta, sirenas de notificación, módulos de comunicación inalámbrica y plataformas de gestión en la nube deben ser evaluados cuidadosamente en función de criterios como compatibilidad, capacidad de integración, alcance operativo y eficiencia energética. Asimismo, optar por tecnologías escalables asegura la adaptabilidad del sistema a futuros requerimientos, mejorando su capacidad de respuesta ante emergencias y su sostenibilidad a largo plazo.
- La implementación del sistema de control y monitorización de alarmas comunitarias para el Taller de Innovación y Desarrollo Tecnológico FISEI de la Universidad Técnica de Ambato ha demostrado un desempeño eficiente y confiable. Con base en las mediciones realizadas, los resultados indican un promedio de error relativo del 9%, lo que se traduce en una eficiencia global del 91%. Este rendimiento refleja la efectividad de los dispositivos de activación, botones físicos y la aplicación móvil, en términos de respuesta y precisión. Además, el análisis detallado de los errores individuales en los dispositivos de control permite identificar áreas de mejora, asegurando la continuidad en la optimización del sistema.

- La validación del sistema implementado, mediante su comparación con un kit disponible en el mercado, permitió destacar tanto las fortalezas como las áreas de mejora del sistema desarrollado. Los resultados demostraron que, si bien ambos sistemas presentan un desempeño similar en términos de eficiencia y respuesta, el sistema implementado ofrece ventajas, como una mayor escalabilidad y la integración de tecnologías personalizadas a las necesidades específicas de cada entorno. Esta comparación valida la funcionalidad del sistema y demuestra su competitividad frente a soluciones comerciales.
- Los tiempos de respuesta observados del sistema, con un retraso promedio de 5,46 segundos para el botón 1 (Planta Baja) y 5,58 segundos para el botón 2 (Piso Uno), están dentro de los parámetros establecidos por la normativa UNE–EN 50131–1. Sin embargo, se evidencia un retraso más significativo en la aplicación móvil, con un tiempo de 9,83 segundos, lo que sugiere un área de mejora. A pesar de que los tiempos de respuesta cumplen con los requisitos normativos, existen oportunidades para optimizar el rendimiento del sistema, especialmente en la aplicación móvil, con el fin de reducir estos retrasos y mejorar la eficiencia general del sistema.

4.2 Recomendaciones

- Investigar sobre alternativas a las tarjetas de desarrollo actuales con el objetivo de optimizar el sistema. Teniendo en cuenta aspectos claves como el consumo energético para garantizar una operación eficiente y sostenible, el alcance de comunicación para cubrir áreas más amplias, y la disponibilidad en el mercado asegurando la viabilidad de su adquisición en el mercado.
- Indagar y evaluar software que ofrezcan un control más seguro y eficiente de la información en el sistema de alarmas, de libre acceso y conexión en tiempo real.
- Para trabajos futuros es aconsejable fabricar el diseño de la placa base en centros o lugares especializados, con el fin de reducir errores humanos y mejorar la presentación, calidad y funcionalidad del mismo.

- Elaborar el conexionado eléctrico bajo la normativa legal vigente en Ecuador, asegurando el cumplimiento de los parámetros necesarios para la operación segura de los equipos y la reducción de riesgos que puedan ocasionar fallos en el sistema.
- Planificar pruebas de funcionamiento basadas en las necesidades específicas del proyecto, considerando diferentes escenarios operativos, variables del entorno y requisitos de desempeño.

REFERENCIAS BIBLIOGRÁFICAS

- [1] O. Velazquez, G. Pescaroli, G. Cremen, and C. Galasso, “A Review of the Technical and Socio-Organizational Components of Earthquake Early Warning Systems,” in *Frontiers in Earth Science*, 2020.
- [2] K. Hugelius, J. Becker, and A. Adolfsson, “Five Challenges When Managing Mass Casualty or Disaster Situations: A Review Study,” *International Journal of Environmental Research and Public Health*, vol. 17, 2020.
- [3] S. N. Herrera, S. V. Coba, and A. Rodríguez-Guerra, “Cultura organizacional para manejo y prevención de riesgos laborales en una institución educativa, Quito-Ecuador,” *Revista Ñeque*, vol. 4, no. 10, pp. 157–169, 2021.
- [4] C. Rueda-Panchano, “Implementación de un sistema de alarma contra intrusiones en la oficina de profesores de la FACI – UTLVTE,” *Ibero-American Journal of Engineering & Technology Studies*, vol. 3, pp. 492–498, 2023.
- [5] L. Haz, I. Carrera, M. F. Molina, and G. V. S. Bernal, “Prototype of smart community alarm for monitoring events and incidents related to citizen safety,” in *2020 14th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, pp. 1–4.
- [6] K. M. Cuadrado Sevillano, “Prototipo de sistema integrado de seguridad comunitaria con tecnología plug and play a bajo costo,” Universidad Regional Autónoma de los Andes (UNIANDES), Ibarra, Ecuador, 2020.
- [7] K. G. Chicaiza Guachi, “Sistema de alarma comunitaria para el mercado San Juan de la ciudad Santiago de Píllaro,” Universidad Técnica de Ambato (UTA), Píllaro, Ecuador, 2020.
- [8] C.-L. Chen, Z.-Y. Lim, and H.-C. Liao, “Blockchain-Based Community Safety Security System with IoT Secure Devices,” *Sustainability*, 2021.
- [9] E. G. Yanchatipán Moposita, “Sistema de seguridad integral para la monitorización de alarmas y alerta de emergencia en la parroquia San Andrés del cantón Píllaro de la provincia de Tungurahua,” Universidad Técnica de Ambato (UTA), Píllaro, Ecuador, 2022.
- [10] M. Babiuch and J. Postulka, “Smart Home Monitoring System Using ESP32 Microcontrollers,” *Internet of Things*, 2020.
- [11] P. L. I. S. Larico, “Sistema de Seguridad electrónica basada en la norma ISO 27001,” *INF-FCPN-PGI Revista PGI*, pp. 156–159, 2021.
- [12] C. S. L. Díaz Vivanco Ángel Santiago, “Descubriendo las alarmas ciudadanas,” *Revista de Seguridad y Tecnología*, vol. 1, no. 1, pp. 1–5, 2023.

- [13] G. Gerber, “Alarm Systems in Buildings – Investigation of Effectiveness Depending on Building Type and User Profile,” *Communications*, vol. 22, pp. 140–147, 2020.
- [14] W. A. G. Rodríguez and J. S. U. Maldonado, *Modelo de integración tecnológica del subsistema de alarmas comunitarias con los sistemas de seguridad y emergencias (SIES)*. Bogotá, D.C.: Pontificia Universidad Javeriana, 2021.
- [15] J. Guerra, P. Acharya, and C. Barnadas, “Community-based surveillance: A scoping review,” *PLoS ONE*, vol. 14, 2020.
- [16] D. Nettikadan, S. Raj, and others, “Smart community monitoring system using Thingspeak IoT platform,” *International Journal of Applied Engineering Research*, vol. 13, no. 17, pp. 13402–13408, 2020.
- [17] X. Lin, M. Guizani, X. Du, C.-K. Chu, and Y. Yu, “Advances of Security and Privacy Techniques in Emerging Wireless Networks,” *IEEE Wireless Communications*, vol. 27, pp. 8–9, 2020.
- [18] M. M. Mansour, A. Gamal, A. I. Ahmed, L. A. Said, A. Elbaz, N. Herencsar, and A. Soltan, “Internet of Things: A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools, and Future Directions,” *Energies*, 2023.
- [19] D. L. Mai and M.-K. Kim, “Multi-Hop LoRa Network Protocol with Minimized Latency,” *Energies*, 2020.
- [20] S. Iglesias-Osores, “Norma editorial: ¿es adecuado no citar los artículos de más de cinco años de antigüedad?,” *FEM: Revista de la Fundación Educación Médica*, vol. 23, pp. 293–293, 2020.
- [21] A. Sherif, S. Sherif, C.-P. Ooi, and W. H. Tan, “A LoRa-Driven Home Security System for a Residential Community in a Retirement Township,” *International Journal of Technology*, 2020.
- [22] S. S. Veit and F. Steiner, “Proposed Changes to the Inspection Strategy for Fire Alarm Systems: Empirical Analysis of Weak Points and Technical Influencing Factors,” *Safety*, 2024.
- [23] W. P. Gamboa and C. S. Erazo, “Análisis del sistema de alarmas del EMS del CENACE,” *Revista Técnica "energía"*, vol. 10, no. 1, pp. 191–199, 2020.
- [24] H. G. Torres and S. T. Díaz, “Las tecnologías de bajo costo y su contribución en la mejora de la seguridad ciudadana: alarmas comunitarias.” Zenodo, Dec-2021.
- [25] R. Fedele and M. Merenda, “An IoT System for Social Distancing and Emergency Management in Smart Cities Using Multi-Sensor Data,” *Algorithms*, vol. 13, p. 254, 2020.

- [26] P. Netinant, T. Utsanok, M. Rukhiran, and S. Klongdee, "Development and Assessment of Internet of Things-Driven Smart Home Security and Automation with Voice Commands," *IoT*, 2024.
- [27] A. Rehman, M. A. Qureshi, T. Ali, M. Irfan, S. Abdullah, S. Yasin, U. Draz, A. Głowacz, G. Nowakowski, A. Alghamdi, A. A. Alsulami, and M. Wegrzyn, "Smart Fire Detection and Deterrent System for Human Savior by Using Internet of Things (IoT)," *Energies*, 2021.
- [28] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, "Internet of Things and Its Applications: A Comprehensive Survey," *Symmetry*, vol. 12, p. 1674, 2020.
- [29] J. A. Afonso, V. Monteiro, and J. L. Afonso, "Internet of Things Systems and Applications for Smart Buildings," *Energies*, 2023.
- [30] A. A. Elngar and M. Kayed, "Vehicle Security Systems using Face Recognition based on Internet of Things," *Open Computer Science*, vol. 10, pp. 17–29, 2020.
- [31] R. Maskeliūnas, R. Damaševičius, and S. Segal, "A Review of Internet of Things Technologies for Ambient Assisted Living Environments," *Future Internet*, vol. 11, p. 259, 2020.
- [32] W. C. Ahiara and C. Ihekweaba, "Implementation of an Internet of Things (IoT) Based Neighborhood Distress Alert System," *ABUAD Journal of Engineering Research and Development*, 2023.
- [33] P. V. Kumar, M. N. Kadnar, M. A. Gosavi, and M. O. Jadhav, "IOT-Based Flood Monitoring and Alarm System," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 5, pp. 5142–5146, May 2023.
- [34] I. M. M. A. Hasani, S. I. A. Kazmi, R. A. Shah, R. Hasan, and S. Hussain, "IoT based Fire Alerting Smart System," *Sir Syed University Research Journal of Engineering & Technology*, vol. 12, no. 2, pp. 46–50, 2022.
- [35] R. Singh, H. Al-Khateeb, G. Ahmadi-Assalemi, and G. Epiphaniou, "Towards an IoT Community-Cluster Model for Burglar Intrusion Detection and Real-Time Reporting in Smart Homes," in *Challenges in the IoT and Smart Environments*, Springer International Publishing, 2021, pp. 53–73.
- [36] I. A. C. E. Industries, "Generic Standard on Printed Board Design." IPC Association Connecting Electronics Industries, Bannockburn, IL, USA, 2023.
- [37] IPC, *Sectional Design Standard for Rigid Organic Printed Boards*, Revision B. Association Connecting Electronics Industries, 2020.
- [38] I. – A. C. E. Industries, *Acceptability of Electronic Assemblies*, J. Bannockburn, IL, USA: IPC, 2024.

- [39] *Reglamento Interno de Seguridad y Salud de los Trabajadores y Mejoramiento del Medio Ambiente del Trabajo*. Quito, Ecuador: Secretaría de Derechos Humanos, 2021.
- [40] *Información geográfica – Modelo Espacial*, no. ISO 19107:2021. Geneva, Switzerland: Organización Internacional de Normalización, 2021.
- [41] AENOR, “UNE-EN 50131-1:2020: Requisitos generales de los sistemas de alarma.” 2020.

ANEXOS

Anexo A. Código completo para la detección de las alarmas generadas.

```
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3  #include <Wire.h>
4  #include <Adafruit_GFX.h>
5  #include <Adafruit_SSD1306.h>
6
7  // Configuración WiFi
8  const char* ssidList[] = {"alarmaTT", "FISEI-WiFi", "Lizbeth"};
9  const char* passwordList[] = {"12345678", "", "VeritoCaiza2001"};
10
11 // Configuración del servidor MQTT
12 const char* mqtt_server = "74.***.***.***";
13 const char* mqtt_user = "****";
14 const char* mqtt_password = "*****";
15
16 // Tópicos MQTT
17 const char* topic_P1_S1 = "huachi/fisei/talleres/P1_S1";
18 const char* topic_P1_S2 = "huachi/fisei/talleres/P1_S2";
19 const char* topic_PB_BP = "huachi/fisei/talleres/PB_BP";
20 const char* topic_P1_BP = "huachi/fisei/talleres/P1_BP";
21 const char* topic_P1_AI = "huachi/fisei/talleres/P1_AI";
22
23 // Pines
24 #define GPIO12 12 // Nueva salida
25 #define GPIO13 13 // Nueva salida
26 #define GPIO34 34 // Entrada
27 #define GPIO35 35 // Entrada
28 #define GPIO26 26 // Entrada
29
30 // Estados previos de los pines de entrada
31 int lastState_GPIO34 = LOW;
32 int lastState_GPIO35 = LOW;
33 int lastState_GPIO26 = LOW;
34
35 // Configuración OLED
36 #define SCREEN_WIDTH 128
37 #define SCREEN_HEIGHT 64
38 #define OLED_RESET -1
39 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
40
41 WiFiClient espClient;
42 PubSubClient client(espClient);
43
44 void setup() {
45     // Inicialización Serial
46     Serial.begin(115200);
47
48     // Configurar pines
49     pinMode(GPIO12, OUTPUT); // Salida
50     pinMode(GPIO13, OUTPUT); // Salida
51     pinMode(GPIO34, INPUT); // Entrada
52     pinMode(GPIO35, INPUT); // Entrada
53     pinMode(GPIO26, INPUT); // Entrada
54
55     // Inicializar pantalla OLED
56     if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
57         Serial.println(F("No se pudo inicializar el OLED"));
58         while (true);
59     }
```

```

61 // Configuración WiFi
62 setupWiFi();
63
64 // Configuración del cliente MQTT
65 client.setServer(mqtt_server, 1883);
66 client.setCallback(callback);
67
68 // Mostrar texto fijo en el OLED
69 display.clearDisplay();
70 display.setTextSize(1);
71 display.setTextColor(SSD1306_WHITE);
72 display.setCursor(0, 0);
73 display.println("HUACHI/FISEI/TALLERES");
74 display.display();
75 }
76
77 void setupWiFi() {
78     delay(10);
79     Serial.println("Conectando a WiFi...");
80
81     for (int i = 0; i < sizeof(ssidList) / sizeof(ssidList[0]); i++) {
82         Serial.print("Intentando conectar a: ");
83         Serial.println(ssidList[i]);
84         WiFi.begin(ssidList[i], passwordList[i]);
85
86         int attempts = 0;
87         while (WiFi.status() != WL_CONNECTED && attempts < 10) {
88             delay(1000);
89             Serial.print(".");
90             attempts++;
91         }
92
93         if (WiFi.status() == WL_CONNECTED) {
94             Serial.println("\nWiFi conectado");
95             Serial.print("Dirección IP: ");
96             Serial.println(WiFi.localIP());
97             return;
98         }
99     }
100
101     Serial.println("No se pudo conectar a ninguna red WiFi");
102     while (true); // Detener el programa si no hay conexión
103 }
104
105 void reconnect() {
106     while (!client.connected()) {
107         Serial.print("Conectando al servidor MQTT...");
108         if (client.connect("ESP32Client", mqtt_user, mqtt_password)) {
109             Serial.println("Conectado");
110             // Suscribirse a los tópicos
111             client.subscribe(topic P1 S1);

```

Anexo B. Código completo para la localización y georreferencia.

```
G:\Users\ASUS\Desktop> Scripts > location.py
1 import mysql.connector
2 from shapely.geometry import Point, Polygon
3 from geopy.distance import geodesic
4 import time
5
6 # Configuración de la conexión a la base de datos
7 db_config = {
8     'user': '*****',
9     'password': '*****',
10    'host': '*****',
11    'database': '*****'
12 }
13
14 # Función para obtener los datos de la edificación (sus cuatro puntos) de la base de datos
15 def get_building_coordinates():
16     connection = mysql.connector.connect(**db_config)
17     cursor = connection.cursor(dictionary=True)
18     query = "SELECT id, latitud1, longitud1, latitud2, longitud2, latitud3, longitud3, latitud4, longitud4 FROM georeferencias"
19     cursor.execute(query)
20     result = cursor.fetchall()
21     cursor.close()
22     connection.close()
23     return result
24
25 # Función para determinar el punto de activación (latitud y longitud)
26 def find_closest_building_id(user_latitude, user_longitude):
27     buildings = get_building_coordinates()
28     user_point = Point(user_longitude, user_latitude)
29
30     closest_building_id = None
31     min_distance = float('inf') # Inicialmente una distancia infinita
32
33     for building in buildings:
34         # Definir el polígono de la edificación con sus cuatro puntos
35         building_polygon = Polygon([
36             (building['longitud1'], building['latitud1']),
37             (building['longitud2'], building['latitud2']),
38             (building['longitud3'], building['latitud3']),
39             (building['longitud4'], building['latitud4'])
40         ])
41
42         # Encontrar las coordenadas del usuario
43         closest_point_in_polygon = building_polygon.exterior.interpolate(building_polygon.exterior.project(user_point))
44
45         # Calcular la distancia geodésica entre el punto del usuario y el punto más cercano del polígono
46         distance = geodesic((user_latitude, user_longitude), (closest_point_in_polygon.y, closest_point_in_polygon.x)).meters
47
48         # Verificar si las coordenadas están dentro del taller
49         if distance < min_distance:
50             min_distance = distance
51             closest_building_id = building['id']
52
53     return closest_building_id
54
55 # Función para obtener el registro más reciente de la tabla alarmas
56 def get_latest_alarm(last_id):
57     connection = mysql.connector.connect(**db_config)
58     cursor = connection.cursor(dictionary=True)
59     query = f"SELECT id, latitud, longitud FROM alarmas WHERE id > {last_id} ORDER BY id ASC LIMIT 1"
60     cursor.execute(query)
61     result = cursor.fetchone()
62     cursor.close()
63     connection.close()
64     return result
65
66 # Función para actualizar el id_georeferencias en la tabla alarmas
67 def update_alarm_georeference(alarm_id, georeference_id):
68     connection = mysql.connector.connect(**db_config)
69     cursor = connection.cursor()
70     query = "UPDATE alarmas SET id_georeferencia = %s WHERE id = %s"
71     cursor.execute(query, (georeference_id, alarm_id))
72     connection.commit()
73     cursor.close()
74     connection.close()
75
76 # Inicializar el seguimiento del último ID procesado
77 last_processed_id = 0
78
79 # Bucle para verificar continuamente nuevos registros en la tabla alarmas
80 while True:
81     new_alarm = get_latest_alarm(last_processed_id)
82     if new_alarm:
83         last_processed_id = new_alarm['id']
84         user_latitude = new_alarm['latitud']
85         user_longitude = new_alarm['longitud']
86
87         # Encontrar el id de la edificación más cercana
88         closest_building_id = find_closest_building_id(user_latitude, user_longitude)
89
90         # Actualizar el registro en la tabla alarmas
91         if closest_building_id is not None:
92             update_alarm_georeference(last_processed_id, closest_building_id)
93     ~
```

Anexo C. Geolocalización del Taller de Innovación y Desarrollo Tecnológico.



Anexo D. Latitud y longitud de los 4 puntos del polígono del taller.

Numeración	Descripción
1	Taller de Innovación y Desarrollo Tecnológico FISEI

Punto 1	Latitud	Longitud
	-1,266888	-78,62381

Punto 2	Latitud	Longitud
	-1,266852	-78,623699

Punto 3	Latitud	Longitud
	-1,267256	-78,623584

Punto 4	Latitud	Longitud
	-1,267295	-78,623699

Anexo E. Código completo para los sensores.

```
1 import time
2 import mysql.connector
3 import paho.mqtt.client as mqtt
4
5 # Configuración de la base de datos
6 db_config = {
7     'host': '*****',
8     'user': '*****',
9     'password': '*****',
10    'database': '*****'
11 }
12
13 # Configuración del servidor MQTT
14 mqtt_server = "74.***.***.***"
15 mqtt_user = "D**"
16 mqtt_password = "*****"
17
18 # Caché para estados de sensores
19 sensor_states = {}
20
21 # Conectar al servidor MQTT
22 def conectar_mqtt():
23     client = mqtt.Client()
24     client.username_pw_set(mqtt_user, mqtt_password)
25     client.connect(mqtt_server, 1883, 60)
26     return client
27
28 # Actualizar el estado en la base de datos
29 def actualizar_estado(id_sensor, estado, conn):
30     global sensor_states
31
32     # Verificar si el estado ya está registrado
33     if sensor_states.get(id_sensor) == estado:
34         print(f"Estado de id_sensor={id_sensor} ya actualizado a estado={estado}. Ignorando mensaje.")
35         return
36
37     try:
38         cursor = conn.cursor()
39
40         # Actualizar estado del sensor
41         query = "UPDATE sensores SET estado = %s WHERE id = %s"
42         cursor.execute(query, (estado, id_sensor))
43         conn.commit()
44         print(f"Estado actualizado para id_sensor={id_sensor}, estado={estado}")
45
46         # Actualizar caché de estados
47         sensor_states[id_sensor] = estado
48
49         # Si el estado es 1, insertar una nueva alarma
50         if estado == 1:
51             generar_alarma(id_sensor, conn)
52
53     except mysql.connector.Error as err:
54         print(f"Error al conectar o actualizar la base de datos: {err}")
55     finally:
56         cursor.close()
```

```

59 def generar_alarma(id_sensor, conn):
70     sensor_data = cursor.fetchone()
71
72     if sensor_data:
73         id_usuario, observacion, latitud, longitud, estado = sensor_data
74         id_tipo = 8 # Tipo fijo
75
76         # Insertar nueva alarma
77         query_alarma = """
78         INSERT INTO alarmas (id_usuario, id_tipo, mensaje, latitud, longitud, estado)
79         VALUES (%s, %s, %s, %s, %s, %s)
80         """
81         cursor.execute(query_alarma, (id_usuario, id_tipo, observacion, latitud, longitud, estado))
82         conn.commit()
83         print(f"Alarma generada para id_sensor={id_sensor}")
84     else:
85         print(f"No se encontró información para el sensor con id={id_sensor}")
86
87 except mysql.connector.Error as err:
88     print(f"Error al generar alarma: {err}")
89 finally:
90     cursor.close()
91
92 # Callback para manejar mensajes entrantes
93 def on_message(client, userdata, msg):
94     topic = msg.topic
95     payload = msg.payload.decode('utf-8')
96
97     # Mapear tópicos a IDs de sensores
98     topic_map = {
99         "huachi/fisei/talleres/PB_BP": 1,
100         "huachi/fisei/talleres/P1_BP": 2,
101         "huachi/fisei/talleres/P1_AI": 3
102     }
103
104     # Verificar si el tópico es relevante
105     if topic in topic_map:
106         id_sensor = topic_map[topic]
107         conn = mysql.connector.connect(**db_config)
108         try:
109             if payload == "1":
110                 actualizar_estado(id_sensor, 1, conn)
111             elif payload == "0":
112                 actualizar_estado(id_sensor, 0, conn)
113         finally:
114             conn.close()
115
116 # Configurar cliente MQTT y suscribir a los tópicos
117 client = conectar_mqtt()
118 client.on_message = on_message
119
120 # Suscribirse a los tópicos relevantes
121 client.subscribe("huachi/fisei/talleres/PB_BP")
122 client.subscribe("huachi/fisei/talleres/P1_BP")
123 client.subscribe("huachi/fisei/talleres/P1_AI")
124
125 # Iniciar bucle de escucha
126 print("Conectado al servidor MQTT. Esperando mensajes...")
127 client.loop_forever()

```

Anexo F. Código completo para la activación de la sirena.

```
1 import time
2 import mysql.connector
3 import paho.mqtt.client as mqtt
4
5 # Configuración de la base de datos
6 db_config = {
7     'host': '*****',
8     'user': '*****',
9     'password': '*****',
10    'database': '*****'
11 }
12
13 # Configuración del servidor MQTT
14 mqtt_server = "74.***.***.***"
15 mqtt_user = "D***"
16 mqtt_password = "*****"
17
18 # Conectar al servidor MQTT
19 def conectar_mqtt():
20     client = mqtt.Client()
21     client.username_pw_set(mqtt_user, mqtt_password)
22     client.connect(mqtt_server, 1883, 60)
23     return client
24
25 # Enviar mensaje MQTT
26 def enviar_mensaje_mqtt(topico, mensaje):
27     try:
28         client = conectar_mqtt()
29         client.publish(topico, mensaje)
30         client.disconnect()
31     except Exception as e:
32         print(f"Error al enviar mensaje MQTT: {e}")
33
34 # Actualizar estado de sirenas y enviar mensajes MQTT
35 def actualizar_y_evaluar_sirenas():
36     conn = mysql.connector.connect(**db_config)
37     cursor = conn.cursor()
38
39     try:
40         # Actualizar sirenas y alarmas según las condiciones
41         cursor.execute("""
42             SELECT id FROM alarmas WHERE id_tipo = 8 AND estado = 1
43             """)
44         alarmas_activas = cursor.fetchall()
45
46         if alarmas_activas:
47             cursor.execute("UPDATE sirenas SET estado = 1 WHERE id = 1")
48             cursor.execute("UPDATE alarmas SET id_sirena = 1 WHERE id_tipo = 8 AND estado = 1")
49
50     except:
51         cursor.execute("""
52             SELECT COUNT(*) FROM alarmas WHERE id_tipo = 8 AND estado = 0
53             """)
54         inactivas_count = cursor.fetchone()[0]
55
56         cursor.execute("""
57             SELECT COUNT(*) FROM alarmas WHERE id_tipo = 8
58             """)
59         total_count = cursor.fetchone()[0]
60
61         if inactivas_count == total_count:
62             cursor.execute("UPDATE sirenas SET estado = 0 WHERE id = 1")
63
64     conn.commit()
65
66     # Evaluar y enviar mensajes MQTT para todas las sirenas
67     cursor.execute("SELECT id, topico, estado FROM sirenas")
68     sirenas = cursor.fetchall()
69
70     for sirena in sirenas:
71         id_sirena, topico, estado = sirena
72         enviar_mensaje_mqtt(topico, str(estado))
73
74     except mysql.connector.Error as err:
75         print(f"Error al interactuar con la base de datos: {err}")
76     finally:
77         cursor.close()
78         conn.close()
79
80 # Ejecutar la función cada 2 segundos
81 while True:
82     actualizar_y_evaluar_sirenas()
83     time.sleep(2)
```

Anexo G. Código para el servidor lógico.

```
1  const express = require('express');
2  const bodyParser = require('body-parser');
3  const cors = require('cors');
4  const mysql = require('mysql2');
5  const multer = require('multer');
6  const cloudinary = require('cloudinary').v2;
7  const http = require('http');
8  const socketIo = require('socket.io');
9  const helmet = require('helmet');
10 const path = require('path');
11 require('dotenv').config();
12
13 const app = express();
14 const server = http.createServer(app);
15 const io = socketIo(server, {
16   cors: {
17     origin: '*',
18     methods: ['GET', 'POST']
19   }
20 });
21 const port = 3004;
22
23 const allowedOrigins = process.env.ALLOWED_ORIGINS.split(',');
24
25 app.use(helmet({
26   contentSecurityPolicy: {
27     directives: {
28       defaultSrc: ["'self'"],
29       connectSrc: ["'self'", ...allowedOrigins.map(origin => origin.replace(/^https?/, 'ws'))],
30     },
31   },
32 }));
33
34 app.use(cors({
35   origin: '*',
36   methods: ['GET', 'POST', 'PUT', 'OPTIONS'],
37   allowedHeaders: ['Content-Type', 'Authorization']
38 }));
39
40 app.options('*', cors());
41 app.use(bodyParser.json());
42 app.use(bodyParser.urlencoded({ extended: true }));
43
44 cloudinary.config({
45   cloud_name: process.env.CLOUDINARY_CLOUD_NAME,
46   api_key: process.env.CLOUDINARY_API_KEY,
47   api_secret: process.env.CLOUDINARY_API_SECRET
48 });
```

```

50  const db = mysql.createConnection({
51    host: process.env.DB_HOST,
52    user: process.env.DB_USER,
53    password: process.env.DB_PASSWORD,
54    database: process.env.DB_DATABASE
55  });
56
57  db.connect(err => {
58    if (err) return;
59  });
60
61  const storage = multer.memoryStorage();
62  const upload = multer({
63    storage,
64    limits: { fileSize: 10 * 1024 * 1024 }
65  });
66
67  app.get('/random-user', (req, res) => {
68    const query = 'SELECT id, username, mail FROM usuarios ORDER BY RAND() LIMIT 1';
69    db.query(query, (err, results) => {
70      if (err) return res.status(500).end();
71      if (results.length > 0) {
72        res.json({
73          id: results[0].id,
74          username: results[0].username,
75          mail: results[0].mail
76        });
77      } else {
78        res.status(404).end();
79      }
80    });
81  });
82
83  app.get('/tipo-alerta', (req, res) => {
84    const query = 'SELECT id, descripcion FROM tipo_alerta';
85    db.query(query, (err, results) => {
86      if (err) return res.status(500).end();
87      res.json(results);
88    });
89  });
90
91  app.post('/trigger-panic-button', (req, res) => {
92    const { id_usuario, latitud, longitud } = req.body;
93    if (!id_usuario) return res.status(400).end();
94
95    const id_tipo = 8;
96    const estado = 1;
97
98    const insertQuery = `
99      INSERT INTO alarmas (id_usuario, id_tipo, mensaje, latitud, longitud, foto_url, id_georeferenci
100      VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)

```

```

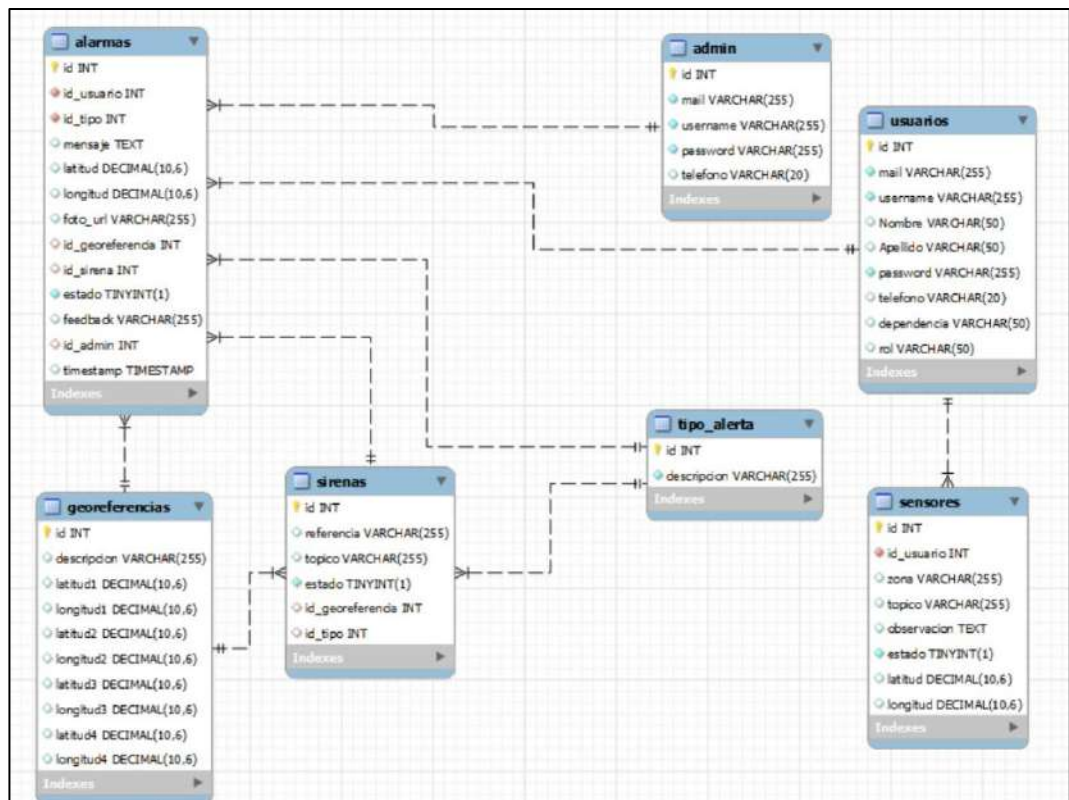
101     `;
102     const values = [id_usuario, id_tipo, null, latitud, longitud, null, null, null, estado, null];
103
104     db.query(insertQuery, values, (insertErr, insertResults) => {
105         if (insertErr) return res.status(500).end();
106         res.json({ id: insertResults.insertId });
107     });
108 });
109
110 app.post('/send-alert', upload.single('foto'), (req, res) => {
111     try {
112         const { id_usuario, id_tipo, mensaje, latitud, longitud } = req.body;
113         const foto = req.file ? req.file : null;
114
115         if (foto) {
116             cloudinary.uploader.upload_stream(
117                 {
118                     resource_type: 'image',
119                     transformation: [
120                         { width: 800, height: 600, crop: 'limit' },
121                         { quality: 'auto' },
122                         { fetch_format: 'auto' }
123                     ]
124                 },
125                 (error, result) => {
126                     if (error) return res.status(500).end();
127
128                     const foto_url = result.secure_url;
129                     const sql = `INSERT INTO alarmas (id_usuario, id_tipo, mensaje, latitud, longitud, foto_url)
130                         VALUES (?, ?, ?, ?, ?, ?)`;
131                     db.query(sql, [id_usuario, id_tipo, mensaje, latitud, longitud, foto_url], (err) => {
132                         if (err) return res.status(500).end();
133                         res.json({ foto_url });
134                     });
135                 }
136             ).end(foto.buffer);
137         } else {
138             const sql = `INSERT INTO alarmas (id_usuario, id_tipo, mensaje, latitud, longitud, foto_url)
139                 VALUES (?, ?, ?, ?, ?, ?)`;
140             db.query(sql, [id_usuario, id_tipo, mensaje, latitud, longitud, null], (err) => {
141                 if (err) return res.status(500).end();
142                 res.json({ foto_url: null });
143             });
144         }
145     } catch {
146         res.status(500).end();
147     }
148 });
149
150 server.listen(port);

```

Anexo H. Código para la generación de mensajes en Telegram.

```
1 import asyncio
2 import requests
3 from telegram import Bot
4
5 # Configuración
6 BOT_TOKEN = "7716751683:AAGTAerXu-wCBKjCSHe-5jI-MOYzyqj2H80"
7 CHAT_ID = -4663928525 # Reemplaza con tu chat_id
8 API_URL = "http://localhost:3004/ultima-alarma"
9
10 # Variable para almacenar el último ID procesado
11 ultimo_id_alarma = None
12
13 # Función para consultar la API
14 async def consultar_api():
15     global ultimo_id_alarma
16     try:
17         response = requests.get(API_URL)
18         if response.status_code == 200:
19             data = response.json()
20             id_alarma = data.get("id_alarma")
21             referencia = data.get("descripcion_georeferencia")
22
23             if id_alarma != ultimo_id_alarma and referencia: # Verifica si hay un nuevo registro y que 'referencia' no sea nulo o vacío
24                 ultimo_id_alarma = id_alarma
25                 await enviar_mensaje(formatar_mensaje(data))
26         else:
27             print(f"Error en la API: {response.status_code}")
28     except Exception as e:
29         print(f"Error al consultar la API: {e}")
30
31 # Función para formatear el mensaje
32 def formatar_mensaje(data):
33     maps_url = f"https://www.google.com/maps?om={data.get('latitud', '0')},{data.get('longitud', '0')}"
34     foto_url = data.get("foto_url", "Sin foto disponible")
35     foto_display = f"[Ver foto]({foto_url})" if data.get("foto_url") else "Sin foto disponible"
36     mensaje = (
37         f"🚨 **Alerta Recibida**\n"
38         f"ID Alarma: {data.get('id_alarma', 'Desconocido')}\n"
39         f"Tipo: {data.get('descripcion_tipo_alarma', 'No especificado')}\n"
40         f"Usuario: {data.get('nombre_usuario', 'N/A')} {data.get('apellido_usuario', 'N/A')}\n"
41         f"Rol: {data.get('rol_usuario', 'N/A')}\n"
42         f"Dependencia: {data.get('dependencia_usuario', 'N/A')}\n"
43         fMensaje: {data.get('mensaje', 'Sin mensaje')}\n"
44         fReferencia: {data.get('descripcion_georeferencia', 'Sin referencia')}\n"
45         fFoto: {foto_display}\n"
46         fMaps: [Ubicación en Google Maps]({maps_url})\n"
47     )
48     return mensaje
49
50 # Función para enviar un mensaje al grupo
51 async def enviar_mensaje(texto):
52     bot = Bot(token=BOT_TOKEN)
53     try:
54         await bot.send_message(chat_id=CHAT_ID, text=texto, parse_mode="Markdown")
55         print("Mensaje enviado correctamente")
56     except Exception as e:
57         print(f"Error al enviar el mensaje: {e}")
58
59 # Función principal
60 async def main():
61     while True:
62         await consultar_api()
63         await asyncio.sleep(2) # Consulta cada 2 segundos
64
65 if __name__ == "__main__":
66     asyncio.run(main())
```

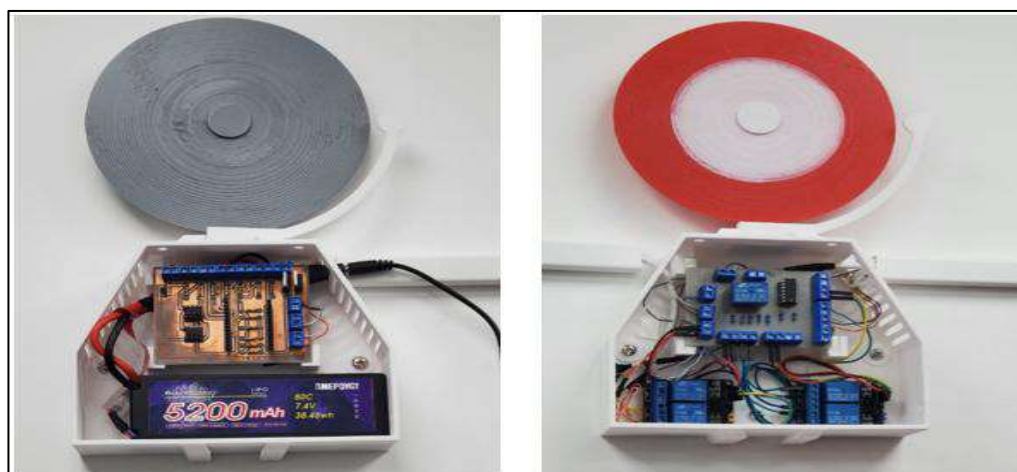
Anexo I. Esquema de correlación entre las diferentes tablas.



Anexo J. Tendido del cable para la conexión con los botones y la sirena.



Anexo K. Disposición final de la carcasa y los componentes.



Anexo L. Manual de operación y funcionamiento del sistema.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabricio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC V01 TLL FISEI

UNIVERSIDAD TÉCNICA DE AMBATO

SISTEMA DE CONTROL Y MONITORIZACIÓN DE ALARMAS COMUNITARIAS PARA EL TALLER DE INNOVACIÓN Y DESARROLLO TECNOLÓGICO FISEI

Manual de usuario



	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

ÍNDICE GENERAL DE CONTENIDOS

PORTADA	i
ÍNDICE GENERAL DE CONTENIDOS	1
ÍNDICE DE FIGURAS.....	2
OPERACIÓN Y UTILIZACIÓN DEL SISTEMA	4
1.1 Objetivo general del Manual	4
1.1.1 Objetivo específico	4
1.2 Puesta en marcha del sistema	4
1.2.1 Ejecución de la máquina virtual	4
1.2.2 Ejecución de los diferentes scripts en Visual Studio Code	6
1.2.3 Activación de la aplicación “Terminal”	9
1.3 Operación y funcionamiento del sistema	11
1.3.1 Generación de alertas	11
1.3.2 Apagado del sistema	16

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

ÍNDICE DE FIGURAS

Figura 1. Ejecución de la máquina virtual.	5
Figura 2. Dirección IP a la que estamos conectados.	5
Figura 3. Apertura de los scripts con Visual Studio Code.	6
Figura 4. Apartados para ejecutar los servicios del sistema.	7
Figura 5. Servidor en línea.	7
Figura 6. Apartado para la generación de las alarmas.	8
Figura 7. Apartado para la gestión de las alarmas generadas.	8
Figura 8. Bot de Telegrama para envío de mensajes.	9
Figura 9. Conexión entre los diferentes dispositivos a través de la terminal.	10
Figura 10. Script de localización.	10
Figura 10. Script sensores, vinculados a la activación de los botones físicos.	10
Figura 12. Script para la activación de los diferentes tonos de la sirena.	11
Figura 13. Medios para activar el sistema.	11
Figura 14. Notificación generada por el botón de pánico.	12
Figura 15. Desactivación de la sirena.	12
Figura 16. Mensaje recibido en Telegram.	13
Figura 17. Alarma detallada.	13
Figura 18. Apartados de la alarma detallada.	14
Figura 19. Generación de la alarma detallada.	14

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

Figura 20. Notificación de la alarma detallada. 15

Figura 21. Mensaje recibo en Telegram de la alarma detallada..... 15

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

OPERACIÓN Y UTILIZACIÓN DEL SISTEMA

1.1 Objetivo general del Manual

Proporcionar instrucciones claras y detalladas sobre el uso adecuado del sistema implementado dentro del taller de Innovación y Desarrollo Tecnológico FISEI.

1.1.1 Objetivo específico

- Facilitar el uso del sistema a través de una explicación detallada paso a paso.

1.2 Puesta en marcha del sistema

Este apartado hace referencia a todos aquellos servicios que deben ser ejecutados para que el sistema funcione adecuadamente. Para ello se deben instalar diferentes programas tales como; Visual Studio Code, Terminal “la cual nos va a permitir conectarnos a la base de datos”, MySQL Workbench, Sistema operativo Ubuntu “que fue instalado dentro de una máquina virtual para este caso”, Máquina virtual, Mosquito, Node.js.

1.2.1 Ejecución de la máquina virtual

Una vez instalados los diferentes programas, iniciamos con la verificación de la red que se empleará para ejecutar el sistema, la misma se puede verificar dentro de la máquina virtual. La figura a continuación muestra el código que se debe utilizar para visualizar la IP de la conexión.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

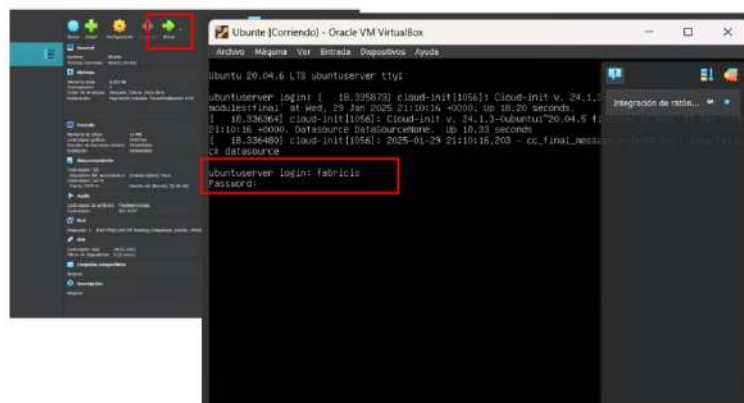


Figura 1. Ejecución de la máquina virtual.

Previamente debemos crearnos un usuario y contraseña, la cual nos va a permitir el ingreso al sistema, una vez dentro se debe emplear el comando *ifconfig* para poder ver la dirección IP a la que estamos conectados tal cual como se muestra en la Figura 2.

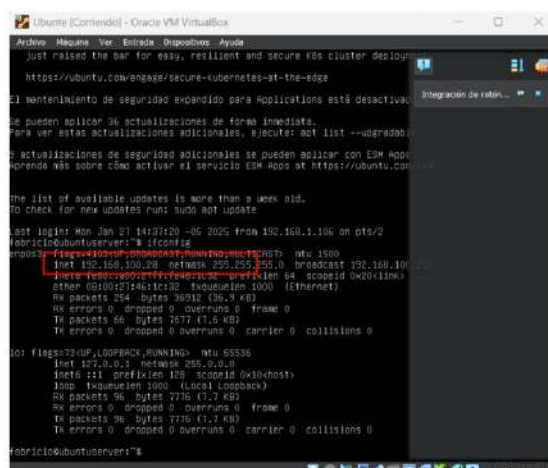


Figura 2. Dirección IP a la que estamos conectados.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

1.2.2 Ejecución de los diferentes scripts en Visual Studio Code

Una vez verificada la dirección IP del internet al que estamos conectados, procedemos a iniciar la aplicación de editor de código “en este caso Visual Studio Code”. Abrimos una nueva terminal y buscamos los documentos que requerimos, en mi caso la carpeta *Tesis*, contiene los código “*server_alarmas*, *notificacion* y *alarmas*”, que voy a ponerlos en servicio, tal como se puede apreciar en la siguiente figura.

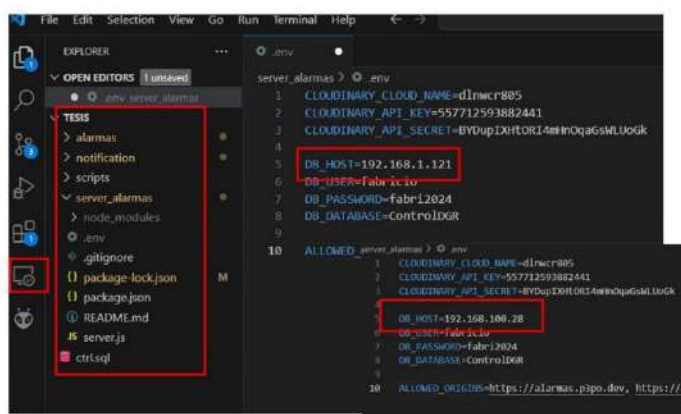


Figura 3. Apertura de los scripts con Visual Studio Code.

En la Figura 3, se puede apreciar que originalmente la dirección que está dentro del archivo *.env*, es diferente a la que se presenta en la máquina virtual, en ese caso debemos cambiar la dirección que estaba anteriormente, tal cual como se apreciar en a figura anterior.

Una vez realizado ese cambio abrimos *4 Git bash* debido a que se pondrán en marcha los diferentes servicios del sistema, incluyendo el *Bot de Telegram*, los cuales se encuentran en la parte inferior derecha del programa, la Figura 4 muestra como se visualiza estos apartados.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

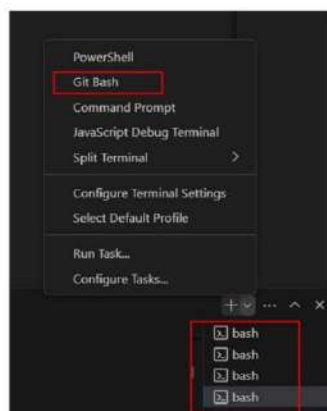


Figura 4. Apartados para ejecutar los servicios del sistema.

Dentro de cada *bash* debemos ejecutar a través de comando el funcionamiento de cada programación, para ello las siguientes imágenes muestra lo que se debe ingresar en cada apartado.



Figura 5. Servidor en línea.

En la anterior figura se puede apreciar los comando necesarios para poder iniciar el servidor; una vez ingresadas las instrucciones el nombre del *bash* se configura con lo que está ejecutando. Seguidamente se activa el scripts de alarmas, como se puede apreciar en la Figura 6.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI



Figura 6. Apartado para la generación de las alarmas.

Al igual que en el caso anterior cuando se ingresan los diferentes comandos, el bash adquirir su respectivo nombre en base a lo que ejecuta, en este caso el número 3001, es el puerto a través de la cual está corriendo el botón de pánico. Para poder visualizar la interfaz se debe dar clic en la dirección que nos aparece tal cual se muestra en la figura anterior. Posteriormente activamos los códigos para recibir las notificaciones, la Figura 7 muestra su respectiva ejecución, con su aparatado para los administradores dentro del diseño web, el cual se ejecuta en el puerto 3002.



Figura 7. Apartado para la gestión de las alarmas generadas.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

En la Figura 7 se puede apreciar que existen dos apartados para recibir las notificaciones del sistema, una de ellas se encarga solo de atender a las alertas generadas por los botones de pánico físicos y del botón de pánico de la aplicación web, mientras que el segundo apartado se encarga de la gestión de las alertas detalladas que podrían generarse. Finalmente activamos el Bot de Telegram, el cual va a enviar un mensaje de texto sobre la situación generada, detallando información sobre el tipo de alerta, la ubicación y en caso de ser detallada la alerta también se podrán adjuntar imágenes sobre la situación. La Figura 8 presenta una visualización de este apartado para mejor comprensión.

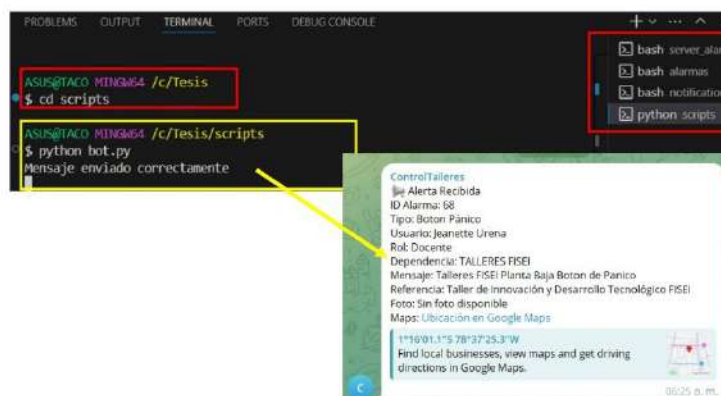


Figura 8. Bot de Telegrama para envío de mensajes.

1.2.3 Activación de la aplicación “Terminal”

Esta aplicación nos permite establecer la conexión entre el servidor y los diferentes microservicios. Para iniciar debemos abrir 3 terminales de *Git Bash*, y en cada uno de ellos debemos conectarnos a través de SSH “sirve para establecer una conexión remota segura entre dispositivos a través de una red, permitiendo la administración y control de servidores o sistemas sin necesidad de acceso físico”.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

La Figura 9 presenta los comandos empleados para establecer dicha conexión. Los cuales involucran la dirección IP con la que estamos trabajando y la contraseña designada.

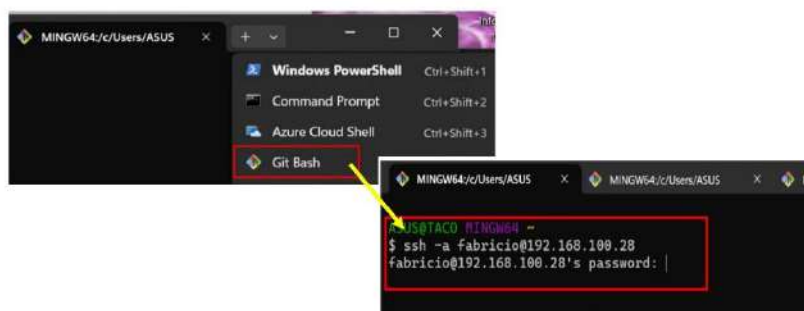


Figura 9. Conexión entre los diferentes dispositivos a través de la terminal.

Una vez establecida la conexión con las respectivas credenciales en las tres ventanas antes mencionadas, debemos poner en funcionamiento los diferentes recursos. La Figura 10, esta relacionada con la localización que el sistema requiere para su activación, puesto que si no se encuentra dentro de los límites establecidos para el taller, el sistema no se ejecuta.

```
Last login: Wed Jan 29 16:10:48 2025
fabricio@ubuntuuserver:~$ cd scripts/
fabricio@ubuntuuserver:~/scripts$ python3 location.py
```

Figura 10. Script de localización.

En la segunda ventana vinculamos el scripts de sensores con el servidor, este script se encarga de controlar la activación de los botones físicos del sistema, tanto de la planta baja como la del piso 1. La Figura 11 muestra su ejecución.

```
fabricio@ubuntuuserver:~$ cd scripts/
fabricio@ubuntuuserver:~/script$ python3 sensores.py
sensores.py:23: DeprecationWarning: Callback API version 1 is deprecated, update to later
client = mqtt.Client()
Conectado al servidor MQTT. Esperando mensajes...
```

Figura 11. Script sensores, vinculados a la activación de los botones físicos.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

Finalmente en la tercera ventana se debe poner en marcha la activación de la sirena, ya sea en su primer tono o en su segundo tono. Para ello, la figura a continuación muestra de manera gráfica el script relacionado con dicha funcionalidad.

```

Last login: Wed Jan 29 18:46:05 2025 from 192.168.100.13
fabrizio@ubuntu-server:~$ cd scripts/
fabrizio@ubuntu-server:~/scripts$ python3 sirenas.py
sirenas.py:20: DeprecationWarning: Callback API version 1 is deprecated, update to latest
client = mqtt.Client()

```

Figura 12. Script para la activación de los diferentes tonos de la sirena.

1.3 Operación y funcionamiento del sistema

Una vez terminado los pasos para poner el sistema en ejecución, se debe prestar especial atención a lo que el sistema realiza durante su activación.

1.3.1 Generación de alertas

Las alertas se pueden generar de 4 maneras diferentes, mediante los botones físicos instalados en la edificación, una señal de la estación Bosh, o a través de la aplicación web botón de pánico. Todas estas fuentes se pueden apreciar en la siguiente figura.



Figura 13. Medios para activar el sistema.

a. Notificación generada por los botones de pánico

Cuando los botones físicos son presionados, o la aplicación web es utilizada en su modo de botón de pánico, los administradores del sistema reciben una notificación generada en la parte de botón de pánico, tal cual se puede apreciar en la Figura 14.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI



Figura 14. Notificación generada por el botón de pánico.

Segundos tras la activación del botón de pánico, se emite la notificación sonora, en otras palabras la sirena comienza a sonar en su primer tono, la misma debe ser desactivada por los administradores, para ello deben ingresar a la notificación generada y presionar el apartado de resuelto, la Figura 15 presenta la interfaz que los administradores visualizan cuando ingresen al panel.



Figura 15. Desactivación de la sirena.

Adicionalmente este apartado contiene cierta información básica de sobre el suceso generado, la misma que es enviada a través de un mensaje a la aplicación de mensajería

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

Telegram, la figura presentada a continuación muestra lo mencionado en torno al mensaje recibido.



Figura 16. Mensaje recibido en Telegram.

b. Notificación generada por una alarma detallada

Como se mencionó en el trabajo realizado, en la parte inferior de la aplicación web, existe una apartado para generar alarmas con información detalla sobre la situación. Dicho apartado se puede apreciar en la siguiente figura.



Figura 17. Alarma detallada.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

Cuando el usuario ingresa a este apartado se topa con la siguiente interfaz, la cual se presenta en la Figura 18, donde existen diversos campos, destinados a recolectar información relevante sobre la situación generada.



Figura 18. Apartados de la alarma detallada.

Una vez seleccionado el tipo de emergencia, agregado información adicional sobre el evento y de manera opcional subida una foto, la notificación que los administradores reciben se muestra en la parte derecha de la interfaz, tal cual se presenta en la siguiente figura.



Figura 19. Generación de la alarma detallada.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

En este tipo de alarma, la sirena no se activa de manera automática como en el caso del botón de pánico, en cambio los administradores pueden activar y desactivar el segundo tono de la sirena de ser necesario, todo ello en base a la evaluación de la situación acontecida, la Figura 20 muestra la interfaz con esta opciones.



Figura 20. Notificación de la alarma detallada.

De igual manera que en el anterior caso, se envía un mensaje de texto a la aplicación de Telegram, divulgando la emergencia generada. La misma se puede apreciar en la Figura 21, mostrada a continuación.



Figura 21. Mensaje recibo en Telegram de la alarma detallada.

	Manual de operación y utilización del sistema de alarmas comunitarias	
	Elaborador por:	Fabrizio Tacoamán
	Revisado por:	Ing. Franklin Salazar, MSc.
	Fecha:	29/01/2025
	Código:	SDAC_V01_TLL_FISEI

1.3.2 Apagado del sistema

Por cuestión del mantenimiento o limpieza el sistema debe ser apagado de manera correcta, en base a las siguientes recomendaciones:

- Primero deshacer la conexión entre el servidor y los microservicios.
- Segundo, detener la ejecución de los diferentes scripts en el editor de código Visual Studio Code.
- Tercero, cortar el suministro eléctrico que alimenta a la placa electrónica.
- Cuarto, en caso de limpieza a la placa electrónica, tener especial cuidado al momento de retirar la tarjeta de desarrollo ESP 32.
- Quinto, si las entradas físicas a la placa han sido removidas “los cables”, tener cuidado al momento de su reconexión.