UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA

CARRERA DE TELECOMUNICACIONES

DESARROLLO DE UNA PLATAFORMA PARA LA GESTIÓN Y MONITOREO DE ALARMAS RESIDENCIALES INTEGRADAS A UN SISTEMA DE ALARMA COMUNITARIA

Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones

AUTORES: JONNATHAN MAURICIO AYABACA ESPINOZA
PABLO FERNANDO COCHANCELA SOLÓRZANO

TUTOR: ING. JUAN PAÚL INGA ORTEGA, MgT.

Cuenca – Ecuador
2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Jonnathan Mauricio Ayabaca Espinoza con documento de identificación N° 0106526049 y Pablo Fernando Cochancela Solórzano con documento de identificación N° 0105393953; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 12 de septiembre del 2023

Atentamente,

Jonnathan Mauricio Ayabaca Espinoza	Pablo Fernando Cochancela Solórzano
0106526049	0105393953

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Jonnathan Mauricio Ayabaca Espinoza con documento de identificación N° 0106526049 y Pablo Fernando Cochancela Solórzano con documento de identificación N° 0105393953, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: "Desarrollo de una plataforma para la gestión y monitoreo de alarmas residenciales integradas a un sistema de alarma comunitaria", el cual ha sido desarrollado para optar por el título de: Ingeniero en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 12 de septiembre del 2023

Atentamente,

Jonnathan Mauricio Ayabaca Espinoza	Pablo Fernando Cochancela Solórzano
0106526049	0105393953

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Juan Paúl Inga Ortega con documento de identificación N° 0104166491, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UNA PLATAFORMA PARA LA GESTIÓN Y MONITOREO DE ALARMAS RESIDENCIALES INTEGRADAS A UN SISTEMA DE ALARMA COMUNITARIA, realizado por Jonnathan Mauricio Ayabaca Espinoza con documento de identificación N° 0106526049 y por Pablo Fernando Cochancela Solórzano con documento de identificación N° 0105393953, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 12 de septiembre del 2023

Atentamente,

Ing. Juan Paúl Inga Ortega, MgT.

0104166491

AGRADECIMIENTOS

A mi querida madre, quien a lo largo del tiempo me ha brindado un apoyo y amor incondicional. Gracias a su constante respaldo, he logrado alcanzar este punto crucial en mi carrera. Los numerosos sacrificios que ha hecho en términos económicos han sido un testimonio de su dedicación. Su ejemplo de perseverancia y entrega ha sido mi mayor fuente de inspiración. A mi familia, quienes siempre me han rodeado con un apoyo incondicional, consejos sabios y orientación, nunca dejándome enfrentar los desafíos solo.

Este logro lleva consigo el reflejo de lo que ellos me enseñaron. Este logro es tanto suyo como mio.

Por Jonnathan Mauricio Ayabaca Espinoza

Deseo expresar mi sincero agradecimiento a mi familia, con especial gratitud hacia mi querida esposa, Alicia, y mi amada hija, Romina. Ambas han demostrado una inquebrantable paciencia y comprensión, incluso en los momentos en que mi presencia a su lado fue limitada debido a mis compromisos académicos. Su apoyo inquebrantable fue un motor fundamental que me impulsó a alcanzar la ambiciosa meta que me propuse.

Asimismo, no puedo pasar por alto el invaluable respaldo brindado por mis padres, quienes, de diversas maneras, contribuyeron a mi camino hacia el éxito. Mi reconocimiento también se extiende a mis amigos de la carrera, en particular a Leonardo, Gabriel y, de manera especial, a Jonnathan. A lo largo de esta travesía, siempre estuvieron ahí para respaldarme y comprender mi situación.

Quiero expresar mi más profundo agradecimiento a mi amigo y compañero de tesis, cuya incansable dedicación y apoyo nunca flaquearon especialmente en los momentos más desafiantes.

Por último, quiero extender mi gratitud a todos aquellos que fueron parte de esta travesía: mi familia, mis amigos y mis profesores. Especial mención merece mi tutor, el Ingeniero Juan Inga Ortega, quien generosamente compartió su vasto conocimiento y mostró una paciencia inquebrantable a lo largo de todo el recorrido académico. Su guía fue esencial para alcanzar este logro.

Por Pablo Fernando Cochancela Solórzano

DEDICATORIA

Dedico este trabajo con profundo cariño a mis abuelos, quienes han sido mi constante ejemplo de lucha y sabiduría. Agradezco infinitamente sus consejos, su apoyo y el amor de padres que me han brindado,También quiero extender esta dedicatoria a mi primo, un modelo a seguir en mi vida, cuyo respaldo ha sido incondicional en cada paso que he dado.

Dedicatoria de Jonnathan Mauricio Ayabaca Espinoza

De manera especial, dedico este trabajo a mi esposa y a mi hija, quienes, con su amor y comprensión, han sido fundamentales para que logre alcanzar esta importante meta en mi vida. A lo largo de este extenso recorrido, siempre demostraron una inquebrantable paciencia y me brindaron un constante aliento para que no desfalleciera. Es importante destacar que este logro también les pertenece, ya que sin su apoyo inquebrantable, no habría sido posible.

Dedicatoria de Pablo Fernando Cochancela Solórzano

Índice general

Índice de figuras

Índice de tablas

Resumen

El objetivo de este proyecto es indagar y desarrollar una plataforma para la gestión y monitoreo de alarmas residenciales con capacidad de integrarse a un sistema de alarma comunitaria, con el propósito de reducir la delincuencia y los robos en el ámbito domiciliario, en especial en la ciudad de Cuenca, donde se ha observado un aumento significativo de estos delitos, generando inseguridad y preocupación entre los propietarios. Así, para este trabajo se logró con éxito la decodificación de una alarma básica marca DSC utilizando el módulo ESP8266 y se procedió a desarrollar e integrar dicha alarma en una plataforma de alarma comunitaria, empleando el dispositivo Cubecell GPS AB02S con tecnología LoRaWAN.

Se incluyó también el uso de actuadores como luces y sirenas para disuadir los actos delictivos. Además, de las alarmas residenciales se implementaron botones de pánico móviles mediante la tecnología LoRaWAN usando el dispositivo Cubecell GPS AB02S para enviar alertas hacia el sistema de alarma comunitaria. Esto busca generar una reacción veloz en casos de alertas urgentes y permite ampliar la cobertura de la alarma comunitaria en un rango de al menos 2 Km.

Para el desarrollo de la red LoRaWAN, se utilizó un servidor alojado en AWS llamado ChirpStack. Este servidor es fundamental para mantener un registro y acceso a la información recopilada por los dispositivos finales.

En adición, mediante el protocolo MQTT, se logró enviar eficientemente la información recolectada hacia el aplicativo móvil que fue desarrollado en Flutter. Dentro de la aplicación móvil, se generan alertas a través de notificaciones push para alertar a los usuarios sobre actos delictivos o situaciones de peligro. Esto brinda más recursos para el uso de alertas con el

afán de elevar la seguridad y protección a los usuarios que en este caso estén registrados en la plataforma.

Palabras clave: Alarma; ChirpStack; Flutter; IoT; LoRaWan; MQTT; Notificaciones Push.

Abstract

The objective of this project is to investigate and develop a platform for the management and monitoring of residential alarms that can be integrated into a community alarm system, to reduce crime and burglaries in the home, especially in the city of Cuenca, where there has been a significant increase in these crimes, generating insecurity and concern among homeowners.

Thus, for this work we successfully decoded a basic DSC alarm using the ESP8266 module and proceeded to develop and integrate this alarm into a community alarm platform, using the Cubecell GPS AB02S device with LoRaWAN technology. The use of actuators such as lights and sirens to deter criminal acts was also included. In addition to the residential alarms, mobile panic buttons were implemented through LoRaWAN technology using the Cubecell GPS AB02S device to send alerts to the community alarm system. This seeks to generate a fast reaction in cases of urgent alerts and allows extending the coverage of the community alarm in a range of at least 2 Km.

For the development of the LoRaWAN network, a server hosted on AWS called ChirpStack was used. This server is essential to keep a record and access to the information collected by the end devices.

In addition, through the MQTT protocol, the information collected was efficiently sent to the mobile application developed in Flutter. Within the mobile application, alerts are generated through push notifications to alert users about criminal acts or dangerous situations. This provides more resources for the use of alerts to increase the security and protection of users who are registered on the platform.

Keywords: Alarm; ChirpStack; Flutter; IoT; LoRaWan; MQTT; Push Notifications.

Antecedentes

En la actualidad el Ecuador se encuentra en un estado de búsqueda por reducir los efectos de la delincuencia e inseguridad, aspectos que han afectado al sector domiciliario provocando miedo, incertidumbre y molestia a los habitantes de bien. Entre los actos delictivos se encuentran robos a mano armada y principalmente sustracción de bienes como; joyas, electrodomésticos, dinero, etc. En un barrio muchas veces una persona no cuenta con la posibilidad de generar alertas para ser socorrida o pueda contrarrestar cualquier actividad sospechosa, ya que la comunicación para pedir socorro, se da en forma oral.

Además, en el Ecuador, el problema de robos a domicilios, lamentablemente, ha aumentado en 27.5 % en 2022, con un promedio mensual 677 domicilios, siendo octubre el mes con mayor saldo registrado con 771 [1].

Según lo mencionado anteriormente acerca de la situación en Ecuador, el robo de domicilios presenta problemas relacionados a (i) el peligro de padecer lesiones o perder la vida por parte de los propietarios, (ii) la pérdida de bienes y dinero que a cada persona le costó obtener con esfuerzo y dedicación; que en muchos casos ya no pueden ser adquiridos nuevamente y (iii) la tranquilidad de cada persona que se ve afectada porque no puede vivir en calma sabiendo que corre el peligro de ser asaltado su hogar.

Justificación

El desarrollo tecnológico actual, brinda herramientas importantes y de grandes prestaciones como son las redes de sensores inalámbricos (de sus siglas en inglés *Wireless Sensor Networks*, WSN), ya que entre sus principales ventajas esta su asequible y eficiente consumo de energía. Además, gracias a la movilidad que permiten las redes inalámbricas, las WSN cuentan con flexibilidad a la hora de implementarlas y facilitan el proceso de escalabilidad. Todas estas características las hacen óptimas para aplicaciones del paradigma conocido como Internet de las Cosas (de sus siglas en inglés *Internet of Things*, IoT), tanto domiciliarias como industriales; por ejemplo, en hogares inteligentes, monitoreo de tráfico, automatización industrial, entre otras.

Entonces, frente a la problemática relacionada con la seguridad domiciliaria y barrial, el presente proyecto surge como propuesta que permita generar una alternativa tecnológica basada en IoT que busca aportar al incremento de la seguridad colectiva dentro de urbanizaciones, barrios o comunidades, dando una respuesta inmediata por parte de los moradores ante un evento delictivo, con el fin de velar la protección de las familias que residen en estas propiedades, proteger sus bienes e integridad física. Dado lo anteriormente expuesto, este trabajo reside en diseñar e implementar una plataforma para la gestión y monitoreo de alarmas residenciales que a diferencia de otras propuestas, el presente proyecto propone la capacidad de ser integradas a un sistema de alarma comunitaria utilizando comunicación inalámbrica para enviar alertas a los dispositivos móviles de los propietarios, identificando el domicilio o lugar donde sucedió el evento.

De esta manera, con este proyecto se busca integrar de manera colectiva a todos los residentes del barrio dentro de un sistema de alarma comunitaria

no convencional, pues este sistema generará distintas alertas informativas para todos los usuarios, los mismos que aportaran de manera solidaria ayuda en una situación de riesgo mejorando su convivencia e incrementando la paz y tranquilidad.

Objetivos

Objetivo General

- Desarrollar una plataforma para la gestión y monitoreo de alarmas residenciales integradas a un sistema de alarma comunitaria.

Objetivos específicos:

- Establecer el estado del arte de arquitecturas para sistemas embebidos y servicios en la nube orientados a la gestión y monitoreo de sistemas de seguridad; además de dispositivos y actuadores disponibles en el ámbito de IoT.
- Diseñar la plataforma de gestión y monitoreo de alarmas residenciales y alarma comunitaria.
- Implementar la plataforma de gestión y monitoreo de alarmas residenciales y alarma comunitaria
- Probar el correcto funcionamiento de la plataforma implementada a través de pruebas de operación.

Introducción

Esta plataforma de gestión y monitoreo de alarmas residenciales integradas a un sistema de alarma comunitaria emerge como respuesta a la inseguridad y delincuencia que va en aumento en el Ecuador; siendo el sector domiciliario uno de los mas afectados provocando miedo, incertidumbre y molestia a sus propietarios. En este sentido, frente a esta problemática de que los delincuentes sustraigan bienes de valor o provoquen daño físico a los propietarios de los inmuebles, con esta plataforma podemos ofrecer una alternativa de bajo costo para monitoreo de la residencia y a su vez enviar una alerta hacia la alarma comunitaria obteniendo una respuesta de ayuda inmediata. Para el desarrollo de este proyecto utilizamos tecnología de comunicación inalámbrica LoRaWAN para comunicar el sistema de alarma residencial con el sistema de alarma comunitaria debido a la distancia que pemite alcanzar, diferenciando este proyecto de otros que utilizan comunicación por radio frecuencia (RF).

Algunos sistemas de alarmas comunitarias basados en RF alcanzan distancias máximas de cobertura de 150 metros, en esta propuesta se logra comprobar una distancia mínima de 2 Km gracias al uso de LoRaWAN.

Además, este trabajo plantea utilizar tecnología moderna, basada en WSN ya que usan dispositivos autónomos siendo capaces de adaptarse a requerimientos de alcance, potencia, velocidad de transferencia de información, reduciendo el consumo de energía como indica [2].

La finalidad del proyecto es el diseño e implementación de una plataforma para la gestión y monitoreo de alarmas residenciales integrada con un sistema de alarma comunitaria para urbanizaciones, barrios o comunidades (rurales o urbanas); el mismo que consta de dos sistemas: el primer sistema

(sistema de alarma residencial), el cual permite el monitoreo de un domicilio y además cuenta con actuadores (sirena, luces) para disuadir actividades sospechosas de forma remota. En caso de existir algún evento de intrusión o activación del botón de pánico se enviara una alerta hacia el segundo sistema (sistema de alarma comunitaria) a través de comunicación inalámbrica LoRaWAN. El segundo sistema consta de botones de pánico inalámbricos distribuidos en lugares estratégicos dentro del área de cobertura; en caso de recibir una alerta sea de un domicilio o botón de pánico, se envía una notificación a cada propietario y se presentara esta información en una pantalla ubicada en un lugar que sea visible para todos.

Los dos sistemas constan en una aplicación móvil, que permite la visualización de las notificaciones generada por estos sistemas, así como su gestión y monitoreo.

La tabla 1 resume una comparativa con trabajos similares resaltando al final los elementos propios que resaltan este trabajo.

Tabla 1: Resumen de las investigaciones vinculadas con el proyecto de tesis actual

Referencias	Gestión de Uso									Conectividad				
	Compatible con alarmas ya instaladas	Permite conectividad con Alarma Comunitaria	Monitoreo Vía Web	Monitoreo Vía Dispositivos Móviles	Basada en IoT	Botón de Pánico	Solo Monitoreo	Monitoreo y Gestión	Monitoreo, Gestión y Control de Actuadores	RF (433MHz)	LAN / WAN	Conectividad Celular (Móvil)	WiFi	LoRaWAN
Quevedo, 2007 [3]		✠					✠							
Rengel, 2012 [4]		✠				✠					✠			
Yunga, 2019 [5]	✠				✠						✠			
Orozco, 2017 [6]				✠			✠				✠	✠		
Villacrés, 2016 [7]				✠								✠		
Gómez, 2018 [8]		✠	✠			✠	✠			✠				
LightSYS 2,2013 [9]			✠	✠				✠		✠	✠	✠		
Alcom Max LTE, 2023 [10]		✠	✠	✠	✠			✠		✠		✠		
NEO,2017 [11]				✠				✠			✠			
Linseg,2018 [12]		✠								✠				
ESTE PROYECTO	✠	✠		✠	✠	✠			✠			✠	✠	✠

Capítulo 1

Conceptualización y Estado del Arte

Puesto que este trabajo tiene como finalidad buscar una solución a la problemática de seguridad en la sociedad, este primer capítulo busca abordar de forma breve conceptos necesarios que expliquen y respalden la razón detrás de la elección de las diversas tecnologías empleadas en la creación de este proyecto.

1.1. Sistemas de seguridad

Un sistema de alarma tiene diferentes elementos que lo componen y que permiten localizar, sensar, diferenciar y notificar un evento de intrusión en un inmueble. Estos eventos son anunciados por diferentes medios como una llamada telefónica por red telefónica fija o móvil o a través del protocolo IP (de sus siglas en inglés *Internet Protocol*, IP). Cuando se activa la sirena se da aviso tanto a un operador de monitoreo, como a la localidad cercana al inmueble o a los mismos propietarios [6].

En las siguientes sub secciones, se exponen los principales sensores usados en la operación de alarmas:

1.1.1. Sensor infrarrojo de movimiento

Este tipo de sensor infrarrojo pasivo se encarga de detectar movimiento (de personas o animales) por medio de rayos infrarrojos dentro de un área. Estos rayos no son visibles por el ojo humano, son emitidos en forma de línea recta y tienen la facilidad de reflejarse sobre cualquier superficie brillante. Cuando

existe un movimiento con suficiencia detectable, la radiación infrarroja que emiten las personas o animales, alteran la cantidad de rayos infrarrojos dentro de su campo de acción. A esto se le conoce como pasivo porque no transmite radiaciones, más bien este las recibe [13].

1.1.2. Sensor Magnético

Un sensor magnético de apertura tiene un sistema de funcionamiento que es relativamente sencilla y se basa en un elemento electrónico fijo y un imán, que al estar lo suficientemente cerca, genera un campo magnético entre ellos que cierra un circuito, al separarse el campo magnético deja de existir entre el elemento fijo y el imán por lo que el circuito se abre [7].

1.1.3. Keybus

Conocido como bus de datos que permite la comunicación entre el panel de alarma y diferentes módulos como son: teclado de Diodo Emisor de Luz(de sus siglas en inglés *Light Emitting Diode*, LED), teclado de pantalla de cristal líquido (de sus siglas en inglés *Liquid Crystal Display*, LCD), sistema Global de Comunicaciones Móviles (de sus siglas en inglés, *Global System for Mobile*, GSM), Comunicador IP, módulos inalámbricos, etc. Siendo la comunicación en ambos sentidos.

Cada modulo posee cuatro terminales que deben ser conectados a los cuatro terminales *keybus* del panel de alarma y tienen la siguiente configuración: (AUX+), (AUX-) son los que proporcionan energía a los diferente módulos, mientras que los terminales (YEL), (GRN) son el reloj y datos respectivamente. Cabe recalcar que todos los módulos se cablean en paralelo y deben ser conectados directamente al panel cableado [14].

1.2. Redes de sensores inalámbricos

La red de sensores inalámbrica (de sus siglas en inglés *Wireless Sensor Networks*, WSN) es una red que contiene numerosos dispositivos colocados

en áreas geográficas específicas utilizados para controlar diversas variables de medición, entre estas el sonido, temperatura, vibración, movimiento, entre otras [15]. Las WSN en contraste de las redes inalámbricas comunes, están formadas por nodos o dispositivos que tienen recursos limitad de cómputo, memoria y energía [16]. Cada nodo hace uso de comunicación inalámbrica a través de una pasarela de datos (en inglés *Gateway*) que actúa como una pasarela de enlace de los sensores y la red de datos lo cual permite procesar la información. Este tipo de tecnología es considerada como una de las más prometedoras del futuro a nivel mundial; sin embargo, al tener nodos distribuidos posee ciertas ventajas y desventajas al momento de desplegar este tipo de red que deben ser tomadas en cuenta [17].

1.2.1. Elementos que conforman una WSN

Las redes de sensores están formadas por dispositivos pequeños denominados nodos sensores que se encuentran dispersos en determinada área de trabajo. Estos nodos se utilizan para recolectar e enviar información a través de un *gateway* hacia una estación base, donde esta información pueda ser almacenada o tratada para su posterior análisis [18].

A continuación se explica cada elemento que conforma una WSN con la finalidad de que se comprenda el diseño que se va a trabajar.

Nodos Sensores

Denominados *motes* en ingles, son dispositivos electrónicos ligeros, pequeños y de bajo consumo energético con la capacidad de captar, procesar y transmitir información del entorno en el que se encuentren hacia un receptor [18].

Estos dispositivos finales están compuestos por diferentes componentes, los cuales se ilustran en la figura 1.1. Además, se proporciona una descripción general de cada uno de estos componentes, basándonos en las fuentes citadas [18], [19].

1. Alimentación: La alimentación de un nodo sensor generalmente utiliza

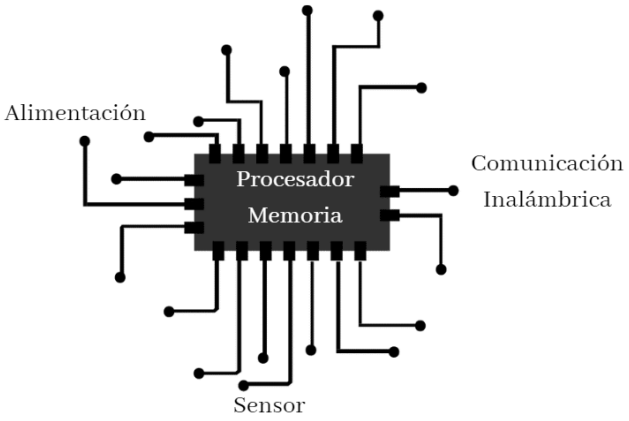

Figura 1.1: Estructura genérica de un nodo sensor inalámbrico [Fuente Autores].

baterías por motivo que las WSN ejercen su trabajo sobre áreas de gran escala. En otros casos de uso, donde es posible usar la red de corriente publica, se opta por esta. En la actualidad el uso de paneles solares a representado una nueva forma de alimentación para los nodos sensores. Por lo general el uso de energía sucede en la etapa de propagación, y en menor medida en el desarrollo de información y utilización de los sensores.

2. Comunicación por RF: Es sin duda de gran importancia del nodo sensor pues permite intercambiar datos por medio de radiofrecuencia con el fin de conectarse con dispositivos presentes en su rango de cobertura.

3. Tratamiento: Es un elemento presente en el nodo sensor que tiene como trabajo el procesamiento de los datos que se reciben desde los dispositivos sensores y se encargan de resguardarlos en la memoria.

4. Sensor: Son equipos que al recibir excitación física como humedad, gases, temperatura, etc, responden con una señal eléctrica ya sea de corriente o voltaje.

5. Memoria: Componente donde se almacena toda la información, su capacidad depende de la necesidad de la aplicación y su propósito sean estos almacenar datos recogidos por la aplicación o almacenar el programa del dispositivo.

Punto de agregación de datos y Puerta de enlace de datos

Para las redes WSN, en general, los nodos sensores envían su información hacia un nodo concentrador conocido como un punto de agregación de datos (PAD) que opera como una puerta de enlace de datos [20], [21]. La puerta de enlace de datos, también conocida en inglés como *gateway*, es un nodo especial que no contiene el elemento sensor. Su principal objetivo es realizar un puente entre la red de sensores y un nivel superior de red como la red de datos que permite el accesos a la información obtenida por los nodos sensores [18], [19].

En diferentes aplicaciones, con el fin de brindar escalabilidad a una WSN de gran tamaño, debido al número de nodos y/o por la escala de la zona geográfica que se busca cubrir, es mejor usar un PAD que recolecte la información de un grupo de nodos sensores y así cada uno de estos concentradores envíe paquetes de datos de mayor tamaño hacia un nodo con la capacidad de permitir conectividad a volúmenes grandes de datos como son las estaciones base de una red móvil celular [20]. En redes de menor escala o que trabajan con tecnologías de largo alcance, el uso de un PAD suele no ser tan necesario y de forma directa se un equipo *gateway* que permita el acceso de forma directa a la Internet.

Cabe indicar que, en general un *gateway* siempre cuenta con dos tarjetas de comunicación, cada una con un protocolos distinto. Una primer tarjeta con la tecnología inalámbrica para comunicarse con los nodos encargados del sensado; dicha tecnología de comunicación suelen usar bajas tasas de transmisión. La segunda tarjeta de comunicación cuenta con una tecnología inalámbrica que permita una mayor tasa de transmisión, en muchos casos, usada para acceso de última milla como WiFi si la WSN es doméstica por ejemplo. No obstante, en el caso de un PAD, la segunda tarjeta suele ser de tipo móvil celular para enlazarse con la correspondiente estación base si el proveedor de acceso a la Internet es móvil.

1.2.2. Topologías de una WSN

Una topología se refiere a la configuración de cada nodo o componente que conforman una red de sensores. Esta configuración puede ser de tipo física *(hardware)* o lógica usando direcciones de capa 3 según el modelo abierto de interconexión de sistemas (de sus siglas en inglés Open Systems Interconnection, OSI). En resumen, la topología permite establecer como se conectan los elementos de una red; en el caso de una WSN, se habla de topología física para identificar como es que los datos que son transmitidos a través de esa configuración y así en el servidor de datos saber cómo y de donde proviene la información [18].

Cada topología presenta desafíos, ventajas y desventajas por lo cual deben ser correctamente seleccionada. Existen diferentes topologías para formar una WSN, las tres más utilizadas se presentan a continuación: en estrella, en malla, e híbrida estrella-malla [22].

Topología en estrella

Es un sistema donde todos los nodos sensores son idénticos y la puerta de enlace capta la información de todos ellos dando solo un salto. Los nodos no intercambia información entre ellos, si es necesario utilizan el gateway. El gateway además se usa con el fin de enviar datos hacia afuera y permitir gestionar la red [18].

La topología estrella es la de menor gasto energético pero limitada por la distancia de trasmisión. En caso de que uno de los nodos falle la información de este se perderá debido ya que solo tiene un camino de comunicación [22].

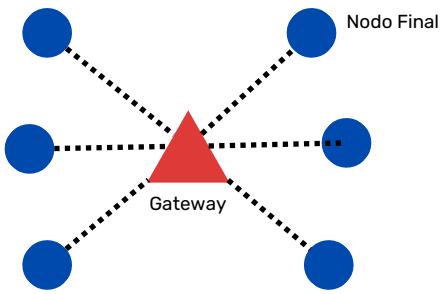

Figura 1.2: Topología en estrella [Fuente Autores].

Topología en malla

Es un sistema donde todos los nodos puedes recibir y enviar información a otro nodo o puerta de enlace a esto se denomina multisalto. Esta topología es tolerante al fallo de uno de sus nodos ya que cada uno de ellos tiene diferentes caminos para comunicarse con la puerta de enlace. La velocidad de envió de información depende del número de nodos y la distancia entre ellos [18].

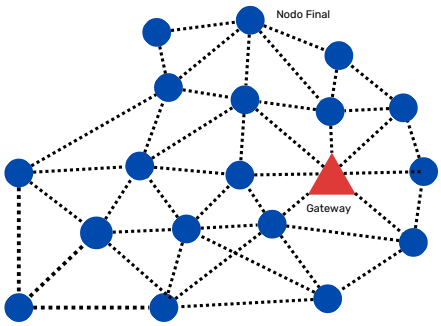

Figura 1.3: Topología en malla [Fuente Autores].

Híbrida estrella-malla

Esta topología busca combinar las ventajas de la facilidad y bajo consumo de una topología de estrella con la posibilidad de cubrir grandes distancias y reorganizarse ante fallos de sensores miembros que posee como ventaja las topología en malla [18].

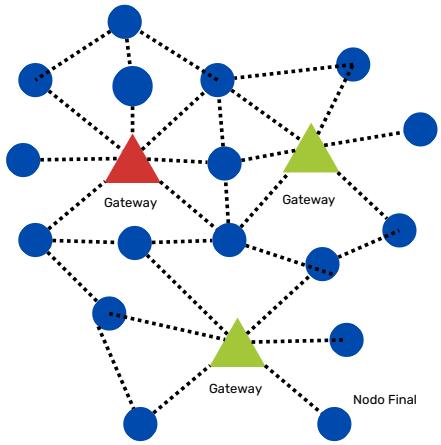

Figura 1.4: Topología híbrida estrella-malla [Fuente Autores].

1.3. Internet de las cosas

El internet de las cosas (de sus siglas en inglés *Internet of Things*, IoT)es la conexión de diferentes dispositivos de la vida cotidiana del ser humano a la Internet. Estos dispositivos pueden ser domésticos comunes como las bombillas de luz, dispositivos para de la salud, equipos médicos, prendas y accesorios propios de cada persona que son inteligentes, además, estructuras en ciudades inteligentes [23]. IoT abarca a todos los dispositivos físicos que envían y reciben información a través de una red inalámbrica donde la intervención humana es mínima, esta tecnología con el transcurso de los años ha transformado la forma en que vive y trabaja el ser humano ofreciendo distintos beneficios y oportunidades [24].

Un ejemplo de IoT son los pulsantes inteligentes que son equipos instalados en un hogar o una oficina que controlan distintos dispositivos, como aire acondicionado, persianas y luces. IoT es la terminación de la tecnología presente en el hogar y su conexión a internet [25].

1.3.1. Arquitectura IoT

En la actualidad existen diferentes arquitecturas en desarrollo de tres, cuatro y cinco capas. Sin embargo, la más adecuada que explica la metodología utilizada en este proyecto es la de cinco capas en referencia mostradas en [26], [27]. De manera detallada presta a continuación cada una de estas:

1. **Capa de percepción o dispositivo:** En esta capa se encuentran los sensores o dispositivos que recogen y trasmiten información a la capa de red.

2. **Capa de red:** Esta capa cumple con trasmitir a través de un medio inalámbrico o cableado la información que capto de los sensores hacia la unidad de procesamiento.

3. **Capa de middleware:** Esta capa se basa en los resultados del procesamiento de información para la toma de decisiones. Además, es la responsable de la gestión de servicios y tiene relación con la base de datos.

4. **Capa de aplicación:** La responsabilidad de esta capa es la gestión total de la información que ha fue procesada por la capa *middleware*.

5. **Capa empresarial** Construye modelos de negocio, gráficos, flujo gráfico. Se basa en los datos recibidos de la capa aplicación. Ayuda a determinar problemas futuros y estrategias comerciales.

La Figura 1.5, muestra de manera más amplia el orden de las capas de la arquitectura resaltando que, los sensores son la capa más baja de la arquitectura y las aplicaciones y uso de los resultados permiten mejorar la toma de decisiones en diferentes nichos de mercado bajo diversos de negocio [28].

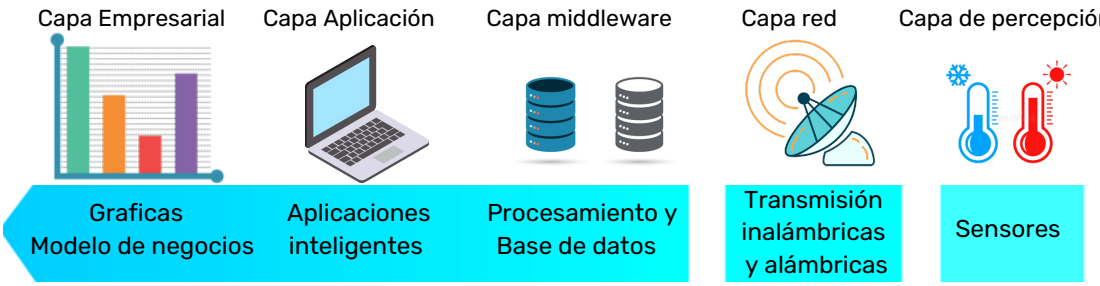

Figura 1.5: Capas de la Arquitectura IoT [Fuente Autores].

1.4. Comunicaciones inalámbricas

La red de comunicaciones inalámbricas requiere ondas de radio para enlazar distintos equipos, sustituyendo a las redes cableadas y de difícil acceso, las redes inalámbricas dan la posibilidad a equipos remotos conectarse con gran facilidad independientemente de que estos equipos se encuentren a grandes distancias[29]. Las comunicaciones inalámbricas se dividen según su rango de alcance en las de corto y largo alcance.

1.4.1. Tecnología inalámbrica de corto alcance

Este tipo de tecnología de corto alcance permiten la comunicación en un rango de distancia muy corto permitiendo así soluciones pequeñas, con eficiencia energética y de bajo costo [29]. Entre las mas destacadas esta la

fidelidad inalámbrica (de sus siglas en inglés *Wireless Fidelity*, Wi-Fi), Bluetooth y Zigbee [30].

1.4.2. Tecnología inalámbrica de largo alcance.

Este tipo de tecnología es utilizado para medición inteligente donde se necesitan bajos niveles de consumo y costo, pero requieren comunicación IoT a larga distancia. Las tecnologías más reconocidas son las redes datos como 2G/3G, 4G, 5G y LTE. A su vez tenemos la red de área amplia de baja potencia (de sus siglas en inglés *Low Power Wide Area Networks*, LPWAN) la cual cuenta con un diseño que permite un alto rango de cobertura pero con tasa pequeña de datos [30].

En la figura 1.6 se representa el cotejo de tecnologías conocidas como corto y largo alcance.

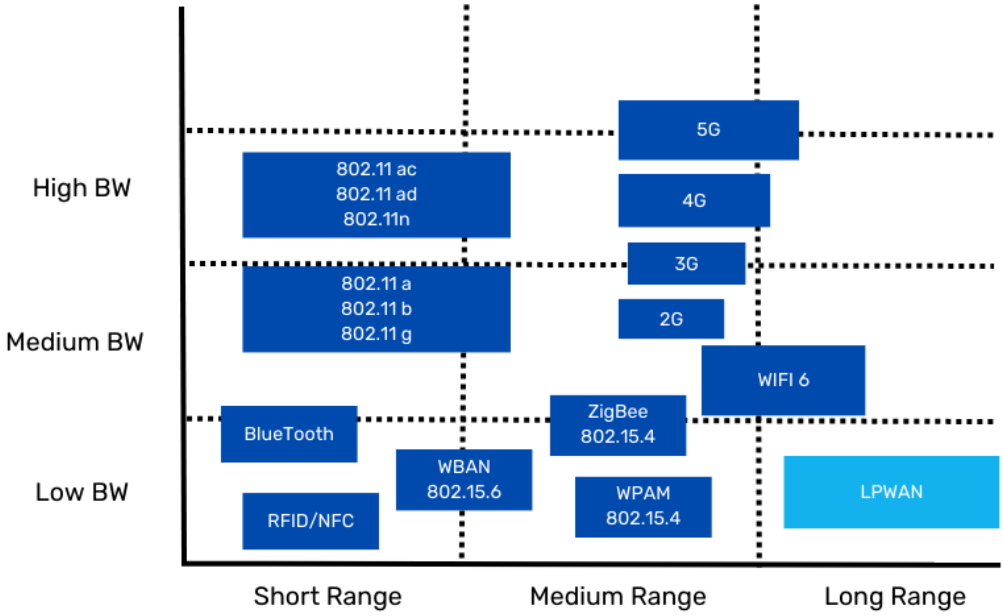

Figura 1.6: Comparación de tecnologías inalámbricas de acuerdo a corto y largo alcance [Fuente Autores].

1.5. Red de área amplia de baja potencia (LPWAN)

Este tipo de redes son tecnologías de comunicación inalámbrica que permiten transmitir datos entre dispositivos y una estación base separados por largas distancias en metros o kilómetros con un bajo costo energético [31].

Las principales características de estas tecnologías en la actualidad posibilitan un mayor despliegue en iniciativas IoT ya que son diseñadas para este entorno, permite colocar varios nodos distribuidos en áreas grandes, alimentados por baterías que duran varios años y sin necesidad de grandes infraestructuras como el cableado [30]. En la figura 1.7 se describe de manera general la comunicación LPWAN.

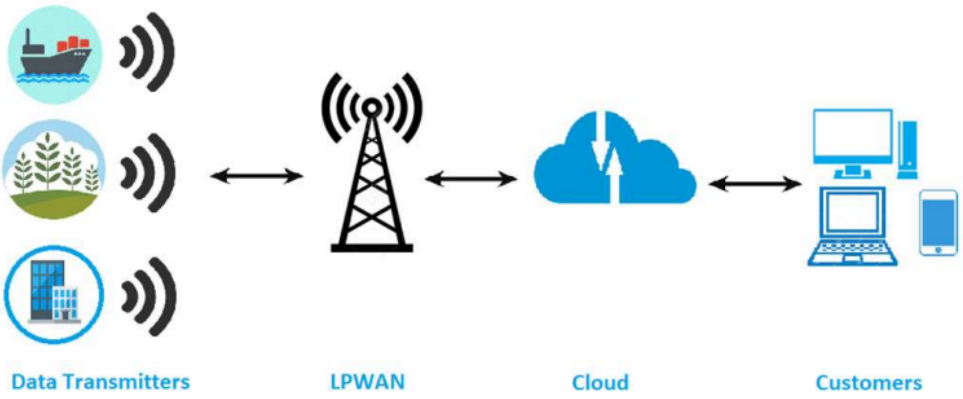

Figura 1.7: Descripción general de la comunicación LPWAN [Fuente Autores].

Existen 3 tipos de tecnología LPWAN más utilizada para IoT los cuales son [32]:

1.5.1. Sigfox

Su objetivo principal es enviar pequeñas cantidades de datos a través de Internet. Utiliza modulación por desplazamiento de fase binaria (de sus siglas en inglés *Binary Phase Shift Keying*, BPSK) como modulación de banda ultra estrecha (de sus siglas en inglés *Ultra Narrow Band*, UNB) cuenta con tasa de bits de 100 bps. Además proporciona mayor sensibilidad recibida y reduce el costo de diseño de la antena, además de comunicación bidireccional. Esta tecnología admite un millón de dispositivos conectados con una cobertura de 30 a 50 Kilometros en zonas rurales y de 3 a 10 kilometros en zonas urbanas [30].

1.5.2. Lora

La tecnología LoRa transmite las señales en banda ISM (de sus siglas en inglés *Industrial, Scientific and Medical*, ISM) (868MHz en Europa, 915MHz en Norte América y 433MHz en Asia) el usar frecuencias bajas le permite cubrir áreas más amplias, especialmente cuando el área está llena de interferencia o los nodos se encuentren dentro de los edificios [30]. LoRa emplea una modulación espectro ensanchado basado en señales de tipo chirrido (de sus siglas en inglés *Chirp Spread Spectrum*, CSS) con elevada resistencia a la interferencia ya que las señales obtienen un bajo nivel de ruido [32]. LoRa en general se despliega bajo una topología en estrella esto permite cubrir entre 3-5 km en zonas urbanas y de 10-20 km en zonas rurales.

1.5.3. NB IoT

Es una tecnología de radio estandarizada por el proyecto Asociación de Tercera Generación (de sus siglas en inglés *3rd Generation Partnership Project*, 3GPP), se desarrolló con la finalidad de emplear una gran cantidad dispositivos IoT. NB IoT se centra significativamente en equipos de alta confiabilidad, bajo consumo de energía, alta seguridad en la red y bajo costo [30].

Consigue coexistir con el sistema global de comunicaciones móviles GSM y evolución a largo plazo (de sus siglas en inglés *Long Term Evolution*, LTE) bajo bandas de frecuencia licenciadas [32]. A continuación, en la tabla 1.1 se muestra una comparación de algunas características de Sigfox, LoRa y NB IoT.

Tabla 1.1: Comparación entre las principales tecnologías LPWAN [30].

Tecnología	Modulación	Frecuencias	Ancho de Banda	Máxima Tasa de Transmisión	Cobertura Máxima	
					Áreas Urbanas	Áreas Rurales
SigFox	BPSK	902MHz 868MHz	200KHz	100bps	10Km	50Km
LoRa	CSS	ISM no Licenciadas: 868 MHz en Europa, 915MHz en Norte América, 433MHz en Asia	125KHz	50Kbps	5Km	20Km
Nb - IoT	QPSK	Frecuencias licenciadas con LTE	200KHz	200Kbps	1Km	10Km

1.5.4. LoRaWAN

Para el presente trabajo se selecciona la tecnología inalámbrica LoRaWAN por su alcance, eficiencia energética, capacidad de penetración en interiores, escalabilidad, costos más bajos y su idoneidad para aplicaciones de IoT, lo que la convierte en una tecnología atractiva para muchas soluciones urbanas.

LoRaWAN es un protocolo de comunicación y la arquitectura del sistema de red que incluye funciones seguras, móviles y de bajo consumo energético. En la actualidad es uno de los protocolos más utilizados en el concepto de IoT. La velocidad de datos que permite este protocolo utilizado en Smart cities y aplicaciones industriales está en el rango de 0.3Kbps hasta 50Kbps considerado estable para el envío de datos de sensores para IoT en tiempo real [33], [34].

LoRa/LoRaWAN están diseñados para diferentes sensores y aplicaciones que envían pequeñas cantidades de datos a larga distancia y en distintos lapsos de tiempo [34].

Arquitectura LoRaWAN

Este tipo de red se basa en una topología estrella con múltiples Gateways, los datos de cada trasmitidos por cada nodo sensor son recibidas por estos Gateways. Cada *gateway* reenviará el paquete recibido desde el nodo final al servidor de red, a través de una red de retorno sea celular, Ethernet, satélite o Wi-Fi [33].

La Figura 1.8 muestra la arquitectura de una red LoRaWAN que esta conformado por pasarelas (Gateways), los nodos finales (LoRa Thing), servidor de red y servidor de aplicación.

Características principales

Las características que competen a LoRaWAN, de acuerdo a lo presentado en [35], a continuación se presenta un breve resume de cada una de estas:

1. **Consumo ultrabajo:** Este tipo de dispositivos están diseñados para el bajo

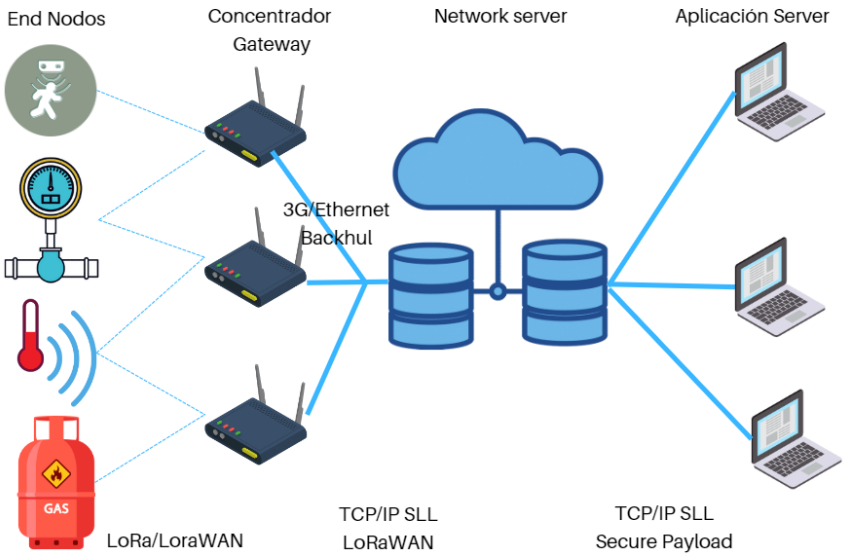

Figura 1.8: Arquitectura de red LoRaWAN [Fuente Autores].

consumo de energía. Utiliza una batería de botón de gran durabilidad.

2. **Largo alcance:** Estos dispositivos LoRaWAN tienen la capacidad de trasmitir y recibir señales de datos a largas distancias (áreas rurales) de 10km y de 3km en áreas pobladas (áreas urbanas). En cuanto a la profundidad de trasmisión las redes LoRaWAN proporciona cobertura en interiores intensa, fácilmente cubre edificios de varios pisos.

3. **Espectro sin licencia:** Licencia de espectro de frecuencia no costosa para implementación de una red LoRaWAN.

4. **Geolocalización:** No existe necesidad de Sistema de posicionamiento global (de sus siglas en ingles *Global Positioning System*, GPS). En una red LoRaWAN mediante triangulación se puede ubicar dispositivos finales.

5. **Alta capacidad:** Se puede implementar redes LoRaWAN públicas y privadas con el mismo hardware (Gateway, sensores, antenas) y software (reenviadores de paquetes UDP, software de estación básica) ya que estos manejas millones de mensajes por diferentes puestas de enlace.

6. **Seguridad de extremo a extremo:** La comunicación de LoRaWAN es segura utiliza cifrado AES-128 además actualizaciones de firmware por aire lo cual permite actualizaciones de forma remota.

7. **Roaming:** los dispositivos finales de LoRaWAN pueden realizar traspasos continuos de una red a otra.

8. **Bajo costo:** se utiliza infraestructura mínima, nodos finales de bajo costo y software de código abierto.

Clases de dispositivos LoRaWAN

En una red LoRaWAN, se pueden identificar tres clases de dispositivos finales, cada una con capacidades específicas, según se menciona en la referencia [36]. Estas clases son conocidas como Clase A, Clase B y Clase C.

1. **Dispositivos LoRa Clase A:** Está planificada para que la transmisión corra por cargo propio del dispositivo final. Esta formada por dos ventanas de recepción, permitiendo que solo se acepten datos de recepción después de una transmisión completada correctamente (también conocidos en inglés como *downlink*). La finalidad del dispositivo es la eficiencia desde el punto energético ya que se adecua a aplicaciones que solo requiere enviar datos, gastando menor energía.

2. **Dispositivos LoRa Clase B:** La creación de ventanas de recepción en esta clase de dispositivos permite recibir datos sin la necesidad una transmisión previa. Se aumenta la capacidad de recibir datos y se planifica el tiempo en el que el dispositivo debe abrir la ventana de recepción, las consecuencias de este es el aumento de consumo eléctrico y la capacidad de recepción.

3. **Dispositivos LoRa Clase C:** Estos dispositivos están en modo de recepción permanentemente y solo son interrumpidos cundo existe una transmisión, presenta una menor latencia en la conexión entre dispositivos finales a cambio de un mayor consumo energético.

Activación de Dispositivos Finales en LoRaWAN

Existen dos tipos de activación de dispositivos finales en LoRaWAN : activación por aire (de sus siglas en inglés Over The Air Activaction,

OTAA) y activación por personalización (de sus siglas en inglés Activation By Personalization, ABP). Ambos métodos implican el intercambio de información de seguridad entre el dispositivo final y la red LoRaWAN para garantizar la autenticidad y la confidencialidad de las comunicaciones.

Para el registro y activación de un dispositivo final se utiliza el envió y almacenamiento de de los siguientes parámetros: dirección del dispositivo *(DevAddr)*, identificación de aplicación *(AppEUI)*, llaves de criptografía de sesión y sesión de aplicación *(NwkSkey, AppKey)*.

A continuación se presenta los dos tipos de activación en una red LoRaWAN según lo mencionado en [36].

- **Activación sobre el Aire (OTAA):** En la activación sobre el aire el dispositivo final envía una solicitud de activación a la red LoRaWAN, requiere información de seguridad como su identificador único *(DevEUI)* y claves. La red autentica y registra el dispositivo, asignándole una dirección de red *(DevAddr)*y claves de sesión temporales *(NwkSkey, AppKey)*. Con estos parámetros, el dispositivo puede comunicarse de manera segura con la red.

- **Activación por personalización (ABP):** En la activación por personalización los parámetros de activación se programan directamente en el dispositivo durante su configuración inicial. Esto incluye la dirección de red *(DevAddr)* y las claves de sesión *(NwkSkey, AppKey)* proporcionadas por el operador de la red LoRaWAN. El dispositivo utiliza estos parámetros predefinidos para iniciar las comunicaciones sin necesidad de autenticación adicional.

Uso de frecuencias de LoRaWAN para Ecuador

En Ecuador, la utilización de bandas de frecuencia libres está regulada por la "Norma Técnica de Uso Libre y de Espectro para Uso Determinado en Bandas Libres". La cual corresponde a las frecuencias de 902-928 MHz. La normativa establece las diferentes formas de empleo de frecuencias en este intervalo, así como los límites de transmisión y las limitaciones aplicables a

dispositivos utilizados en los sectores industriales, científicos y médicos [37].

En Ecuador, se pueden usar las bandas de frecuencia regionales de LoRaWAN que se encuentren en dentro de los 902-928 MHz. A continuación la tabla 1.2 muestra las bandas de frecuencia LoRaWAN que podrían ser utilizadas en el Ecuador siempre y cuando no se despliegue con fines comerciales [37].

En la actualidad, ARCOTEL se encuentra en proceso de establecer regulaciones para la implementación de comunicación inalámbrica LoRaWAN en Ecuador. Además, han hecho la recomendación de considerar el trabajo con frecuencias por debajo de los 921 MHz, permitiendo la utilización del espectro a partir de los 902 MHz.

Tabla 1.2: Frecuencias útiles en Ecuador [37].

Nombre	Rango [Mhz]
US915	902-928
AU915	**915-928**
AS1	920-923
AS2	923-925
KR920	920-923

1.6. Sistemas embebidos

Este dispositivo incorpora un computador programable que se caracteriza principalmente por ser de bajo costo y por su consumo de energía y potencia relativamente bajo [38], [39]. Un sistema embebido se compone fundamentalmente por un microprocesador y un software que se ejecuta sobre el mismo; se necesita almacenar el software en algún lugar, y este requerimiento lo puede abarcar la memoria RAM o incluso podría alojarse en la misma organización del chip del procesador. Adicionalmente, este sistema suele contar con varios puertos que le facultan el uso de entradas y salidas que le permiten interactuar con el medio externo [39], [40].

En la tabla 1.3, se presentan las características principales de las placas de desarrollo que podrían ser consideradas para su utilización en este trabajo, con el fin de identificar la opción más adecuada y evaluar su disponibilidad

Tabla 1.3: Comparación de Placas de Desarrollo

Característica	Arduino Uno	ESP 8266	Raspberry Pi Zero	CubeCell AB02S
Procesador	ATmega328P	ESP 8266	ARM1176JZF-S	ARM Cortex-M0
Memoria Flash	32 KB	4 MB	512 MB	128 KB
Memoria RAM	2 KB	520 KB	512 MB	24 KB
Interfaces Inalámbricas	Ninguna	Wi-Fi	Wi-Fi (mediante adaptador)	LoRaWAN, GPS
Interfaces de Comunicación	UART, SPI, I2C	UART, SPI, I2C	I2C, SPI	UART, SPI, I2C
Velocidad del Procesador	16 MHz	80 MHz	1 GHz	48 MHz
GPIO	14	17	40	14
Bibliotecas para Arduino IDE	Amplia selección	Compatibilidad	Limitada	Compatibilidad
Colaboración de la Comunidad	Muy activa	Activa	Muy activa	Activa
Costo Aproximado (USD)	20-25	10-15	15-20	20-25
Disponibilidad en Ecuador	Amplia	Amplia	Media	Limitada

Después de revisar detalladamente las diferentes placas y llevar a cabo una comparación exhaustiva, se ha tomado la decisión de seleccionar la placa de desarrollo de la empresa Heltec, el CubeCell AB02S para la implementación de las alertas para la alarma comunitaria. Esta elección se debe a que el objetivo principal de esta plataforma implica la transmisión a través de una interfaz inalámbrica basada en LoRaWAN. Además, esta placa brinda la capacidad de interactuar con dispositivos externos mediante los pines GPIO, lo que permite interactuar con los demás sistemas de la plataforma. Para llevar a cabo la decodificación de la alarma residencial, se optó por seleccionar la placa NodeMCU. Esta placa cuenta con una interfaz WiFi que posibilita la conexión a Internet y la utilización del protocolo de comunicación MQTT. Este protocolo se utilizará con el propósito de obtener información acerca del estado de la alarma residencial y para administrarla de manera efectiva

Es importante resaltar que, a pesar de que la disponibilidad de la placa CubeCell es limitada en el país, el grupo de investigación en Telecomunicaciones y Telemática (GITEL) dispone de un dispositivo que ha sido utilizado como base para este proyecto. Esta selección estratégica está en consonancia con los objetivos y requisitos particulares de la plataforma, asegurando la viabilidad y funcionalidad del desarrollo propuesto.

1.7. Base de datos

Una base de datos se determina como una máquina abstracta con la capacidad de almacenar grandes cantidades de datos que se relacionan y estructuran. Las bases de datos contienen modelos y características

fundamentales de un sistema, sin tener detalle de implementación pero que permite la consulta de datos a gran velocidad, de acuerdo a características seleccionadas por los usuarios[41].

1.7.1. Funciones de base de datos

Toda base de datos es creada con la finalidad de satisfacer alguna necesidad de una organización, las principales actividades de una base de datos constan en dos funciones: consulta y actualización que según lo menciona en [41] se presenta a continuación.

Funciones de actualización

La transacción que se genera de las funciones de actualización es un cambio de información o datos, de un estado a otro. alternado la información establecida.

Funciones de consulta

Una de las funciones más usadas en un base de datos es la función de consulta. Este tipo de función no modifica la información de ninguna manera, su principal tarea es comprobar si un hecho o grupos de hechos se mantienen en un estado determinado en la base de datos.

Funciones y restricciones

Las funciones de actualización y restricción están interrelacionas, una función de actualización tiene una seria de reglas y acciones asociadas a ella. estas reglas de restricción especifican que pasara si las condiciones son ciertas.

1.7.2. MongodDB

La base de datos MongoDB proporciona alto rendimiento, disponibilidad y automatización escalada debido a su documentación de código abierto, los registros de esta base de dato son un documento compuesto por pares de datos de campo y valor. MongoDB utiliza documentos similares a los objetos JSON,

los datos de cada campo en esta base de datos pueden incluir documentos, matrices y matrices de documentos [42].

Las ventajas que presenta MongoDB por el uso de documentos, de acuerdo con lo presentado en [42], a continuación se presenta un breve resumen:

1. Los documentos u objetos corresponden a tipos de datos nativos en muchos lenguajes de programación.
2. Los documentos y matrices integrados reducen la necesidad de uniones costosas.
3. El esquema dinámico admite polimorfismo fluido.

Características de MongoDB

A continuación, se resumen las características principales de MongoDb de acuerdo con lo presentado en [42]:

- **Alto rendimiento:** MongoDB facilita una alta persistencia de rendimiento de datos, utiliza modelo de datos integrados que permite a los índices admitir consultas mas rápidas y puedan incluir en ellas claves de documentos incrustados y matrices.
- **Lenguaje de consulta enriquecido:** MongoDB admite consultas enriquecidas que permite admitir operaciones de lectura, escritura, búsqueda de texto y agregado de datos.
- **Alta disponibilidad:** denominado conjunto de replicas,conjunto de servidores que mantienen el mismo grupo de datos, proporcionando redundancia y aumenta la disponibilidad de datos.
- **Escalabilidad horizontal:** MongoDB proporciona escalabilidad horizontal.Distribuye datos a través de un grupo de máquinas.

1.8. Desarrollo de aplicaciones móviles

Una aplicación móvil se refiere a software establecido específicamente para ser utilizados por usuarios a través de equipos móviles. Estas aplicaciones

móviles tienen como objetivo principal brindar asistencia al cliente en diferentes campos y, en general, buscan reducir el tiempo que conlleva realizar una actividad de la vida cotidiana. Además, también se utilizan para la gestión y control de eventos, entre otros usos.

La arquitectura de una aplicación móvil se compone de dos componentes esenciales: el *backend* y el *frontend*, los cuales trabajan en conjunto para asegurar el correcto funcionamiento de la aplicación. Por tanto, resulta fundamental seleccionar una solución viable que abarque ambas partes de manera efectiva. A continuación se explica estas dos partes [43].

1.8.1. Backend

El término "Backend", hace referencia a la capa de aplicación que permite el acceso a los datos y funcionalidades que no son visibles directamente para el usuario. En esta capa, se encuentran la operatividad, seguridad y optimización de los procesos asociados a la página web o aplicación. Además, aquí reside la "business logic", o lógica de negocio, que comprende la lógica interna necesaria para el correcto funcionamiento de la aplicación [43].

Entre las funcionalidades del backend se incluye la capacidad de proporcionar acceso a la información solicitada por el usuario, gestionar y presentar los datos requeridos de acuerdo a los diferentes requisitos, y establecer la comunicación con sistemas externos [44].

En términos sencillos, el desarrollo del backend implica la creación de un servidor y una base de datos. El código correspondiente al backend se ejecuta en el servidor, no en el navegador [45].

1.8.2. Servidor NodeJS

Node.js es un entorno de ejecución de código JavaScript que se utiliza principalmente en el lado del servidor para desarrollar aplicaciones web y backend. A diferencia de JavaScript, que se ejecuta en el navegador del cliente, Node.js permite ejecutar código JavaScript en el servidor, lo que lo convierte en una opción popular para construir aplicaciones web y servicios backend

escalables y de alto rendimiento.

En consecuencia, Node.js opera como un intérprete que facilita la creación de aplicaciones, desempeñando su rol en el lado del servidor. Esta plataforma representa una transformación en la manera convencional de establecer conexiones con los servidores, adoptando el enfoque de programación orientada a eventos. Una de las notables virtudes que caracterizan a esta tecnología es su capacidad para ejecutar tareas de manera asíncrona, lo cual conlleva a una disminución en el tiempo requerido para llevar a cabo diversas operaciones.

Esta agilidad en la ejecución resulta en una eficiencia mejorada en el funcionamiento de las aplicaciones. Considerando la funcionalidad previamente expuesta de la app móvil, es imperativo establecer una conexión con un servidor específico. En esta perspectiva, la incorporación de Node.js resulta invaluable, ya que esta tecnología se encarga de orquestar las conexiones mediante la administración de hilos o subprocesos, minimizando la probabilidad de generar cuellos de botella. Esto, a su vez, garantiza una disponibilidad superior de los servicios y conexiones a la base de datos, lo cual es esencial para mantener un funcionamiento fluido y efectivo de la aplicación.

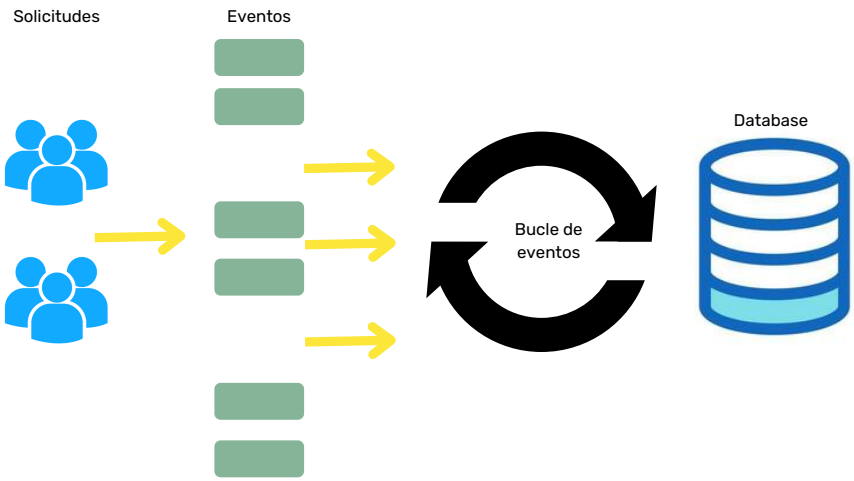

Figura 1.9: Modelo de servidor Node.js [Fuente Autores].

1.8.3. Frontend

El frontend comprende el uso de tecnologías con las cuales el usuario interactúa para utilizar y gestionar una página web o aplicación, también

conocido como el lado del cliente. Su objetivo principal es desarrollar una interfaz gráfica (GUI) que presente la información de manera amigable y accesible al usuario.

Para su desarrollo, es necesario tener conocimientos en técnicas y diseño de interacción con el usuario, con el fin de ubicar estratégicamente los elementos que faciliten su uso de manera fácil y ágil. De esta manera, se busca lograr una interacción eficaz entre el usuario y la aplicación, promoviendo una experiencia fluida y satisfactoria.

En la actualidad, existen una variedad de lenguajes de programación utilizados en el desarrollo del frontend. Algunos de ellos son JavaScript, Java, Vue, TypeScript, React, Dart, Angular, entre otros. Además, también se cuenta con lenguajes de transferencia de información como JSON, Ajax y XML, los cuales posibilitan realizar solicitudes al servidor [43].

1.8.4. Flutter

Flutter se trata de un framework que permite el desarrollo de aplicaciones móviles el cual fue creado por Google. Permite crear y ejecutar aplicaciones en diversas plataformas tales como Android, iOS, escritorio (*Linux, macOS y Windows*) y web y su principal objetivo es permitir a los desarrolladores construir aplicaciones multiplataforma a partir de una única base de código, y esta se compila a código nativo para las plataformas de destino [43], [46].

Maneja Dark para construir toda la aplicación incluso la interfaz de usuario. Entonces, se apoya en la programación orientada a objetos, puntualmente en la composición y herencia de widgets: texto, botones, margenes, paletas de color, etc. Los widgets se construyen a nivel de aplicación lo que permite que el desarrollador pueda personalizarlos o extenderlos de manera sencilla dando como resultado libertad para el diseño de aplicaciones [43], [46].

A continuación se presenta las siguientes características principales que hacen a flutter una de las plataformas mas utilizadas para el desarrollo de aplicaciones móviles descritas en [43]:

- Gratis.

- Código abierto.
- Respaldado y originado en Google.
- Flutter es continuamente mejorado y mantenido por un equipo de desarrolladores tanto de Google como por una amplia comunidad de desarrolladores externos.
- En la actualidad, miles de desarrolladores en distintas organizaciones alrededor del mundo utilizan Flutter para crear aplicaciones en entornos de producción.
- Flutter se caracteriza por su rapidez, ya que se compila directamente en aplicaciones nativas sin necesidad de utilizar WebViews ni puentes de JavaScript.

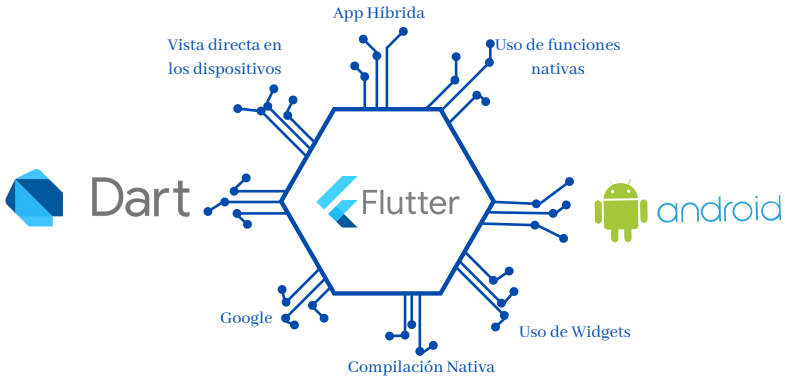

Figura 1.10: Estructura de Flutter [Funte Autores].

1.9. Servicios de monitoreo en la nube.

Se llama servicios en la nube a infraestructuras, plataformas o sistemas de software que se alojan y se ponen a orden de los usuarios a través de Internet. Toda infraestructura, plataforma, sistema de software que son accesibles para un usuario a través de Internet sin descargar software, los servicios en la nube facilitan el flujo de datos a diferentes clientes, fomentan el diseño de aplicaciones y la rapidez al trabajar ya que se puede acceder a estos servicios solamente utilizando un computador y conexión a internet [47].

1.9.1. Clasificación de servicios en la nube

Los servicios en la nube incluyen las soluciones que se mencionan a continuación:

Infraestructura como servicio (IaaS)

Una infraestructura de servicio en la nube brinda a los usuario recursos informáticos de red y de almacenamiento. Para colocar una infraestructura en la nube a disposición de los usuarios, diferentes proveedoras de este servicio separan las funciones informáticas de los elementos de hardware mediante virtualización o máquinas virtuales, a continuación se presentan los siguientes casos de separación según [48]:

- La potencia de procesamiento de las unidades centrales de procesamiento(CPU).
- El procesamiento de gráficos de las unidades de procesamiento de gráficos (GPU).
- La disponibilidad de almacenamiento de datos de los discos duros o los centros de datos.

La separación de estos elementos informáticos brindan a Los usuarios, elementos informáticos y de almacenamiento a través de una red de internet. Este servicio ha permitido hasta la actualidad generar un gran aumento de almacenamiento en la nube, Además permite guardar *Big data* como parte fundamental de IoT [48].

Plataforma como servicio (PaaS)

La plataforma en la nube permite a los usuarios la ejecución de aplicaciones o cualquier tipo de infraestructura informática. Para la implementación de estas plataformas se puede utilizar recursos de hardware: entornos en linea donde se puedan desarrollar código o ejecutar aplicaciones. EL nivel de desarrollo para esta plataforma es superior debido a que se integran tecnologías como: organización en contenedores, la organización de los

sistemas, las interfaces de programación de aplicaciones (API), el enrutamiento, la seguridad, la gestión y la automatización. Además, y no menos importante se debe tener en cuenta la experiencia en línea de usuario [48].

Software como servicio (SaaS)

Conocido como software en la nube ofrece a los usuarios una aplicación en la nube donde se pueda ejecutar cualquier tipo de plataforma. Para el desarrollo de este tipo de Software se requiere una inversión mayor en el desarrollo arquitectura debido a que esta ofrece a sus usuarios una aplicación en línea [48].

Este tipo de software en la nube está diseñado con un enfoque de arquitectura de aplicación ya que combina y a grupa micro servicios como estilo o modo de programación, independiente o conectados indirectamente dentro de contenedores linux que están gestionados por un motor de organización de contenedores. La finalidad de este software en la nube es ofrecer un producto conformado por micro servicios que puedan ser mejorados de manera individual, si afectar al resto [48].

En la figura 1.11 se muestra la clasificación de los servicios en la nube seguían lo explicado anteriormente.

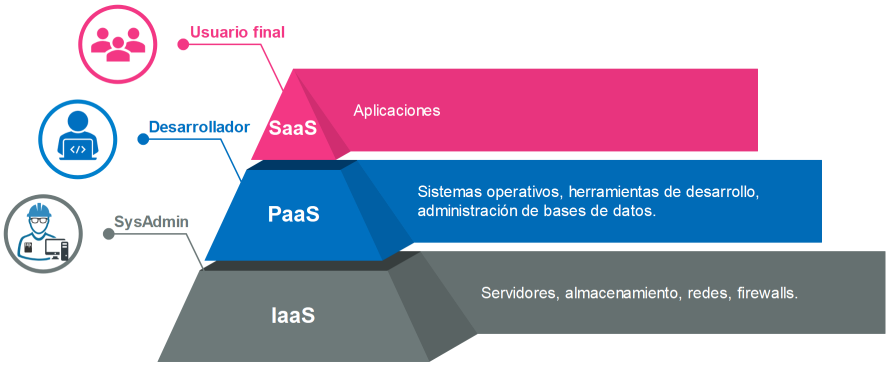

Figura 1.11: Clasificación de servicios en la nube [48].

1.9.2. Funcionamiento de los servicios de nube

Para el funcionamiento de los servicios en la nube los usuarios necesitan:un sistema operativo o computador, conexión a internet que brinden

el acceso a estos servicios [47], [48].

La figura 1.12 muestra el funcionamiento de un servicio en la nube como es el almacenamiento de archivos, Cuando un usuario sube un archivo a la nube, estos se guardan en el espacio que tienen contratado el proveedor de servicio, en este caso se tiene un ejemplo: Dropbox almacena en Amazon.

Figura 1.12: Funcionamiento de servicio en la nube [Fuente Autores].

1.9.3. Ejemplos de servicios en la nube

En la actualidad existen diferentes proveedores de servicios en la nube ofrecen grandes beneficios y prestaciones según la actividad en la que sea requerida. A continuación, se presentan dos de los más reconocidos.

Amazon Web Services

Los servicios web de Amazon, (de sus siglas en ingles *Amazon Web Services*, AWS), es una de las más completas y adaptables que ofrece a sus millones de cliente 200 servicios de centro de datos a nivel mundial. Además, tiene una estrecha relación con las empresas emergentes ya que les brinda la posibilidad de crecer más rápido, incluso se relaciona con compañías grandes y gubernamentales ayudando a reducir costos, aumentando su agilidad e innovación de manera mas rápida [49].

La nube AWS permite activar una máquina virtual con características de almacenamiento, memoria y cantidad de núcleos de vCPU en cuestión de minutos. Una ventaja mas reconocidas de esta plataforma es aprovisionar recursos en la región o regiones donde se adapta al uso de un caso especifico, cuando se a terminado con los recursos, simplemente se pueden eliminar. La escalabilidad y flexibilidad que este posee integradas, puede generar un

aplicación que sirve a un cliente y con el pasar del tiempo escalar con esta misma aplicación sirviendo a próximos clientes [49].

Amazon EC2

La Nube informática elástica de Amazon (de sus siglas en ingles *Amazon Elastic Compute Cloud*, Amazon EC2) proporciona un gran capacidad computacional estable en AWS. el uso de esta elimina la necesidad de un hardware permitiendo el desarrollo de aplicaciones en poco tiempo. Amazon EC2 permite varios servidores virtuales como se necesiten. Además, permite escalar hacia arriba o hacia abajo controlando los requisitos o picos de popularidad, con lo que se reduce la necesidad prever el trafico [49].

Características de Amazon EC2 Las principales características que posee esté según [49], se presentan a continuación:

- Entornos informáticos virtuales, conocidos como instancias.
- Plantillas para instancias preconfiguradas.
- Tipos de instancias como CPU, memoria, almacenamiento y capacidad.
- Inicio de sesión segura para las instancias.
- Volumen de almacenamiento para datos temporales, usados y ubicaciones físicas.
- Firewall donde se especifican protocolos, puertos y rangos de direcciones IP.
- Direcciones IP elásticas.
- Metadatos que crean y asignan recursos de Amazon EC2

Digital Ocean

Digital Ocean es una empresa, que ofrece a sus clientes un hospedaje en la nube con características de escalabilidad permitiendo contar con mayores recursos mientras avanza la estrategia digital. Los servicios de *Digital Ocean*

están basados en la nube, considerada como opción superior a las tradicionales [50].

Este tipo servicio en la nube permite ser usado en diversos proyectos, desde un simple blog hasta aplicaciones complejas tanto en ambientes de producción como de prueba, gracias a esto se pude mencionar que *Digital Ocean* es muy fácil manejar, ya que no requiere tener conocimientos avanzados de hosting tradicional. Algo que merece ser mencionado como destacado es el valor que tiene cada uno de los servicios ya que permite utilizar varias plantillas a diferentes precios [50].

Las características principales se presentan a continuación como se menciona en [50]:

- Servidores privados en la nube.
- Permite alojar diferente tipo de información.
- Permite integrarse con diferentes plataformas de terceros.
- Permite Gestionar bases de datos.
- Consolida varias funciones a través de una sola plataforma.
- La opción de escalabilidad le permite a las compañías. adquirir únicamente lo que requieran en cada etapa de sus proyectos.
- Posee relación de coste-beneficio.

1.10. Ciberseguridad

La ciberseguridad se trata de la protección de todo lo que se preserva en el medio intangible del ciberespacio; en la actualidad las organizaciones tanto públicas o privadas tienen como desafío manejar enormes y crecientes volúmenes de información, los mismos que tienen que ser administrados de forma segura, eficaz y eficiente. Por lo tanto, el tipo de información y su magnitud hace que las organizaciones se conviertan en objetivos preferidos de los ciberataques [51].

En la actualidad, cada vez más los individuos son objeto de ataques cibernéticos. Entre los principales objetivos se encuentran: bases de datos nacionales, sistemas financieros, empresa de desarrollo de ciencia y tecnología, instalaciones estratégicas, entre otros [51].

1.10.1. Seguridad en IoT

Las tecnologías de Internet de las Cosas (IoT) han experimentado un rápido desarrollo debido a su creciente demanda. Estas tecnologías se pueden observar en diversos entornos, como áreas urbanas, entornos industriales, oficinas y hogares. Sin embargo, este rápido proceso de adopción ha generado una falta de formalidad en su implementación, lo que ha llevado a descuidar aspectos cruciales como la seguridad [52].

La seguridad en las aplicaciones IoT es de vital importancia debido a la creciente cantidad de dispositivos conectados y la sensibilidad de los datos transmitidos. A continuación, se presentan algunas consideraciones clave y recomendaciones encontrado en [52], para implementar medidas de seguridad en aplicaciones IoT:

1. **Autenticación y autorización**: Es fundamental asegurarse de que solo los dispositivos y usuarios autorizados puedan acceder a la aplicación IoT. Los mecanismos de autenticación y autorización sólidos son la primera línea de defensa contra ataques no autorizados. Esto se puede lograr utilizando técnicas como contraseñas robustas, autenticación de dos factores y certificados digitales.

2. **Comunicación segura**: La comunicación entre los dispositivos IoT y la plataforma o servidor debe ser segura y encriptada. El uso de protocolos como TLS (de sus siglas en inglés *Transport Layer Security*,TLS) o VPN (de sus siglas en inglés *Virtual Private Network*,VPN) garantiza que los datos transmitidos estén protegidos contra ataques de interceptación y manipulación.

3. **Actualizaciones y parches de seguridad**: Es esencial mantener los dispositivos IoT actualizados con los últimos parches y actualizaciones de

seguridad. Los fabricantes deben proporcionar actualizaciones regulares para abordar nuevas vulnerabilidades descubiertas.

4. **Pruebas de seguridad y auditorías**: Las pruebas de seguridad periódicas y las auditorías son esenciales para identificar y abordar posibles vulnerabilidades. Las pruebas de penetración, los análisis de vulnerabilidades y las auditorías de seguridad ayudan a identificar debilidades y a tomar medidas correctivas.

1.11. Protocolos de comunicación para IoT

El Internet de las Cosas(IoT) ha revolucionado la comunicación entre dispositivos, mejorando la vida diaria del ser humano. Se utilizan protocolos específicos para dispositivos con recursos limitados y redes de baja potencia. A continuación, se presenta una breve descripción de estos protocolos detallados en la referencia [53].

- **CoAP:** CoAP (de sus siglas en inglés *Constrained Application Protocol*) es un protocolo diseñado específicamente para dispositivos con recursos limitados, como sensores y actuadores en IoT. Utiliza un enfoque cliente/servidor permitiendo una comunicación eficiente y liviana en redes con restricciones de energía y ancho de banda, y es ampliamente utilizado en aplicaciones de IoT que requieren una interacción directa entre dispositivos.

- **AMQP:** AMQP (de sus siglas en ingles *Advanced Message Queuing Protocol*) es un protocolo de mensajería orientado a la empresa que permite la comunicación segura y confiable entre dispositivos IoT. Es altamente escalable y se basa en un modelo de cola de mensajes donde los dispositivos envían y reciben mensajes a través de intermediarios llamados “brokers”. AMQP es adecuado para escenarios donde se requiere una comunicación bidireccional y confiable entre dispositivos y aplicaciones en IoT.

- **HTTP:** Aunque HTTP (de sus siglas en ingles *Hypertext Transfer Protocol*) es un protocolo ampliamente utilizado en la web, también se utiliza en IoT para la comunicación entre dispositivos y servidores. HTTP proporciona un enfoque cliente/servidor y se basa en el intercambio de solicitudes y respuestas. Aunque puede ser más pesado en términos de uso de ancho de banda en comparación con otros protocolos mencionados anteriormente, HTTP es ampliamente compatible y se beneficia de la infraestructura existente en la web.

1.11.1. MQTT

El protocolo de mensajería ligero (de sus siglas en inglés *Message Queuing Telemetry Transport*, MQTT) esta diseñado para la comunicación entre dispositivos IoT con comunicación máquina a máquina en redes IoT. Funciona según el modelo de publicación/suscripción, donde los dispositivos publican mensajes en un "BROKER", central y otros dispositivos suscritos reciben esos mensajes. MQTT es ampliamente utilizado en aplicaciones de IoT debido a su bajo consumo de ancho de banda y su capacidad para adaptarse a conexiones de red inestables [54].

Características de MQTT

A continuación, se presentan las principales características de MQTT detalladas en la referencia [55].

1. **Ligero:** MQTT es un protocolo de bajo consumo de recursos, lo que lo hace ideal para dispositivos con recursos limitados, como sensores y dispositivos IoT de baja potencia.

2. **Eficiente:**Utiliza un modelo de publicación/suscripción, lo que permite una comunicación eficiente y orientada a eventos entre los dispositivos conectados.

3. **Conexiones persistentes:** MQTT mantiene conexiones persistentes entre los dispositivos y el broker, lo que facilita la entrega de mensajes incluso en caso de desconexiones temporales.

4. **Calidad de servicio adaptable:** MQTT ofrece diferentes niveles de QoS para garantizar la entrega de mensajes según las necesidades del sistema.

5. **Baja sobrecarga:** El protocolo tiene una sobrecarga de datos mínima, lo que reduce el consumo de ancho de banda y la duración de la batería en dispositivos móviles.

1.11.2. Broker MQTT

Un broker MQTT es un componente fundamental en la arquitectura de comunicación basada en el protocolo MQTT. Actúa como intermediario entre los dispositivos que publican mensajes y los dispositivos que los reciben. A continuación, se describen las características principales de un broker MQTT descritas en [54], [56]:

1. **Funcionalidad de publicación/suscripción**: Un broker MQTT sigue el modelo de publicación/suscripción, donde los dispositivos pueden publicar mensajes en diferentes temas (topics) y otros dispositivos se suscriben a esos temas para recibir los mensajes correspondientes.

2. **Gestión de conexiones**: El broker es responsable de administrar y mantener las conexiones con los dispositivos clientes. Permite que los dispositivos se conecten y desconecten de manera dinámica, garantizando una comunicación continua.

3. **Calidad de servicio (QoS)**: El broker MQTT admite diferentes niveles de QoS para asegurar la entrega confiable de mensajes. Los niveles de QoS incluyen:

    - QoS 0 (Entrega al más rápido esfuerzo): El mensaje se entrega una vez y no se garantiza la confirmación de recepción.
    - QoS 1 (Entrega al menos una vez): Se garantiza que el mensaje se entregue al menos una vez.
    - QoS 2 (Entrega exactamente una vez): Se garantiza que el mensaje se entregue exactamente una vez, evitando duplicados.

4. **Retención de mensajes**: El broker MQTT puede retener los últimos mensajes publicados en un tema. Esto permite que los dispositivos que se suscriban más tarde reciban los mensajes previos y evita la pérdida de información crítica.

5. **Seguridad**: Los brokers MQTT pueden proporcionar mecanismos de seguridad para proteger la comunicación entre los dispositivos. Estos mecanismos pueden incluir autenticación, autorización y encriptación de datos.

6. **Escalabilidad**: Los brokers MQTT deben ser capaces de manejar un alto número de conexiones simultáneas y garantizar un rendimiento adecuado en entornos de IoT a gran escala.

1.11.3. Mosquitto broker

Mosquitto es un broker de mensajes de código abierto con licencia EPL/EDL, que ofrece una solución eficiente para implementar la comunicación MQTT. Es altamente versátil, ya que brinda soporte para múltiples versiones del protocolo MQTT, incluyendo 5.0, 3.1.1 y 3.1. Su diseño ligero y su enfoque en el modelo de publicación/suscripción lo hacen especialmente adecuado para aplicaciones de IoT que involucran dispositivos de baja potencia y móviles [57].

Mosquitto se destaca como una solución sólida y ampliamente adoptada para la implementación de la comunicación MQTT en proyectos de IoT. A continuación, se enumeran algunas de sus características principales, detalladas en la referencia [57]:

1. **Ligero y eficiente:** Mosquitto está diseñado para ser ligero y consumir pocos recursos del sistema. Esto lo hace adecuado para dispositivos con recursos limitados, como microcontroladores y sistemas embebidos. Además, su bajo consumo de recursos garantiza un rendimiento eficiente y una latencia mínima en la comunicación MQTT.

2. **Fácil configuración y uso:** Mosquitto proporciona una configuración sencilla y una interfaz de línea de comandos para la administración.

Su instalación y puesta en marcha son rápidas, lo que facilita la implementación del broker en diversos entornos.

3. **Compatibilidad multiplataforma:** Mosquitto es compatible con múltiples sistemas operativos, incluyendo Linux, Windows, macOS y diversas distribuciones de sistemas embebidos. Esto lo hace versátil y accesible para una amplia gama de dispositivos y sistemas.

4. **Soporte completo de MQTT:** Mosquitto implementa completamente el protocolo MQTT versión 3.1 y 3.1.1, cumpliendo con los estándares establecidos. Ofrece características esenciales como publicación/suscripción, retención de mensajes, niveles de QoS y gestión de sesiones.

5. **Configuración de seguridad:** Mosquitto admite mecanismos de seguridad para proteger la comunicación MQTT. Permite configurar autenticación mediante nombre de usuario y contraseña, así como el uso de certificados TLS/SSL para establecer conexiones cifradas.

6. **Integración y extensibilidad:** Mosquitto cuenta con una API (*Application Programming Interface*,API) que permite su integración con aplicaciones y sistemas existentes. Además, se pueden desarrollar extensiones y complementos personalizados para adaptar el broker a necesidades específicas.

Capítulo 2

Diseño e implementación

El siguiente capítulo empieza por identificar elementos relacionados con los factores de riesgo para el sector domiciliario en la región sur del Ecuador. También, se realiza un análisis conciso de la problemática para gestionar y conectar las diversas bases de datos de usuarios con la plataforma de la alarma, los nodos LoRa y la aplicación móvil. En este contexto, se proporcionan detalles generales sobre las configuraciones de cada elemento utilizado, así como su correspondiente implementación.

2.1. Breve Análisis sobre la Inseguridad Inmobiliaria

La delincuencia e inseguridad en el sector domiciliario se ha convertido en una preocupación significativa en diversas partes del mundo, y su incidencia ha ido en aumento en los últimos años debido a la falta de seguridad en los hogares [58].

En este análisis, se examinarán los factores clave que contribuyen a este problema y se presentarán posibles soluciones para abordarlo de manera efectiva.

2.1.1. Factores de riesgo

Según lo expuesto en [59], los principales factores son:

1. Desigualdad socio-económica: Las áreas con altos niveles de pobreza y desigualdad tienden a tener mayores tasas de delincuencia domiciliaria.

2. Desempleo: La falta de oportunidades de empleo puede llevar a que algunas personas opten por actividades delictivas, incluido el robo a domicilios.

3. Deterioro de los lazos comunitarios: La falta de cohesión social y la pérdida de confianza entre los vecinos pueden crear un entorno propicio para la delincuencia.

4. Acceso a drogas y armas: La disponibilidad de drogas ilícitas y armas de fuego facilita la comisión de delitos en el sector domiciliario.

2.1.2. Tipos de delitos domiciliarios

A continuación se resumen los tipos de delitos que se exponen en [59], los cuales se brinda un abreve explicación:

1. Robo: El robo en viviendas es uno de los delitos más comunes en el sector domiciliario. Los ladrones pueden ingresar a las casas cuando los residentes no están presentes y robar objetos de valor.

2. Allanamiento ilegal: Algunas personas ingresan ilegalmente a las propiedades con la intención de cometer delitos, como robo, vandalismo o agresión.

3. Estafas: Los estafadores pueden aprovecharse de los residentes mediante engaños y fraudes para obtener acceso a sus hogares o información personal.

2.1.3. Delincuencia domiciliaria en el Ecuador

La inseguridad en los domicilios ecuatorianos se ha convertido en un problema social de proporciones alarmantes. Según de la entidad análisis y estadística CEDATOS en estudios realizados en el año 2020, se reveló que un preocupante 65 % de las familias ecuatorianas han sufrido algún tipo de

delito domiciliario o han tenido algún familiar afectado por esta situación. Además, según informes recientes del periódico El Telégrafo, en el año 2021 se ha registrado un incremento del 15 % en estos incidentes [60].

Estas cifras alarmantes reflejan claramente la falta de seguridad en la sociedad ecuatoriana [58].

2.1.4. Posibles Respuestas para Mitigar la Inseguridad

Existe ideas diferentes que se pueden generar para poder invitar o combatir este problema este problema, a continuación presentamos algunas de ellas enfocadas en esta área de implementación expuestas en [58].

- **Mejorar la seguridad física:** Instalar sistemas de seguridad como alarmas, cámaras de vigilancia y cerraduras de alta calidad puede disuadir a los delincuentes y ayudar a capturar pruebas en caso de un delito.
- **Fomentar la participación comunitaria:** Promover la participación de los residentes en programas comunitarios y establecer redes de vecinos vigilantes puede fortalecer los lazos comunitarios y ayudar a detectar y prevenir la delincuencia.

El presente proyecto tiene como objetivo integrar estas dos posibles soluciones: la mejora de la infraestructura física y la participación activa de las personas, en una única plataforma. En la siguiente sección, presentaremos el diseño e implementación de esta plataforma, que busca combinar de manera efectiva ambos enfoques para abordar el problema de la delincuencia y la inseguridad en el sector domiciliario.

2.2. Diseño de la plataforma

El diseño consta de dos sistemas: el primero se encuentra en la residencia del propietario, el cual ya posee una alarma básica representada por una vivienda como se observa en la parte superior de la figura 2.1. El segundo sistema es el denominado “alarma comunitaria” como se puede observar en la

parte baja de la figura 2.1 y recibe los datos de forma inalámbrica en su propio concentrador

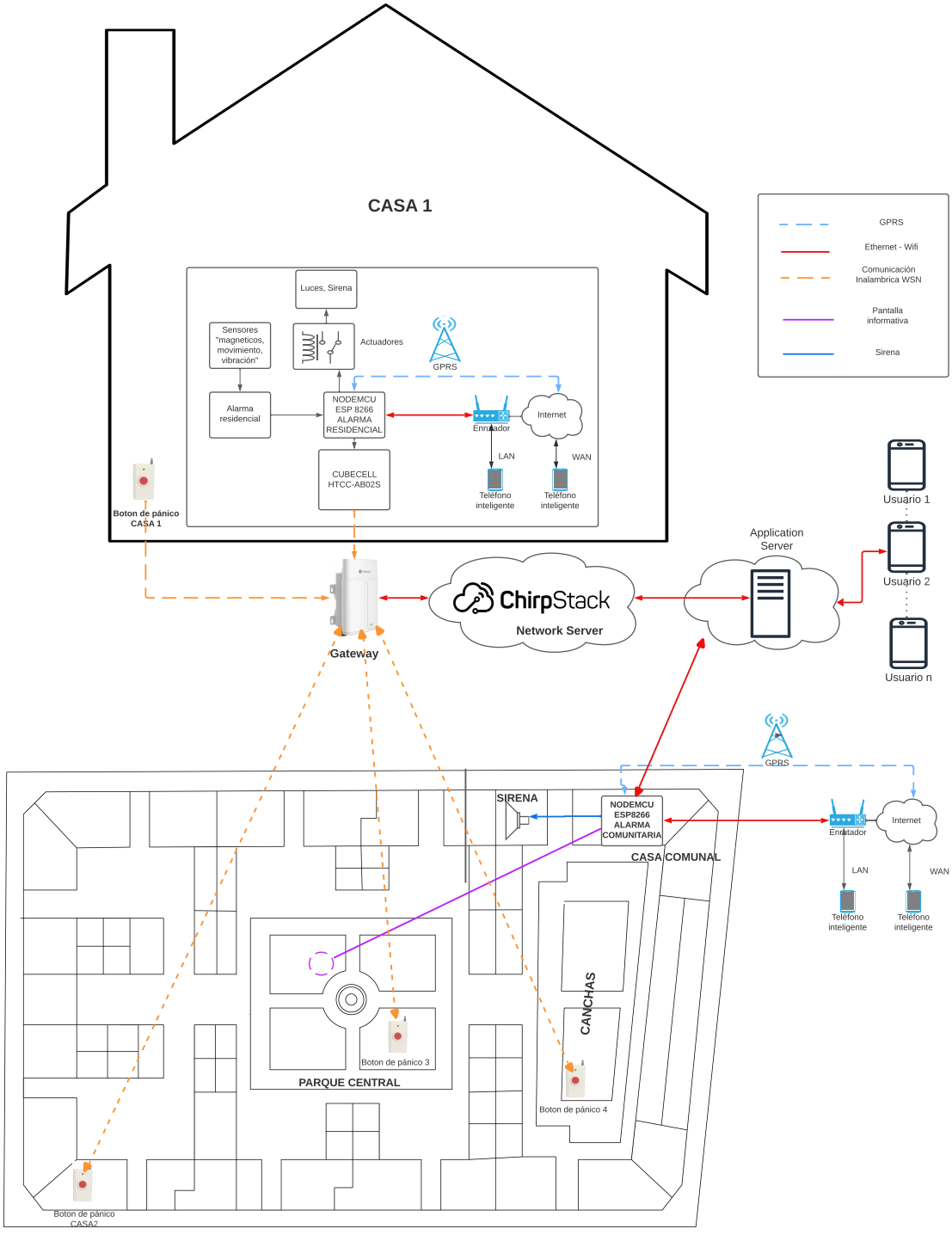

Figura 2.1: Infografía de la topología de funcionamiento [Fuente Autores].

Sistema 1: Alarma Residencial

Esta alarma residencial cuenta con sensores de intrusión, como sensor de movimiento y sensor magnético, para detectar movimientos o actividades sospechosas. Además, dispone de un botón de pánico que permite activar la alarma en caso de emergencia.

Toda la data generada por el sistema de la alarma residencial es procesado por la placa de desarrollo ubicado en el hogar, representado por el bloque "*NODEMCU* Alarma residencial", en la figura 2.1. Estos datos generados, son enviados hacia broker Mosquitto MQTT para poder visualizar tanto de manera gráfica, a través de iconos que diferencian las alertas generadas en la residencia incluyendo el estado de la alarma, como de forma escrita mediante notificaciones de texto en una aplicación móvil. Lo permite llevar a cabo la gestión y monitoreo de la alarma residencial.

Se puede acceder a la información generada por el sistema de alarma residencial a través de la nube de Internet, por medio de la aplicación móvil donde se puede verificar el estado del sistema como: armado, desarmado, estado de energía, estado de zonas, etc.

Además, se pueden incorporar actuadores como una sirena, luces, bombas de agua, entre otros, que permiten disuadir acciones sospechosas o actos de vandalismo para mostrar que hay alguien en la residencia. Estos actuadores pueden ser controlados de forma remota mediante la aplicación móvil, que brinda la opción de encenderlos o apagarlos mediante un botón específico. El acceso al monitoreo y control pasa por el servicio MQTT.

También, el usuario dispondrá de un aplicativo móvil, de manera que, cuando este fuera del hogar el dispositivo móvil acceda a verificar el estado de la alarma a través de la red de área amplia (de sus siglas en inglés *Wide Area Network*, WAN), usando el servicio de Internet del dispositivo móvil del usuario.

Para gestionar y monitorear la alarma, activar o desactivar los actuadores y recibir las notificaciones en caso de que suceda un evento, la comunicación es de doble vía entre "*NODEMCU ALARMA RESIDENCIAL*" y el dispositivo móvil. Para esto dispone de una conexión a Internet a través de la red de área local. Como respaldo al sistema de comunicación del mismo, también

cuenta con un acceso a Internet a través del uso de una red de datos móviles ya que dispone de un chip de telefonía móvil donde el usuario puede elegir a su operador. Esta comunicación móvil se activa solamente cuando el sistema de comunicación principal (WiFi) no esté disponible. El detalle de estas comunicaciones inalámbricas corresponde al lado derecho del bloque "*NODEMCU* ALARMA RESIDENCIAL" de la figura 2.1.

Enlace entre la Alarma Residencial con la Alarma Comunitaria

En caso de existir algún evento como una intrusión o activación del sistema por medio del botón de pánico, el sistema 1 activará la sirena de la residencia y enviará una bandera al nodo inalámbrico "*CubeCeLL HTCC-AB02*" que cuenta con tecnología LoRaWAN. El nodo *CubeCeLL HTCC-AB02*, envía la bandera al *Gateway* para cargar la notificación en el servidor de red ChirpStack. Esta notificación luego es consumida desde el servidor de aplicación del segundo sistema denominado "alarma comunitaria" y que está basado en MQTT.

La base de datos mapea la bandera recibida para identificar al usuario que generó la alerta y si la generación fue de parte del botón de pánico de la residencia o desde algún aplicativo móvil de los residentes. Una vez identificado al usuario y el dispositivo, se construyen los mensajes para publicar notificaciones de tipo Push para todos los residentes del barrio. También, se procede a enviar una notificación al sistema de la alarma comunitaria para que suene la sirena del barrio. Estas alertas son almacenadas en un historial de eventos sobre las alertas generadas.

En el caso de que un propietario de una vivienda no desee implementar la alarma ni el sistema de monitoreo y gestión en su hogar, podrá poseer únicamente un botón de pánico inalámbrico (como se observa en el bloque de la casa, esquina inferior izquierda, figura 2.1), el cual tendrá baterías que le brindará autonomía dándole además la comodidad de ser portable. El botón tiene como función enviar alertas hacia el sistema de alarma comunitaria tal como se observa en la figura 2.1 por medio del Gateway y a su vez la data sea recibida en el bloque "*NODEMCU* ALARMA COMUNITARIA".

Sistema 2: Alarma Comunitaria

Como se mencionó, el segundo sistema definido como Alarma comunitaria, ubicado en la parte baja de la figura 2.1 recibe los datos de forma inalámbrica en su propio concentrador. Este sistema estará instalado en un sitio estratégico que pudiera ser la casa comunal, casa del presidente del barrio o garita de guardia; aquí se recogen los datos tanto de las casas que cuenten con el módulo de comunicación LoRaWAN, así como de los botones de pánico instalados en lugares estratégicos dentro de la urbanización, barrio, comunidad o en los hogares de las personas que optaron por esta modalidad.

En este sentido, cuando se genere una alerta inmediatamente llegará una notificación al móvil de cada propietario y a su vez se refleja dicha información en una pantalla informativa (bloque "pantalla informativa", figura 2.1 representada por línea color morado), la cual estará ubicada en un lugar estratégico. La alerta visualizará el nombre del propietario, esta información podrá ser vista por el resto de moradores facilitando el envío de ayuda y verificación del evento existente.

Dentro de la aplicación móvil también se tendrá un botón de pánico virtual el cual servirá también para activar la alarma comunitaria en caso de así requerirse. De esta manera, el usuario que cuenta con la alarma tendrá dos secciones, una para gestionar su propia alarma y otra para acceder al botón de pánico de la alarma comunitaria. Si un usuario sólo desea tener acceso a la alarma comunitaria entonces tendrá acceso solo a esta sección.

La comunicación entre el dispositivo móvil y el concentrador de la alarma comunitaria (segundo sistema) se realiza a través de la WAN al encontrarse fuera de la casa comunal, y por medio de la LAN al encontrarse dentro de la misma. Al igual que en cada hogar que posee el sistema de gestión de la alarma residencial, en el sistema de la alarma comunitaria también cuneta con respaldo de red vía acceso de telefonía móvil.

2.3. Decodificación de Alarma DSC

Para empezar, se necesita obtener los datos que se generan por el sistema de alarma residencial; en este caso se uso una alarma básica DSC modelo 585. Para realizar la gestión y monitoreo de forma remota se utilizó la librería "*DSC KEYBUS*" la cual se encuentra alojada en la plataforma "Github" y es de dominio publico [61].

Como primer punto se construyó un decodificador que consta de hardware y software, los mismos que combinados permiten conocer el estado de la alarma; el hardware se conecta directamente en la interfaz "*KEYBUS*" y se encargara de tomar los bytes que se generan por el sistema de alarma residencial y a su ves decodificarlos. Para esto se usó un divisor de voltaje tanto en el terminal "*Clock*" y "*Data*" para conseguir un voltaje apropiado para el microcontrolador; además se utilizó una placa de desarrollo "*NODEMCU*", la que se encarga del procesamiento de los datos y entrega el estado de la alarma. El esquema de conexión se puede observar en la figura 2.2.

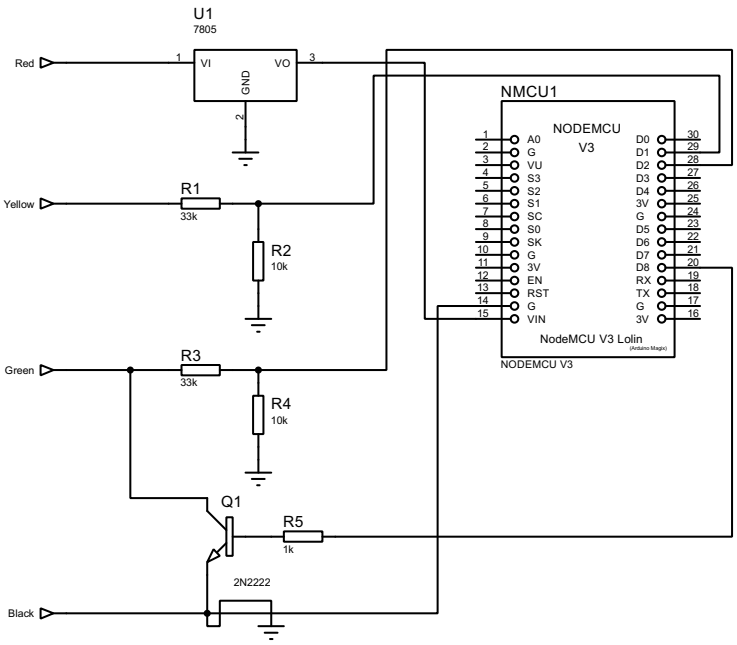

Figura 2.2: Esquema de conexión decodificador KEYBUS [Fuente Autores].

Con la ayuda de la librería "*DSC KEYBUS*" se desarrolla el programa que permite obtener información sobre el estado del sistema de alarma. Esto incluye el estado de las zonas, el estado de la energía y si está activada o desactivada.

Además, el programa permite el armado y desarmado remoto utilizando las contraseñas establecidas por el usuario. En la figura 2.3, se muestra el diagrama de flujo que representa el funcionamiento del decodificador que se construyó. Este diagrama detalla las distintas etapas que sigue el programa para obtener y procesar los datos del sistema de alarma. A través de este flujo lógico, se logró una efectiva decodificación de la información y su posterior representación en una interfaz serial, lo que permite realizar pruebas locales y garantizar el correcto funcionamiento del decodificador.

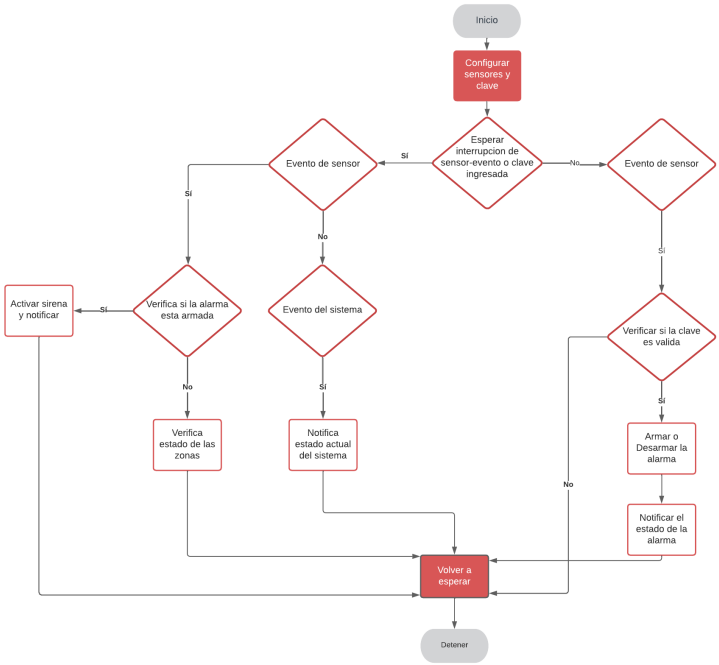

Figura 2.3: Diagrama de flujo de funcionamiento decodificador alarma DSC 585 [Fuente Autores].

El programa utiliza la decodificación de los datos provenientes del bus "*KEYBUS*" y los muestra en una interfaz serial, lo que permite verificar el funcionamiento del decodificador de manera local.

Una vez que el programa ha sido desarrollado, es crucial realizar pruebas exhaustivas para verificar su funcionamiento. Para ello, se montó el circuito en un *protoboard* y se procedió a comprobar que la decodificación se realizó correctamente, proporcionando información precisa sobre los eventos que se están generando en el sistema de alarma residencial.

Una vez que se confirmó que el programa funciona de manera adecuada

y satisface las necesidades, el siguiente paso es el diseño del PCB (de sus siglas en ingles *Printed Circuit Board*, PCB) para montar el circuito.

El diseño del PCB facilita la integración de los componentes electrónicos y proporciona una solución más profesional y duradera en comparación con el montaje en un protoboard. Además, reduce el riesgo de conexiones sueltas o errores de cableado, lo que mejora la confiabilidad del sistema en general. Al concluir el proceso de diseño del PCB, se obtiene un diseño final, tal como se muestra en la figura 2.4. Esta versión mejorada y optimizada del circuito garantiza una mejor eficiencia y un rendimiento óptimo en el sistema de alarma residencial.

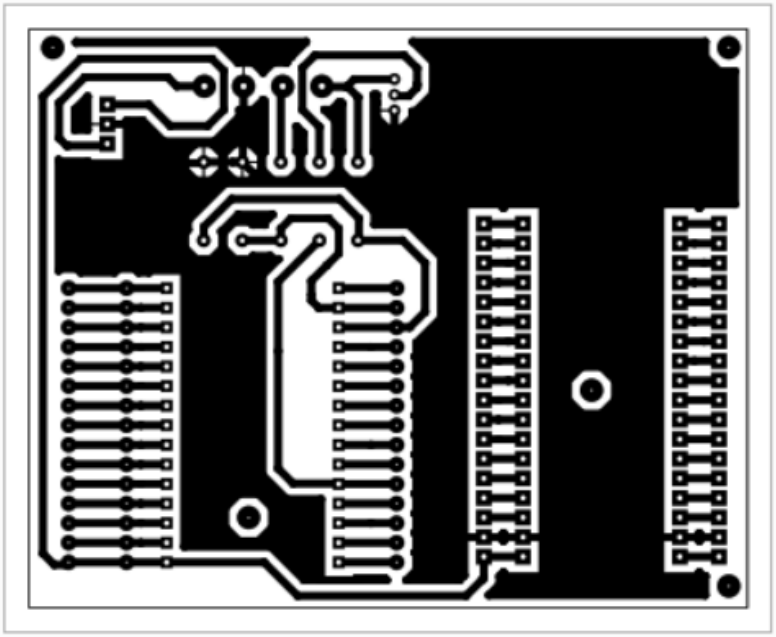

Figura 2.4: Diseño PCB decodificador alarma residencial [Fuente Autores].

Una vez completado los pasos anteriores, procedió a realizar las configuraciones necesarias en la alarma. Esto incluyó la configuración de la hora y fecha, atributos de las zonas, creación de usuarios y contraseñas, configuración de tiempos del sistema, entre otros. Puedes encontrar instrucciones detalladas sobre la programación en [14].

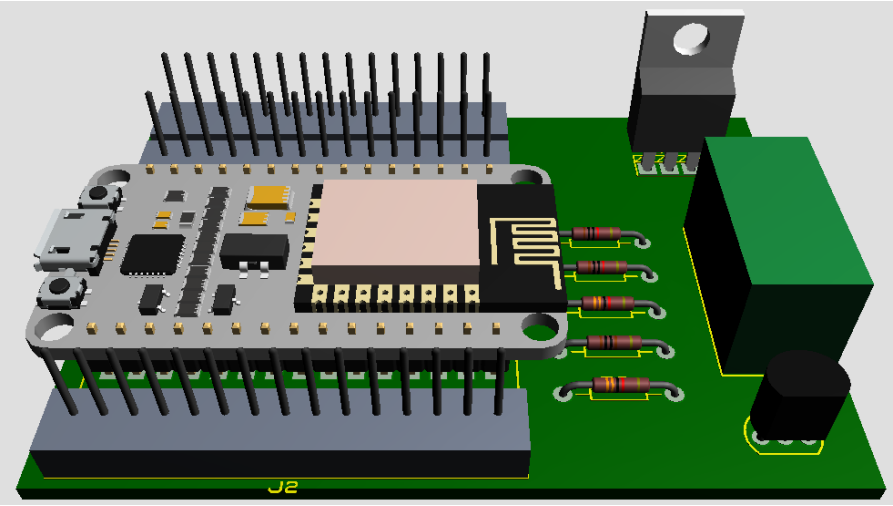

Figura 2.5: Diseño PCB alarma residencial 3D [Fuente Autores].

2.4. Implementación de servidor Eclipse Mosquitto

Para conocer el estado de la alarma de forma remota se utilizó un protocolo de comunicación, el cual envía esta información por medio de la nube, en este sentido se optó por el protocolo MQTT ya que es un protocolo ligero de mensajería diseñado para la comunicación entre dispositivos en redes de sensores y escenarios IoT, además de las características mencionadas en el capitulo 1.

Para la implementación de la plataforma, el broker Mosquitto es la elección ideal. Este software de código abierto cuenta con una amplia comunidad de desarrolladores y es gratuito, lo que lo hace altamente accesible y adaptable a las necesidades.

2.4.1. Alojamiento de Eclipse Mosquitto en Aws

Para configurar el broker Mosquitto, es necesario hacerlo en un servidor. Se eligió utilizar *Ubuntu 22.04*, como sistema operativo para el servidor. Para aprovechar las ventajas mencionadas en el 1, se optó por utilizar los servicios de AWS) en una instancia EC2. En la figura 2.6 se muestra la instancia EC2 seleccionada para montar el broker Mosquitto.

Una vez instalado el servidor, se procedió a la implementación de Mosquitto. Para ello, se estableció una conexión *SSH* al servidor a través del puerto 22. Esto permitió acceder y administrar el servidor de forma remota.

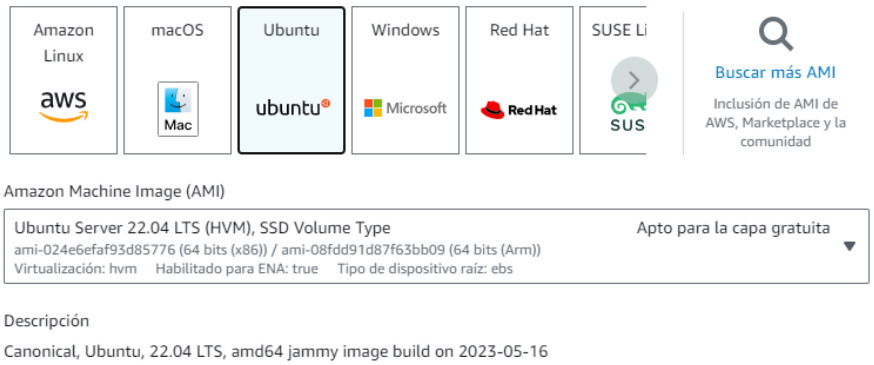

Figura 2.6: Instancia EC2 con Ubuntu Server 22.04 para Mosquitto [Fuente Autores].

La figura 2.7,muestra la conexión al servidor Ubuntu alojado en AWS mediante SSH utilizando CMD de Windows.

Para esta conexión, se establece la clave generada durante la configuración inicial de la instancia en AWS. Esta clave es única y la que permite el acceso y la administración de la instancia de forma segura.

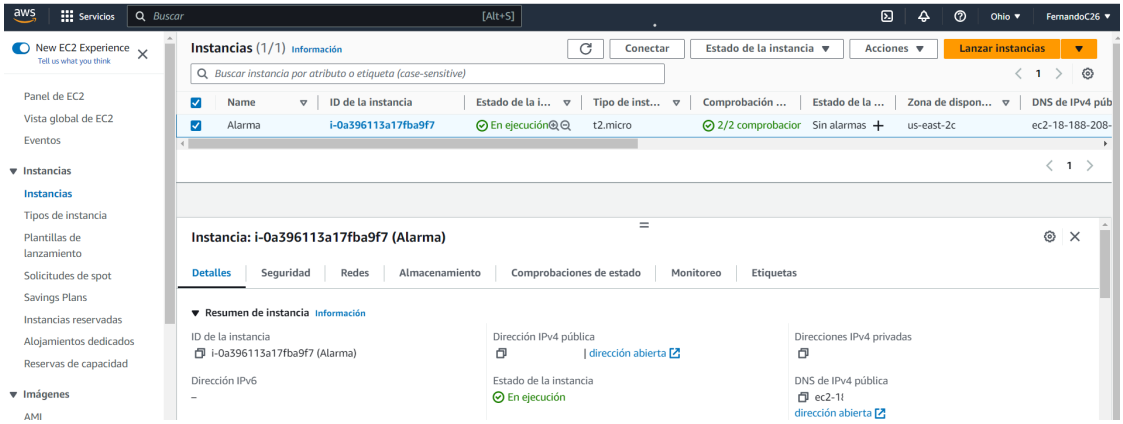

Figura 2.7: Instancia EC2 creada en AWS [Fuente Autores].

Para conectarse se da clic en el botón conectar y a su vez se selecciona la pestaña cliente SSH, como se puede observar en la figura 2.8. Además, se copia el comando del ejemplo en donde se toma por defecto la clave que se creo al inicio de la configuración de la instancia; este comando se lo usa en el CMD, pero primero es importante ubicarse en el directorio en donde se encuentra guardada la clave, como se puede observar en la figura 2.9. Una vez se ubicó en el directorio se procede a pegar el comando copiado y se confirma la conexión.

Cuando se establece la conexión SSH, como se observa en la Figura 2.10, se procede a instalar el broker Mosquitto siguiendo la

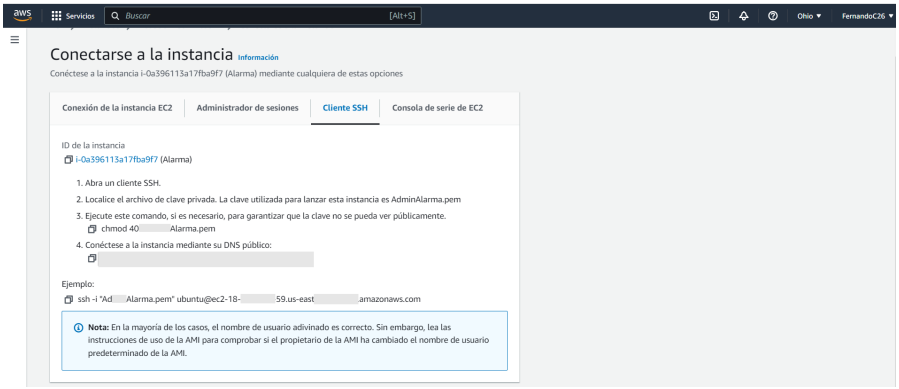

Figura 2.8: Conexión a la instancia cliente SSH [Fuente Autores].

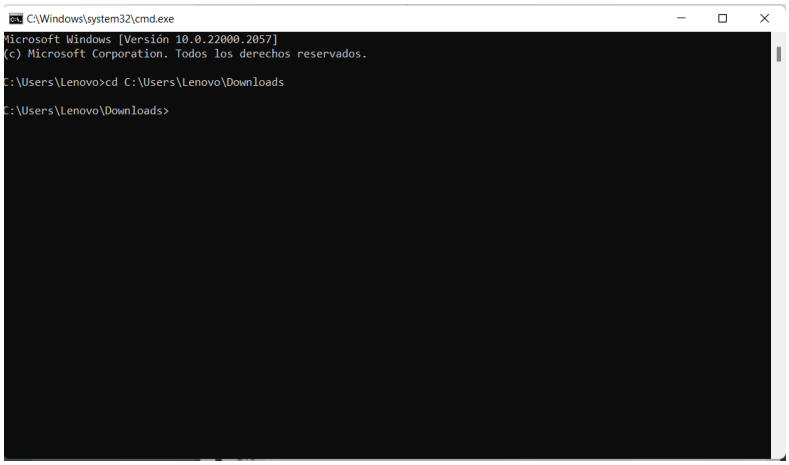

Figura 2.9: Ubicación en la ruta donde se guarda la contraseña para la instancia EC2 [Fuente Autores].

documentación disponible en `https://mosquitto.org/documentation/`. En esta documentación se encuentran los pasos detallados para instalar Mosquitto y las dependencias necesarias.

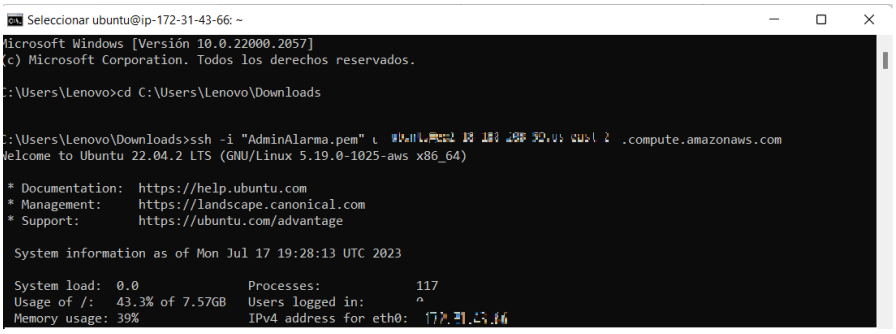

Figura 2.10: Conexión al servidor Ubuntu mediante ssh desde CMD de Windows [Fuente Autores].

2.4.2. Instalación de broker Mosquitto

Como primer punto se necesita actualizar los repositorios del sistema

```
sudo apt update
```

Se instala el paquete de Mosquitto, durante la esta instalación se agregaran los paquetes necesarios y Mosquitto se configura como un servicio del sistema.

```
sudo apt install mosquitto
```

Ya instalado se verificar el estado del servicio, con la finalidad de ver si se esta ejecutando de forma correcta o contiene errores.

```
sudo system clt status mosquitto
```

Si no existe problemas con el servicio se habilita el inicio automático de Mosquitto cuando arranca el sistema.

```
sudo system ctl enable mosquito
```

Se instala la herramienta de lineas de comandos de Mosquitto para publicar y suscribir mensaje MQTT y con consigo realizamos las configuraciones de seguridad. Para esto se crea un archivo de contraseñas que se usa para la autenticación.

```
sudo apt install mosquitto-clients
sudo mosquitto_passwd-c/etc/mosquitto/passwd<username>
```

Se remplaza "*<username>*" con el nombre de usuario que se va usar para la autenticación. Esto crea un archivo de contraseñas ubicado en "*/etc/mosquitto/password*", el mismo que solicita el ingreso de una contraseña para este usuario. Una vez creado el archivo de contraseñas, se habilita la autenticación en Mosquitto utilizando el archivo. Es necesario ir al archivo de configuración y realizar los cambios respectivos. Sin embargo esta configuración se deja para el final luego de crear conexión TLS.

Habilitación de conexión TLS en Mosquitto

Se genera un par de claves y certificados auto-firmados para la conexión *TLS* y se agrega al archivo de configuración la ruta de los mismos. Como primer paso ubicarse en el directorio donde se guarda los archivos de claves y certificados; consigo se genera una clave privada *(archivo.key)*. Este comando genera una clave privada *RSA* con cifrado AES-256 y se guarda en el archivo mosquitto.key. Durante este paso, se estableció la contraseña para la clave privada.

```
openssl genpkey-out mosquitto.key-algorithm RSA-aes256
```

A continuación, se genera una solicitud de firma de certificado (*CSR - Certificate Signing Request*) donde se incluye la clave pública. Este proceso pide información tales como el país, organización, ciudad, contacto etc.

```
openssl req-new-key mosquitto.key-out mosquitto.csr
```

Se firma el certificado usando la clave privada y el CSR auto-firmado; cuando se genere el certificado auto-firmado este se guarda como un archivo 'mosquitto.crt.

```
opensslx509-req-in mosquitto.csr-signkey mosquitto.key-out
mosquitto.crt
```

Para evitar que Mosquitto solicite ingresar la clave privada cada vez que se inicia, se puede eliminar la misma y así guardarla en un archivo "*mosquitto.key*". Durante este proceso se pide el ingreso de la contraseña que se estableció al crear la clave privada.

```
openssl rsa-in mosqutto.key-out mosquitto.key
```

Una vez concluido este proceso se obtuvo el par de claves y el certificado auto-firmado para la conexión TLS. Finalmente se agregan las rutas de estos archivos al archivo de configuración de mosquito quedando de la siguiente manera.

```
#Connection settings
port 1883
allow_anonymous false
password_file/etc/mosquitto/passwd
#TLS connection settings
listener 8883
cafile/etc/mosquitto/certs/mosquitto.crt
certfile/etc/mosquitto/certs/mosquitto.crt
keyfile/etc/mosquitto/certs/mosquitto.key
require_certificate false
```

Ya configurado y funcionando correctamente el broker. Ahora, es necesario complementar el programa de decodificación de la alarma para que, además de proporcionar el estado de la alarma de forma local, también publique estos estados a través de MQTT. De esta manera, se pueda visualizar la información en tiempo real mediante la aplicación móvil.

Con la implementación de la publicación MQTT, se puede monitorear y recibir notificaciones sobre el estado de la alarma desde cualquier lugar, lo que mejora significativamente la gestión y el control remoto de la alarma residencial. Esto brinda una solución más completa y versátil para mantener informados a los usuarios sobre el sistema de alarma y actuar de manera oportuna ante cualquier eventualidad.

2.4.3. Implementación del protocolo MQTT para decodificador de alarma DSC 585

Una vez que el broker Mosquitto está en funcionamiento, es fundamental incorporar el protocolo MQTT dentro del decodificador. De esta manera, además de proporcionar el estado de la alarma en el entorno local, puede enviar estos estados a través de la nube para que sean visualizados desde la aplicación móvil desarrollada para esta plataforma.

Para lograr esto, se creó los “topics” que son utilizados para representar

cada estado de la alarma.

A continuación, se detallan los "topics" relevantes para este propósito. Además, se puede visualizarlos en la figura 2.11

- **"PabloCochancela/alarma/casa/z1, z2, .., z8":** Se utiliza estos "topics" para publicar el estado de cada zona individual de la alarma. Cada zona tendrá su propio "topic".
- **"PabloCochancela/alarma/casa/Ready":** Este "topic" se utiliza para publicar el estado de disponibilidad de la alarma para ser armada. Si la alarma está lista para ser armada, se publica un mensaje que lo indique. Sin embargo, si alguna zona no está cerrada adecuadamente, el sistema envía un mensaje que señala que la alarma no puede ser armada hasta que todas las zonas estén cerradas correctamente.
- **"PabloCochancela/alarma/casa/energia":** Este "topic" se utiliza para publicar el estado de energía de la alarma, incluyendo información sobre si está conectada o desconectada de la red pública de suministro eléctrico. Además, el "topic" notifica si la alarma ha entrado en funcionamiento utilizando su batería de respaldo,en caso de que haya una interrupción del suministro eléctrico.
- **"PabloCochancela/alarma/casa/clave":** Este "topic" es utilizado para enviar la contraseña que permitirá el armado y desarmado de la alarma desde la aplicación móvil. Al publicar la contraseña en este "topic", proporciona de una forma segura y autorizada para activar o desactivar la alarma de manera remota mediante el aplicativo móvil. .
- **"PabloCochancela/casa/actuadores/rele1, rele2, ..., rele4":** Este "topic" es utilizado para controlar los actuadores que disuaden acciones sospechosas. Al publicar mensajes en este "topic", se activan o desactivan los actuadores necesarios para generar una respuesta disuasoria ante situaciones de alarma o eventos sospechosos. Los actuadores pueden incluyen sirenas, luces estroboscópicas, sistemas de alarma sonora o cualquier dispositivo que contribuya a aumentar la seguridad y disuadir a posibles intrusos o amenazas.

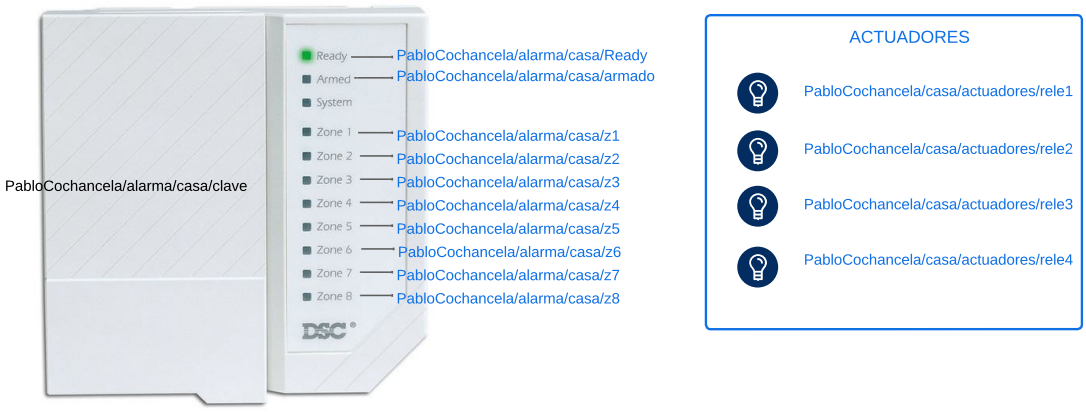

Figura 2.11: Topics usados para conocer estado alarma residencial DSC 585 [Fuente Autores].

Una vez definidos los "topics", el siguiente paso es establecer la conexión del decodificador a Internet. A través de la placa de desarrollo NodeMCU que permite la conexión a internet mediante Wi-Fi.

Se realiza la conexión hacia el broker a a través de la dirección IP pública proporcionada por el proveedor de servicios y las credenciales creadas para el broker.

Para verificar la conexión, se publicó y se suscribió a los "topics" desde la consola del broker Mosquitto. Esto permitió comprobar que la conexión y la configuración de MQTT funcionan correctamente.

Una vez que se realizaron las pruebas y verificación de conexión, el decodificador estará listo para publicar y suscribirse a los "topics" de forma automática. De esta manera, se puede enviar y recibir información del estado de la alarma a través de MQTT y visualizarla desde la aplicación móvil o cualquier otro cliente MQTT.

Esta conexión a Internet por parte del decodificador permite tener una gestión remota y en tiempo real del sistema de alarma, lo que mejora de manera satisfactoria la funcionalidad y el control del mismo.

2.5. Despliegue de la red LoRaWAn

La red LoRaWAN de este proyecto está compuesta por los siguientes elementos:

- **Servidor de Red:** Para gestionar y coordinar la comunicación en la red

LoRaWAN, se utiliza el servidor ChirpStack. Este servidor es responsable de administrar los dispositivos finales, las transmisiones de datos y la seguridad de la red.

- **Gateway:** El Gateway utilizado en esta red es el MileSight UG67. Este dispositivo actúa como un enlace entre los nodos finales y el servidor de red. Su función principal es recibir las transmisiones de datos de los nodos finales y reenviarlas al servidor para su procesamiento y almacenamiento.

- **Nodos Finales:** Se utiliza dos nodos CubeCell-GPS (HTCC-AB02S). Estos nodos finales son dispositivos equipados con capacidad de comunicación LoRaWAN y están diseñados para recopilar y enviar datos a la red.

Los datos de alerta provenientes de los dispositivos finales , son transmitidos utilizando modulación LoRa hacia el Gateway Milesight UG67. A su vez, el Gateway envía esta información al servidor de red ChirpStack utilizando el protocolo de comunicación TCP/IP. Desde allí, se establece una conexión utilizando el protocolo Message Queing Telemetry Transport (MQTT) hacia la aplicación móvil, todas las alertas generas podrán ser visualizadas a través de esta aplicación móvil por notificaciones push.

La figura 2.12 ilustra la topología de red propuesta, donde se describe de manera detallada la operación general de la implementación de la red.

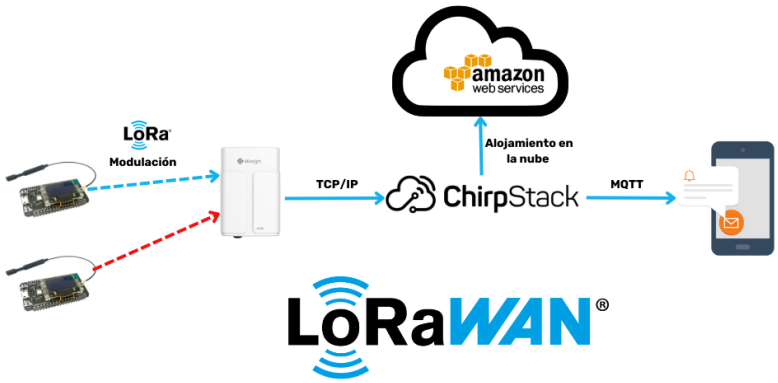

Figura 2.12: Red LoRaWAN Propuesta [Fuente Autores].

2.5.1. Configuración del servidor de Red

Para configurar el servidor ChirpStack, se recomienda utilizar un servicio de alojamiento en la nube en la plataforma Amazon Web Services (AWS). Este servicio es responsable de almacenar la información recibida por el Gateway. Además, proporciona informes detallados sobre cada dispositivo final conectado.

Alojamiento de ChirpStack en Amazon Web Server

La configuración del servidor ChirpStack se llevó a cabo en una instancia EC2 de AWS ejecutando el sistema operativo Ubuntu. Para lograr esto, se deben seguir los pasos recomendados en la documentación oficial de ChirpStack, disponible en `https://www.chirpstack.io/project/install/configuration/`. Una vez completada la instalación, se deben tener en cuenta las reglas de seguridad adecuada y la dirección IP estática generada por la instancia, para permitir el tráfico SSH a continuación los pasos de configuración del gateway.

2.5.2. Configuración del Gateway MileSight UG67

Para el correcto funcionamiento, es necesario llevar a cabo los siguientes pasos: en primer lugar, se debe poner en marcha el Gateway y acceder a su software (WEB EUI) a través de una red de área local. Es importante tener en cuenta el modo de funcionamiento de las antenas del Gateway. Si es necesario, se debe cambiar al modo de antena interna o asegurarse de tener las antenas conectadas. Si no se tiene en cuenta este aspecto, el sistema de transmisión LoRa podría sufrir daños.

La puesta en marcha del Gateway consta de conectar la alimentación correctamente del dispositivo. Para el funcionamiento correcto del Gateway se debe tener en consideración los siguientes elementos:

1. PoE conector
2. Switch o router

3. Power Adapter

La figura 2.13 muestra cómo se deben conectar los elementos mencionados anteriormente. Para comenzar, el Gateway se conecta al puerto (*Data+Power*) del PoE utilizando un cable Ethernet. A continuación, se debe utilizar otro cable Ethernet para conectar la salida del puerto de PoE (*Data*) a un router o switch con conexión a internet. De esta manera, se establece la conexión para acceder a la plataforma (*WEB EUI*) a través de este dispositivo router ya sea por cable Ethernet o Wi-fi.

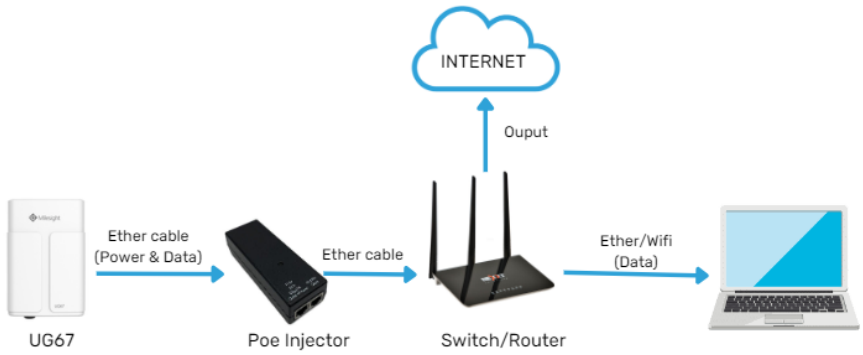

Figura 2.13: Conexión correcta Gateway MileSight UG67 [Fuente Autores].

Como primer paso en la configuración, se debe acceder a la plataforma web del Gateway a través de una conexión Wi-Fi utilizando los parámetros predeterminados de fábrica.

```
ETH IP Address: 192.168.23.150
Wi-Fi IP Address: 192.168.1.1
Wi-Fi SSID: Gateway_UG67
Username: admin
Password: password
```

Para configurar el acceso a Internet del Gateway,se cambia el estado del Port 1. Por defecto, este puerto viene configurado con una IP estática. En este caso, se recomienda seleccionar el estado DHCP para que el Gateway pueda obtener una dirección IP desde el router y así tener salida a Internet.

La figura 2.14 muestra donde se realiza esta configuración.

A continuación, se procede a configurar las frecuencias de trabajo,

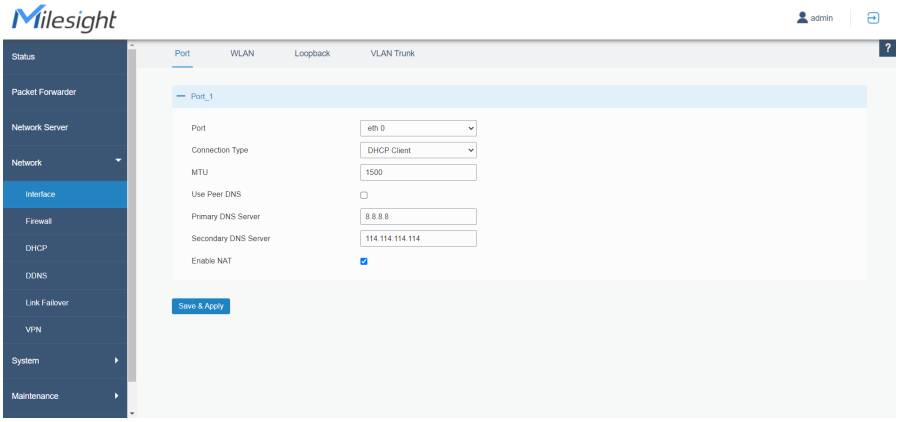

Figura 2.14: Configuración Port Gateway MileSight UG67 [Fuente Autores].

asegurando que coincidan con las seleccionadas previamente en el servidor ChirpStack y los dispositivos finales, como se muestra en la figura 2.21.

La frecuencia de trabajo del Gateway y su configuración correspondiente se presentan detalladamente en la figura 2.15.

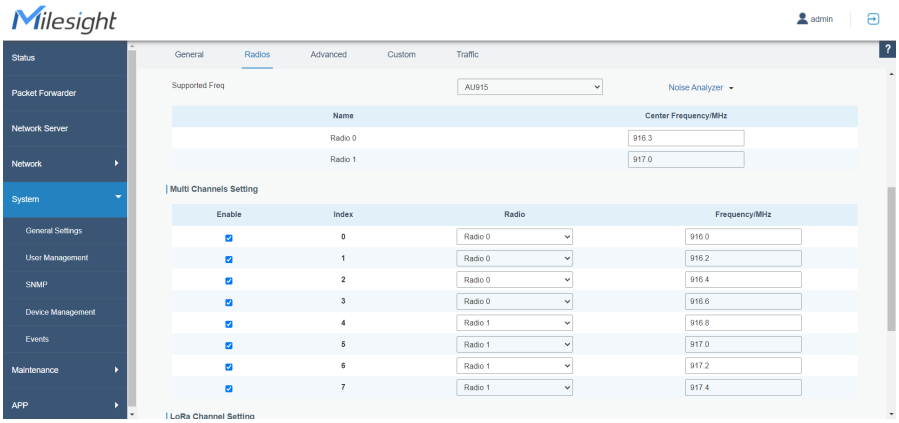

Figura 2.15: Configuración de frecuencias MileSight UG67 [Fuente Autores].

Como último paso, se configura el servidor al que se dirigen los paquetes. La figura 2.16 muestra el lugar donde se realiza esta configuración. Aquí se pueden establecer los parámetros necesarios para que los paquetes sean enviados correctamente al servidor designado.

La configuración de la IP del servidor se ilustra en la figura 2.17, al igual que en TTN, debe ser Semtech. En el campo "*Server Address*", se debe ingresar la dirección IP del servidor, y en el campo "Puerto", tanto para la dirección ascendente (*uplink*) como para la dirección descendente (*downlink*), se debe establecer el valor 1700.

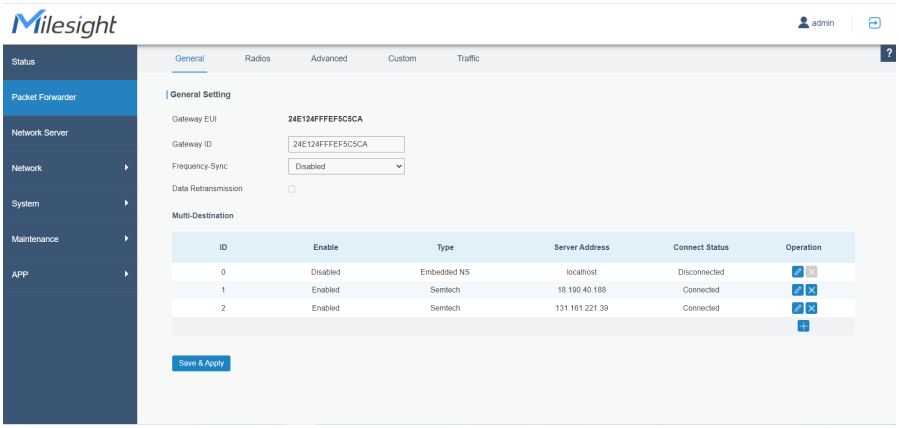

Figura 2.16: Configuración de servidor en MileSight UG67 [Fuente Autores].

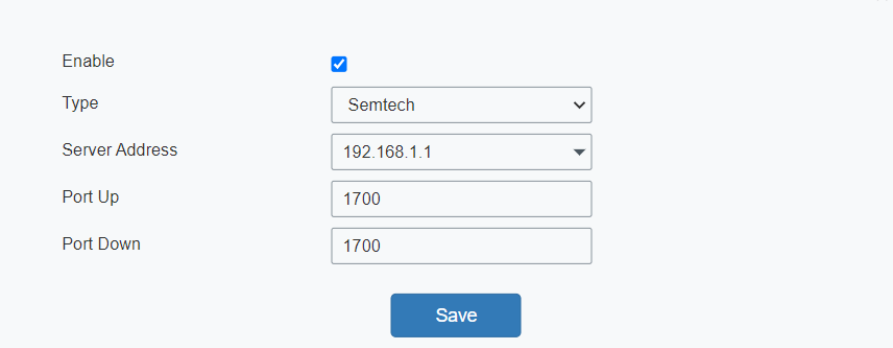

Figura 2.17: Configuración de IP de servidor [Fuente Autores].

2.5.3. Configuración de dispositivos finales

La configuración de los dispositivos finales se lleva a cabo a través del entorno de programación *Arduino IDE*. Este entorno proporciona una plataforma intuitiva para crear y cargar el código en la placa *Cubecell GPS-AB02S* mediante la conexión a un puerto en el ordenador.

Configuracion Cubecell GPS-6502S

Antes de comenzar con la configuración de los módulos *Cubecell GPS-6502S* para que se comuniquen con el Gateway, es importante tener en cuenta las siguientes consideraciones:

- El *Cubecell GPS-AB02S* cuenta con un transceptor LoRa compatible con LoRaWAN 1.0.2, lo que permite una comunicación inalámbrica de largo alcance con baja potencia y alta fiabilidad.
- Permite utilizar baterías de Lithium que tengan un voltaje entre 3.3 V - 4.2

V, se recomienda usar una de 3.7 V de Li-Ion.

- Es necesario realizar la conexión de la antena al trabajar con tecnología LoRa o LoRaWAN, ya que no hacerlo podría dañar el dispositivo. La figura 2.18 ilustra la ubicación de la conexión de la antena.

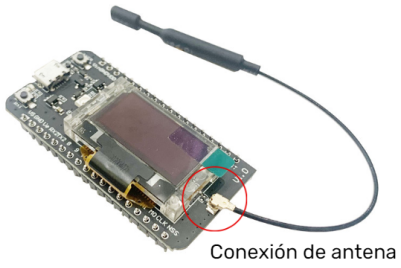

Figura 2.18: Conexión de Antena Cubecell GPS [Fuente Autores].

Para la configuración e integración al servidor Chirpstack se basa en un ejemplo del repositorio Github como parte de las librerias de CubeCell en Arduino IDE: `https://github.com/HelTecAutomation/CubeCell-Arduino/tree/master/libraries/LoRaWanMinimal/examples/SendReceive`.

El algoritmo permite registrar el dispositivo final colocando las claves correctas para el registro OTAA. Las claves que se deben colocar se visualizan en las figuras 2.28, 2.29.

```
#Set these OTAA parameters to match your app/node
devEui[] = { 0x5e, 0x34, 0xdb, 0xa6, 0x7b, 0x95, 0x87, 0xa6 };
appEui[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
appKey[] = { 0xef, 0x54, 0xf0, 0x08, 0x34, 0x6c, 0xac, 0xe1,
0xbd, 0xe1, 0x36,0xfa, 0xc2, 0xff, 0x3f, 0x2b };
userChannelsMask[6]={ 0x00FF,0x0000,0x0000,0x0000,0x0000,0x0000 };
```

Por seguridad se cambian los parámetros mencionados con anterioridad.

En el contexto de la programación, es necesario utilizar los parámetros específicos de LoRaWAN correspondientes a la región en la que se despliega la red. Dado que este trabajo se lleva a cabo en Ecuador, se proporcionarán los valores apropiados de frecuencia para configurar los dispositivos dentro de dicho país. La figura 2.19 muestra cada uno de los valores configurados en el menú de herramientas.

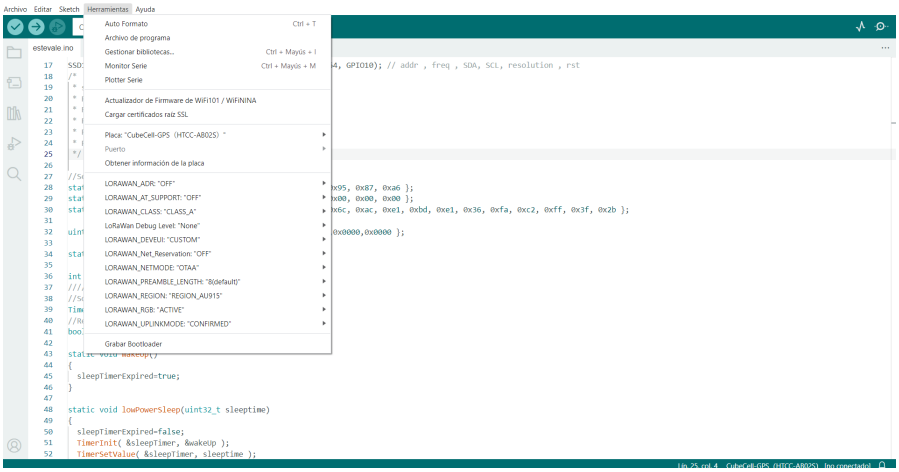

Figura 2.19: Parámetros de configuración Cubecell Gps [Fuente Autores].

2.5.4. Registro de dispositivos en ChirpStack

Para registrar cada uno de los dispositivos en el servidor ChirpStack, es necesario seguir un proceso que comienza con la creación del servidor de red. En este caso particular, dado que el servidor se encuentra alojado en AWS, se debe proporcionar la dirección IP asignada y el puerto correspondiente. Estos detalles se encuentran visualmente representados en la figura 2.20, brindando una guía clara para llevar a cabo esta configuración inicial.

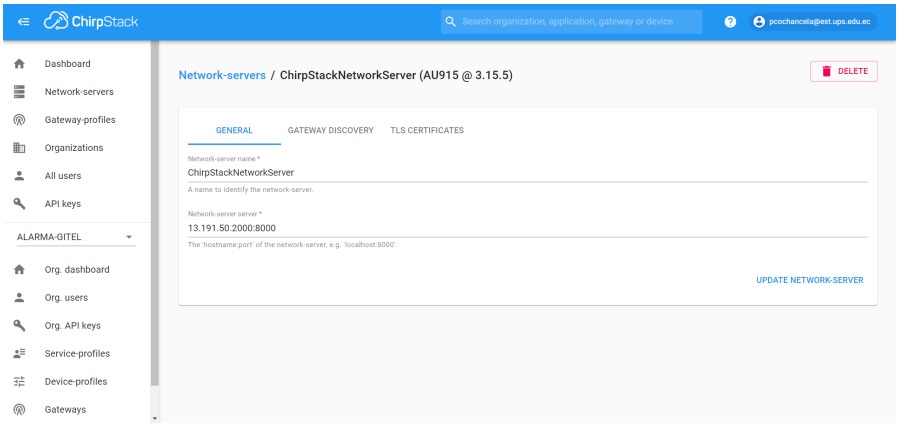

Figura 2.20: Configuración de servidor de red ChirpStack [Fuente Autores].

Registro de Gateway MileSight UG67

Para registrar un gateway en el servidor ChirpStack, el primer paso consiste en crear un perfil que incluya información como el nombre del gateway, el intervalo de muestreo de datos y los canales que se utilizarán. A continuación,

la figura 2.21 muestra cómo se realiza esta configuración, brindando una representación visual de los pasos necesarios para completar el registro del gateway en el servidor ChirpStack.

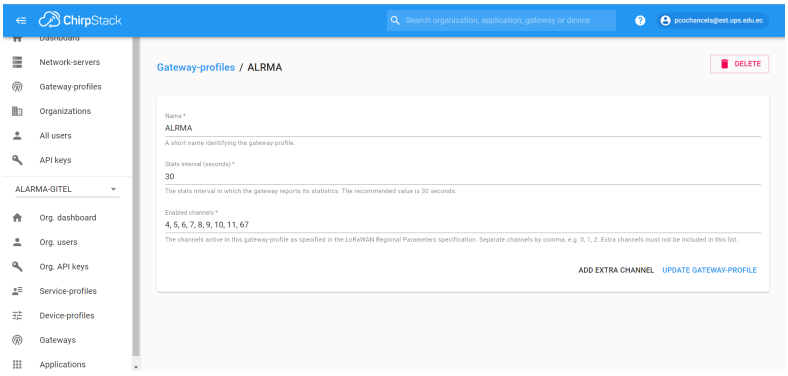

Figura 2.21: Configuración de Gateway profile en ChirpStack [Fuente Autores].

El siguiente paso consiste en configurar el perfil de servicio para el gateway. En esta configuración, se asigna un nombre al perfil y se habilita la generación de metadatos, así como también se puede incluir información de geolocalización y establecer la tasa de datos máxima según las necesidades específicas de la aplicación.

La figura 2.22 ilustra detalladamente esta configuración, brindando una referencia visual para llevar a cabo el proceso de manera efectiva.

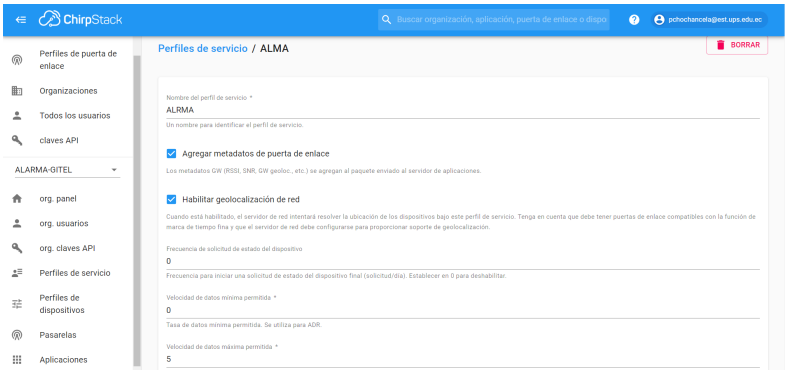

Figura 2.22: Configuración de Service profile en ChirpStack [Fuente Autores].

El paso para completar el registro del gateway consiste en crear una nueva puerta de enlace, donde se completan varios parámetros importantes. Estos incluyen: el nombre de la puerta de enlace, una descripción relevante, el ID del gateway, la vinculación con los servicios previamente configurados, así como la altitud y la geolocalización donde se encuentra ubicada. Todo lo mencionado se muestra en la figura 2.23.

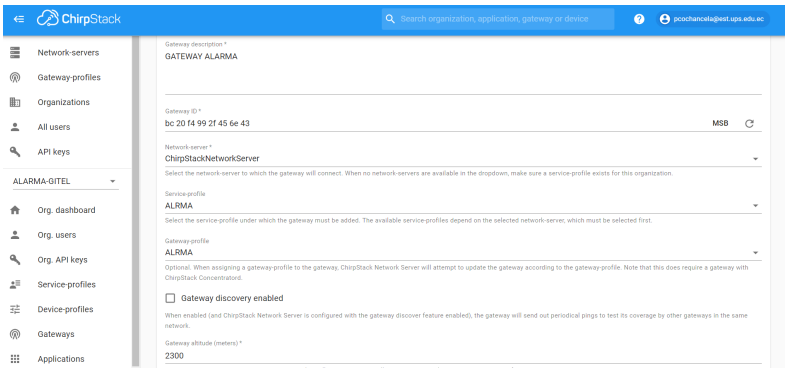

Figura 2.23: Configuración de Gateway ChirpStack [Fuente Autores].

Para verificar el correcto funcionamiento de las configuraciones realizadas, se lleva a cabo una breve revisión en la sección de Dashboard. Aquí, el gateway registrado se reflejará con un indicador de color verde, tal como se muestra en la figura 2.24. Este indicador visual confirma que las configuraciones se han aplicado correctamente y que el gateway está funcionando adecuadamente dentro de la red.

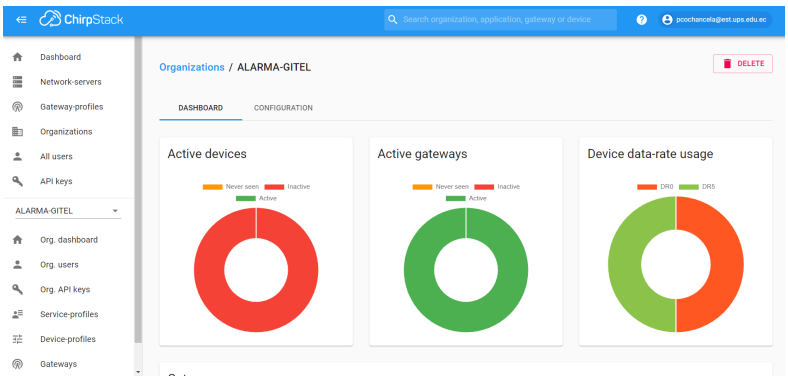

Figura 2.24: Activación correcta de gateway en ChirpStack [Fuente Autores].

Registro de dispositivos finales

Para llevar a cabo el registro de dispositivos finales, es necesario considerar el tipo y la clase de dispositivo mencionados en el capítulo 1. En este caso, se utilizará el método de registro OTAA y se seleccionará la clase A. El primer paso consiste en crear un perfil de dispositivo, donde se deben completar los campos requeridos según se muestra en la figura 2.25.

Para indicar el registro por aire OTAA, se habilita lo siguiente ilustrado en la figura 2.26.

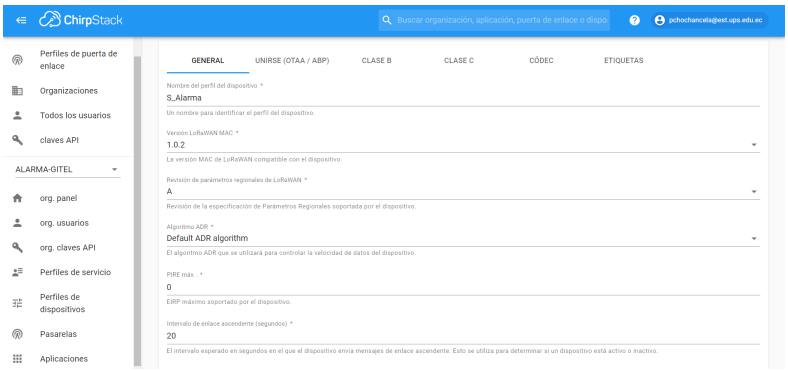

Figura 2.25: Perfil de dispositivo final en ChirpStack [Fuente Autores].

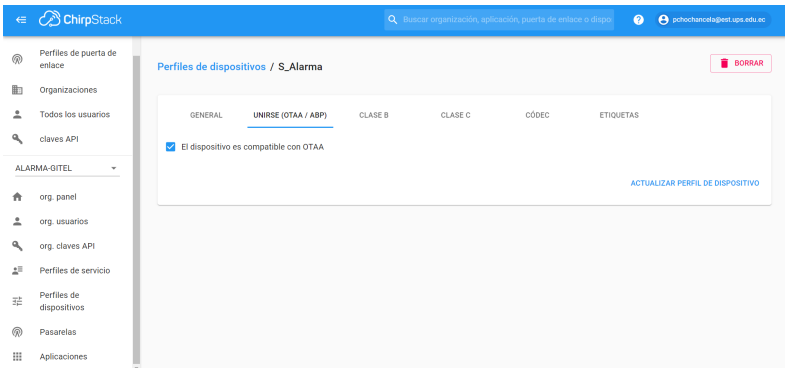

Figura 2.26: Conexión OTAA en ChirpStack [Fuente Autores].

El siguiente paso es crear una aplicación donde todos los dispositivos tenga la misma configuración a través del perfil de dispositivo creado con anterioridad y a su ves mismo verificar los datos recibidos, la figura 2.27 muestra la configuración de la aplicación de estos dispositivos.

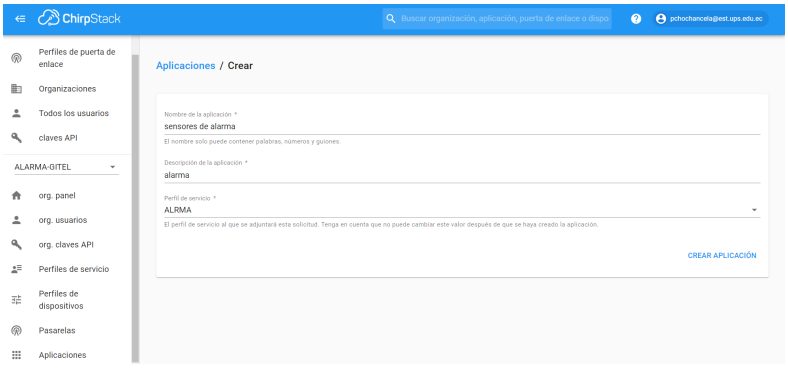

Figura 2.27: Creación de aplicación en ChirpStack [Fuente Autores].

Al momento de tener creado la aplicación, el siguiente paso es crear los dispositivos finales generando las claves(Application key,Device EUI) para su registro OTAA como se muestra en las figuras 2.28, 2.29.

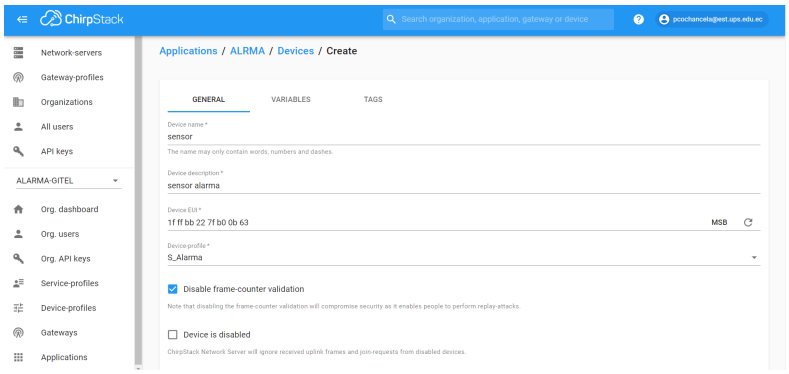

Figura 2.28: Creación de dispositivos finales en ChirpStack [Fuente Autores].

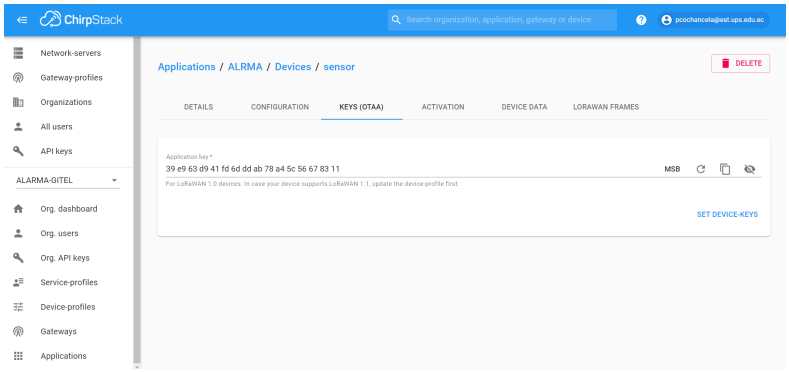

Figura 2.29: Creación de dispositivos finales en ChirpStack [Fuente Autores].

2.5.5. Configuración de servidor ChirpStack para integración MQTT

En esta sección, se utiliza la implementación del broker Mosquitto descrito en la sección 2.4.2. Para ello, se configura el servidor de aplicaciones para que utilice un broker externo. Para integrar MQTT a la aplicación, se debe dirigir al archivo de configuración.

```
sudo nano /etc/chirpstack-application-server/chirpstack-application
-server.toml
```

Como primer punto, se asegura de que MQTT se encuentra habilitado. se busca la sección correspondiente en el archivo de configuración y se verifica que la opción “enabled” esté configurada como “true”.

A continuación, se debe seleccionar el formato en el que se publican los mensajes MQTT. En este caso, se configura el formato como JSON v3 para una mejor organización y estructura de los datos. Esta sección de configuración,

también proporciona los datos de la dirección IP pública del broker MQTT, así como las credenciales de usuario y contraseña para establecer la comunicación con el broker. Esto permite una conexión segura y autentica.

Además, se asegura de pasar el certificado CA (de sus siglas en ingles *Certificate Authority*, CA) necesario para establecer una conexión segura por TLS. Este certificado CA es el mismo que generamos previamente en el broker Mosquitto durante la configuración de seguridad.

Al utilizar este certificado, se asegura que las comunicaciones entre el cliente y el broker MQTT se realicen de manera encriptada y confiable. Con todas las configuraciones realizadas en el archivo de configuración, la aplicación estará lista para utilizar MQTT como backend para la comunicación con el broker Mosquitto. El archivo de configuración se observa en la figura 2.30.

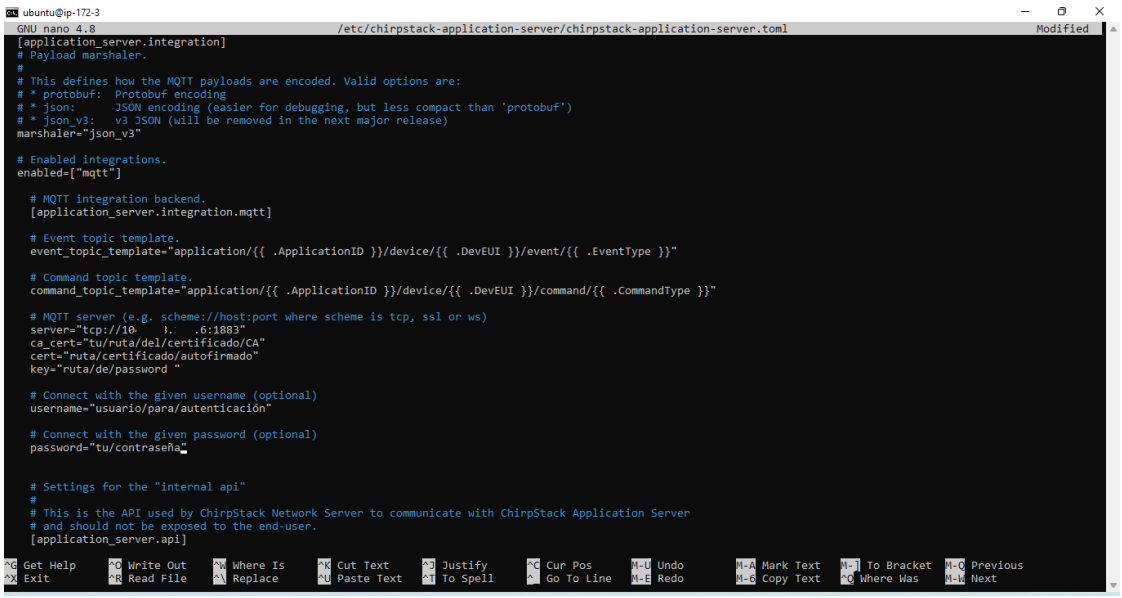

Figura 2.30: Archivo de configuración del servidor aplicaciones ChirpStack [Fuente Autores].

2.6. Desarrollo de aplicativo móvil

Esta sección se enfoca en planificar el funcionamiento y desarrollo de la aplicación móvil destinada a la gestión y monitoreo de una alarma residencial, la cual está conectada a un sistema de alarma comunitario. el objetivo es diseñar una aplicación que satisfaga todas las necesidades del proyecto de manera efectiva. El proceso inicia con el maquetado de la aplicación, como se observa

en la figura 2.31 donde se establece la estructura y el flujo de operación del sistema de gestión y monitoreo de la alarma residencial y sistema de alarma comunitaria.

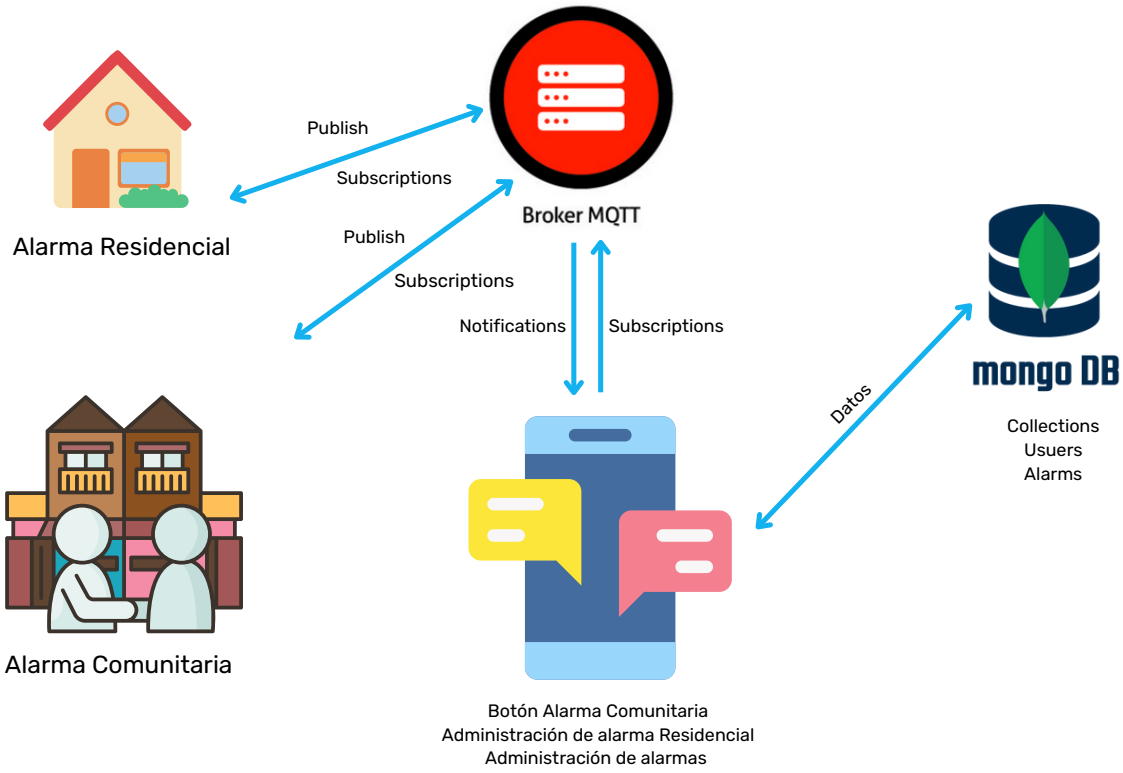

Figura 2.31: Diagrama de aplicación Móvil [Fuente Autores].

El objetivo es crear una experiencia amigable para las personas interesadas en la plataforma. Además, se busca mejorar la seguridad y proporcionar un sentido de comunidad al integrar la aplicación con el sistema de alarma comunitaria. Para lograr esto, se autoriza el envío de alertas en caso de eventos relevantes, lo que permite que los usuarios se mantengan informados y puedan tomar medidas adecuadas en tiempo real.

El maquetado de la aplicación se centrará en garantizar una experiencia fluida y sencilla, brindando a los usuarios el control y la tranquilidad que buscan en la gestión y monitoreo de su alarma residencial.

El primer requisito es implementar un sistema de autenticación en la aplicación para validar a los usuarios. Para ello, es necesario crear una base de datos que contenga los diferentes atributos requeridos al momento de crear un usuario .

2.6.1. Tipos de usuarios

Se han identificado tres tipos de usuarios con distintos privilegios y funciones:

- **Usuario 1: Administrador** Es el usuario encargado de administrar el sistema de alarma comunitario; tiene la capacidad de crear usuarios residenciales y usuarios del sistema de alarma comunitaria y es el único usuario autorizado para desactivar la alarma comunitaria en caso de ser activada, garantizando la seguridad del sistema.

- **Usuario 2: Residencial con Alarma Comunitaria** Son aquellos que desean gestionar su alarma residencial y tienen acceso al sistema de alarma comunitaria. Este tipo de usuarios tienen la capacidad de monitorear en tiempo real el estado de su sistema de alarma residencial, lo que les permite estar informados sobre cualquier evento relevante. Además, cuentan con la posibilidad de activar actuadores, como luces y sirenas, lo que les brinda un mayor control sobre la seguridad de su hogar. También tienen la opción de activar o desactivar su alarma residencial según sus necesidades. Disponen de una interfaz para activar el sistema de alarma comunitaria en caso de ser necesario.

- **Usuario 3: Alarma Comunitaria** Son los usuarios que únicamente cuenta con el sistema de alarma comunitaria en donde podrán activar el sistema en caso de así requerirlo.

El sistema de autenticación se encargará de identificar y otorgar los privilegios correspondientes a cada tipo de usuario, asegurando así que solo las personas autorizadas puedan acceder a las funcionalidades y datos pertinentes.

La base de datos contendrá la información necesaria para administrar los usuarios y sus permisos, brindando un sistema seguro y confiable para la aplicación de gestión y monitoreo de la alarma residencial y comunitaria.

Una vez definidos los diferentes tipos de usuarios, se crea la base de datos, se asegura de incluir todos los atributos necesarios para garantizar un correcto funcionamiento del sistema. Los atributos que se establecen son los

siguientes como se observa en la figura 2.32; cabe mencionar que el atributo "type" que se observa en la figura 2.32 corresponde a los privilegios que tendrá el usuario que se esta creando como se explico en el apartado 2.6.1.


```
"email": "pcochancela@hotmail.com"
"first-name": "Pablo",
"last-name": "Cochancela",
"password": "1234",
"type": 1,
"zona1": "zona1",
"zona2": "zona2",
"zona3": "zona3",
"zona4": "zona4",
"zona5": "zona5",
"zona6": "zona6",
"zona7": "zona7",
"zona8": "zona8",
"code": 1,
"etiqueta": "PabloCochancela",
"actuador1": "actuador1",
"actuador2": "actuador2",
"actuador3": "actuador3",
"actuador4": "actuador4"
```


Figura 2.32: Atributos requeridos para la base de datos [Fuente Autores].

Una vez que se hayan creado los distintos usuarios con sus correspondientes atributos, cada usuario podrá las interfaces permitidas según el tipo de usuario asignado durante su creación.

2.6.2. Asignación de interfaces de la aplicación móvil según tipo de usuario

A continuación, se presenta la propuesta de las distintas interfaces asignadas para cada tipo de usuario. Para el "Usuario 1: Administrador", se proporcionará una interfaz completa que permite gestionar y controlar todos los aspectos del sistema de alarma comunitario.

Esta interfaz brinda la capacidad de agregar, editar o eliminar usuarios residenciales y usuarios del sistema de alarma comunitaria como se observa en la figura 2.33. También se incluye la opción de apagar la alarma comunitaria en

caso necesario. Además, es posible revisar el historial de activación de la alarma comunitaria, que detalla información sobre el usuario que activó el sistema, la fecha y hora de activación.

Adicionalmente, se puede acceder a un mapa para visualizar la ubicación exacta desde donde se activó la alerta.

Figura 2.33: Interfaces propuestas para la creación, visualización, modificación, eliminación de usuarios e historial de alarmas activadas asignadas al usuario administrador [Fuente Autores].

Para el "Usuario 2: Residencial con Alarma Comunitaria", se otorga una interfaz específica que le permite monitorear en tiempo real el estado de su sistema de alarma residencial. Además, tiene la opción de activar actuadores, como luces y sirenas como se puede observar en 2.34, y puede activar o desactivar su alarma residencial mediante claves agregadas al sistema de alarma residencial según sus necesidades como se ve en la figura 2.35.

También cuenta con acceso a la interfaz de alarma comunitaria, lo que le permite activar dicho sistema en caso de requerirlo.

Por último, para el "Usuario 3: Alarma Comunitaria", proporciona una interfaz que le permite activar únicamente el sistema de alarma comunitaria como se observa en la figura 2.36, en caso de ser necesario.

Esta interfaz simple y específica se adapta a las necesidades de este tipo de usuario, brindando una experiencia de uso clara y concisa.

Con esta asignación de interfaces, aseguramos que cada usuario tenga acceso a las funcionalidades adecuadas para su tipo, lo que garantiza una

Figura 2.34: Interfaces propuestas para el manejo de alarma residencial y pedido de ayuda alarma comunitaria, administración del estado del sistema residencial, encendido y apagado de actuadores [Fuente Autores].

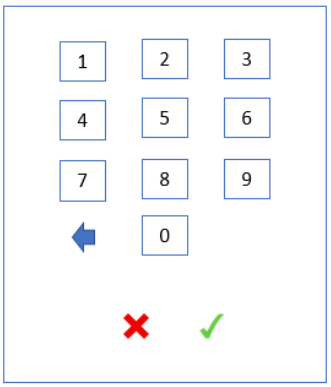

Figura 2.35: Interfaz propuesta para el envió de clave alarma residencial [Fuente Autores].

Figura 2.36: Interfaz propuesta para usuarios de alarma comunitaria [Fuente Autores].

experiencia personalizada y segura para todos los usuarios de la aplicación.

2.6.3. Aspectos importantes para el desarrollo del aplicativo móvil en Flutter

Se presentan distintos aspectos importantes en Flutter que permiten el desarrollo del aplicativo móvil para la gestión y monitoreo de una alarma residencial integrada a un sistema de alarma comunitario.

Conexión MQTT

Para publicar y recibir "topics" se utiliza una conexión MQTT la cual se muestra en la figura 2.37. Se coloca la IP del broker y las credenciales de autenticación. En este caso, por seguridad no se están usando los datos reales.

```
36  void _setupMqtt() async {
37    client = MqttServerClient('18.122.xxx.156', 'client1');
38    client!.port = 1883; // MQTT default port
39    client!.keepAlivePeriod = 30;
40
41    final MqttConnectMessage connectMessage = MqttConnectMessage()
42        .withClientIdentifier('client1')
43        .startClean() // Start a clean session
44        .keepAliveFor(30) // Keep alive interval
45        .withWillQos(MqttQos.atLeastOnce)
46
47    client!.connectionMessage = connectMessage;
48
49    try {
50      await client!.connect();
51      print('Connected to broker');
52    } catch (e) {
53      print('Connection failed: $e');
54    }
55  }
56
```

Figura 2.37: Conexion broker MQTT Mosquitto desde Flutter[Fuente Autores].

Suscripción a topics

Para comprobar el estado de la alarma mediante la aplicación, se lleva a cabo una suscripción a los "topics" correspondientes, tal como se representa en la figura 2.38.

Los siguientes "topics" que se muestran en la figura 2.39 permiten accionar la alarma comunitaria identificando el usuario que acciono la misma.

Figura 2.38: Suscripción Topics para recibir estado alarmas [Fuente Autores].

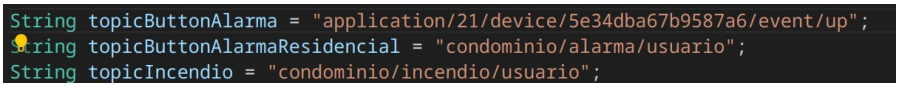

Figura 2.39: Topics usados para alarmas [Fuente Autores].

Mapeado de datos del servidor ChirpStack

Se lleva a cabo un proceso de mapeo de datos, como se ilustra en la figura 2.40, con el propósito de adquirir la información necesaria acerca de la ubicación y la identificación del cliente. Estos datos mapeados se transmiten mediante el protocolo MQTT desde el servidor Chirpstack.

Figura 2.40: Barrido de mensaje MQTT de alarma residencial recibido desde servidor de aplicación ChirpStack [Fuente Autores].

Generación de notificaciones Push

Para generar notificaciones push, se emplea el siguiente código, detallado en la figura 2.41, el cual posibilita la identificación de los datos del usuario que ha enviado la alerta de la alarma comunitaria.

```
Future<void> mostrarNotificationMessage(String message) async {
  const AndroidNotificationDetails androidNotificacions =
      AndroidNotificationDetails(
    "alarma2",
    "alarma2",
    importance: Importance.max,
    priority: Priority.high,
    icon: String.fromEnvironment("logo.jpg"),
  ); // AndroidNotificationDetails

  const NotificationDetails notificationDetails = NotificationDetails(
    android: androidNotificacions,
  );

  await flutterLocalNotificationsPlugin.show(
    1,
    "Alarma",
    message,
    notificationDetails,
  );
```

Figura 2.41: Generar notificaciones tipo push en aplicativo móvil [Fuente Autores].

2.6.4. Diagrama de clases de aplicación móvil

El diagrama de Lenguaje Unificado de Modelado (de sus siglas en ingles *Unified Modeling Language*, UML) de la aplicación móvil es una representación visual esencial que brinda una comprensión de cómo está construida la aplicación y cómo se interconectan sus diferentes componentes y módulos, como se ilustra en la figura 2.42.

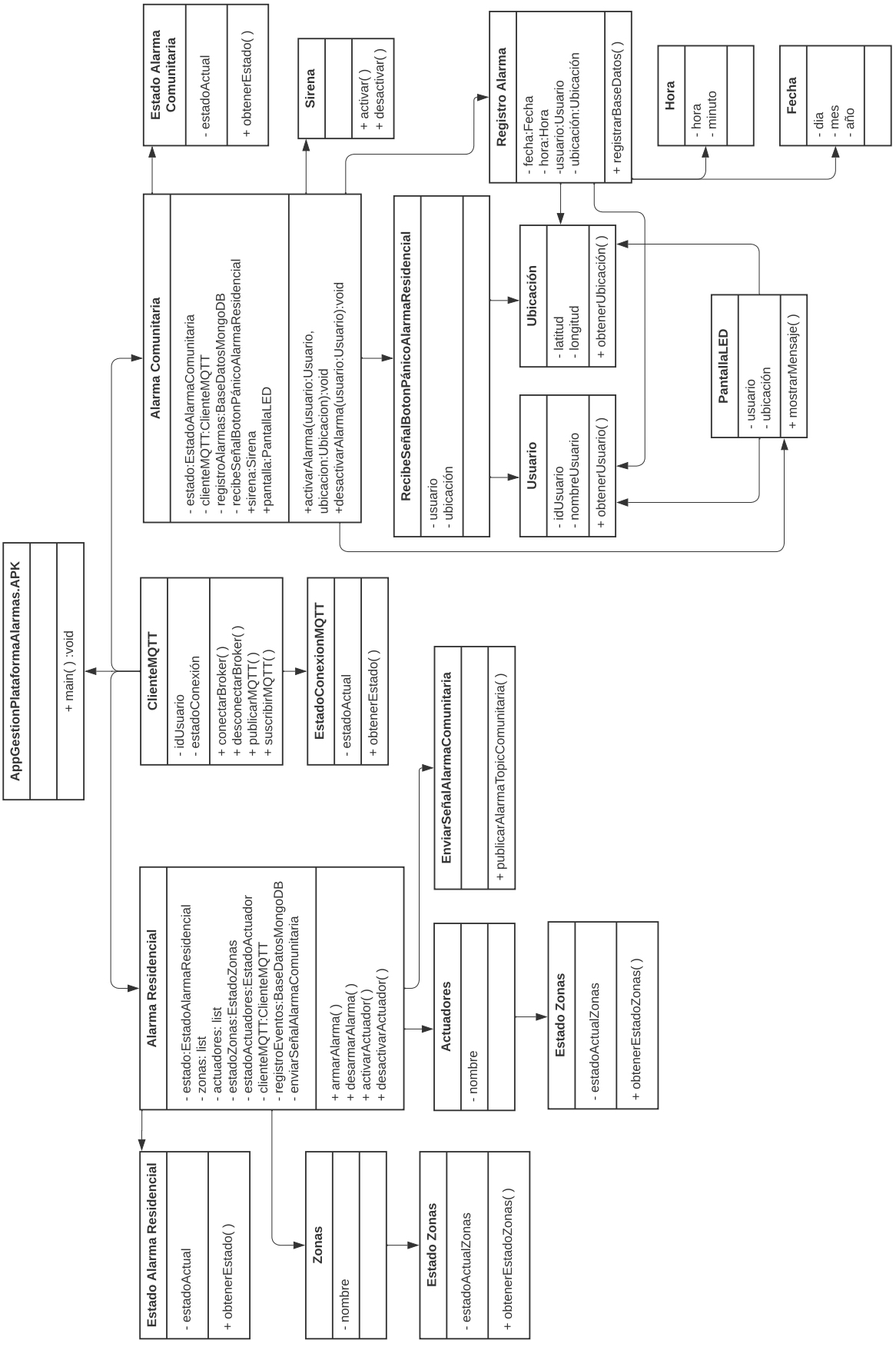

Figura 2.42: Diagrama de clases UML de la aplicación móvil de la plataforma diseñada [Fuente Autores].

Capítulo 3

Pruebas y Análisis de Resultados

En esta sección, se presentan los resultados de las pruebas realizadas para el presente trabajo. El desarrollo del proyecto se llevó a cabo utilizando una maqueta que permite simular una urbanización o vecindario. Además, se realizaron pruebas en el espacio libre para obtener resultados del tiempo y cobertura al momento de tener un aviso de intrusión.

3.1. Pruebas de funcionamiento de la plataforma sobre maqueta

Para la puesta en marcha del proyecto se diseño una maqueta que representa un encantador vecindario moderno que incluye dos viviendas, una caseta de guardia, áreas de entretenimiento, como canchas de baloncesto y fútbol. Además, espacios verdes que cuentan con iluminación. Por ultimo, se instalo diversos actuadores (luces, sirena, pantalla) que permiten verificar el funcionamiento de la plataforma. Todos los elementos mencionados se aprecian claramente en la figura 3.1.

A continuación, se presentan en las siguientes secciones las funcionalidad de cada parte que conforma la plataforma de gestión y monitoreo, las cuales son: alarma comunitaria y alarma residencial integrada al sistema de alarma comunitaria.

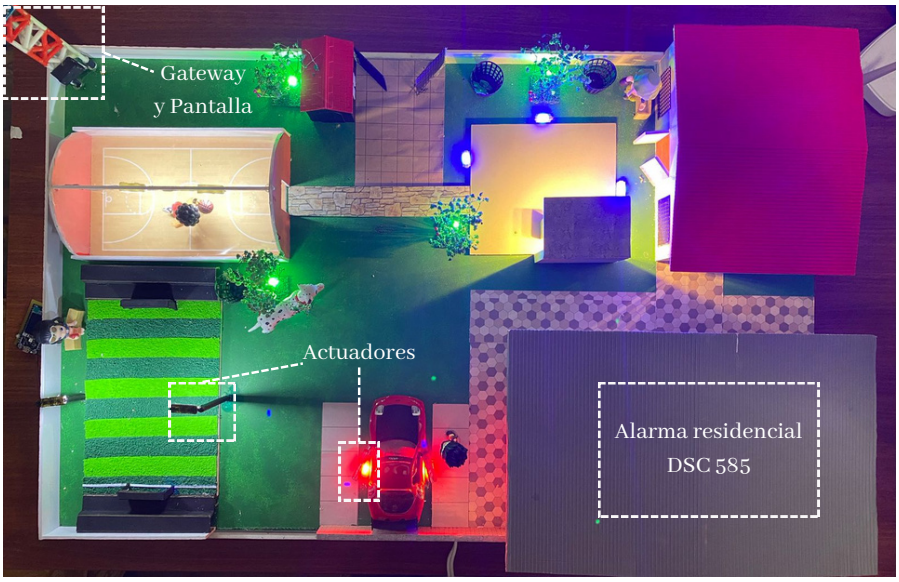

Figura 3.1: Vista superior de la de Maqueta [Fuente Autores].

3.1.1. Funcionamiento de Alarma Comunitaria con botón portable

El funcionamiento de la alarma comunitaria inicia al presionar el botón de pánico, el cual se muestra en la figura 3.2.

Figura 3.2: Botón de pánico activado [Fuente Autores].

Esto desencadena la alerta correspondiente a través de la red LoRaWAN y protocolo MQTT hacia el aplicativo móvil a través de una notificación push. La figura 3.3 muestra el registro del mensaje en el servidor ChirpStack y su correspondiente visualización en el aplicativo móvil.

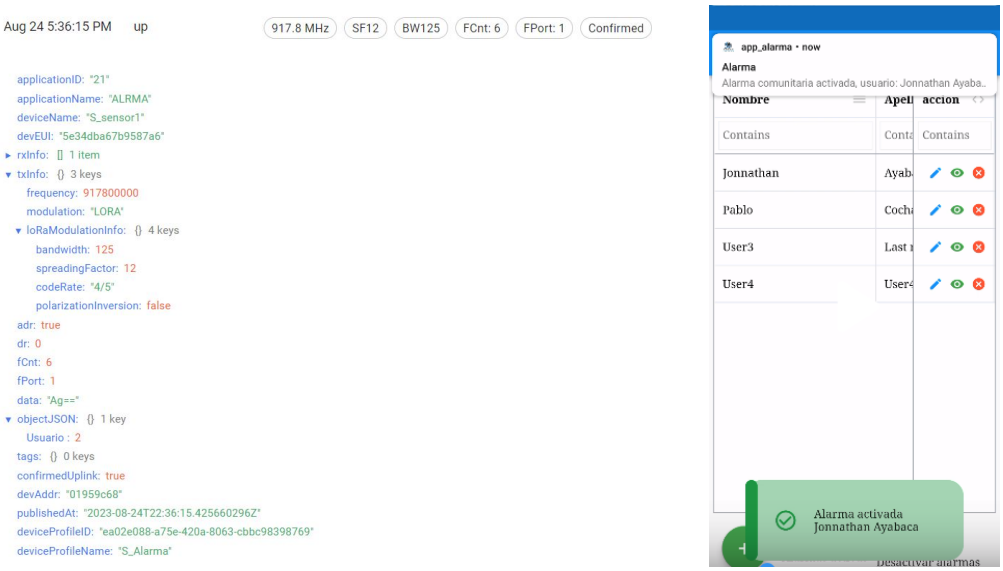

Figura 3.3: Uplink ChirpStack y Visualización en aplicativo móvil [Fuente Autores].

Además, en la pantalla de visualización se muestra el nombre del usuario que activó el botón de pánico. La figura 3.4 presenta el mensaje de visualización y los mensajes recibidos en el protocolo MQTT.

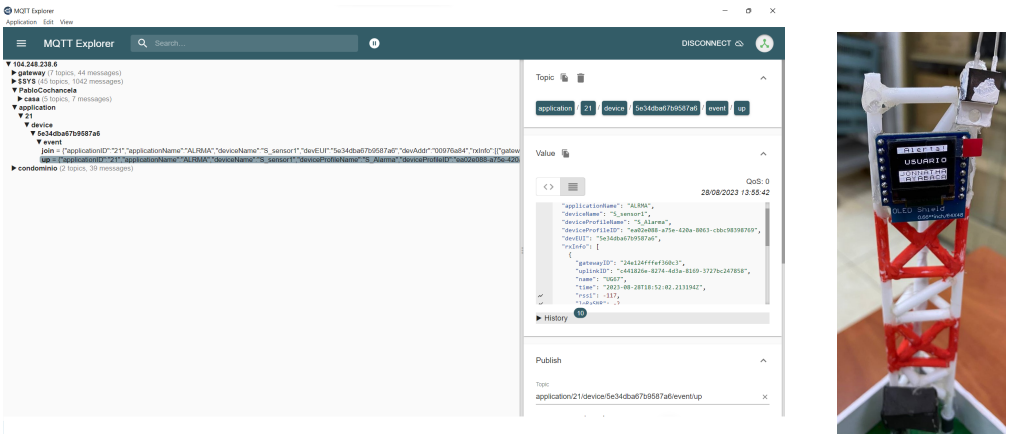

Figura 3.4: Protocolo MQTT y Visualización en pantalla informativa.

Como último paso en el funcionamiento, se activa una sirena que contribuye a persuadir la alerta generada. Esta sirena solo puede ser desactivada a través de la aplicación móvil por un usuario con privilegios de administración. Para una exploración más detallada de esta parte, se ofrece una profundización en la sección 3.2.

3.1.2. Funcionamiento de Alarma Residencial integrada a sistema Alarma comunitaria

El funcionamiento de la alarma residencial que se integra al sistema de alarma comunitaria parte de la instalación previa de un sistema de alarma básica DSC 585 instalado en el primer domicilio como muestra en la figura 3.5. La alarma está equipada con tres sensores: dos sensores magnéticos ubicados en la puerta principal y la ventana frontal. Además, un sensor de movimiento ubicado dentro del domicilio y una sirena.

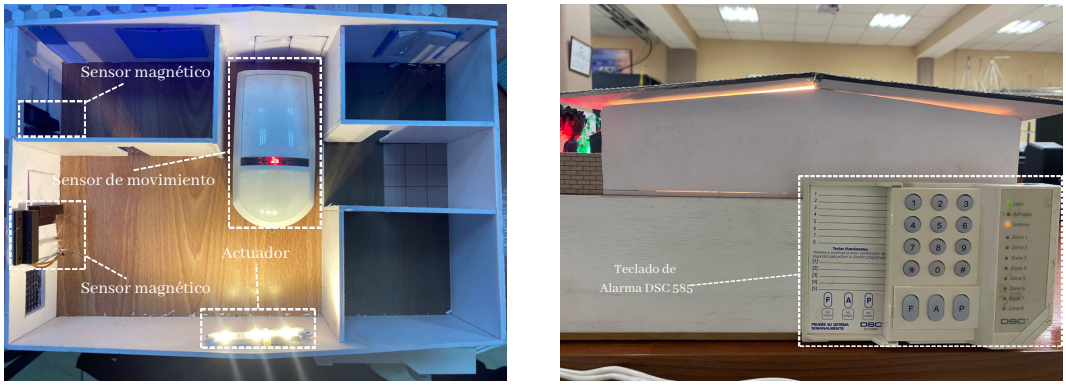

Figura 3.5: Ubicación de alarma DSC [Fuente Autores].

La gestión de la alarma residencial se realiza por medio del protocolo MQTT el cual envía el estado de la alarma a la aplicación móvil. Esta aplicación presenta el funcionamiento a través de la verificación de estados como armado y desarmado de la alarma residencial, zonas abiertas o cerradas, estado de energía, actuadores e historial de alertas.

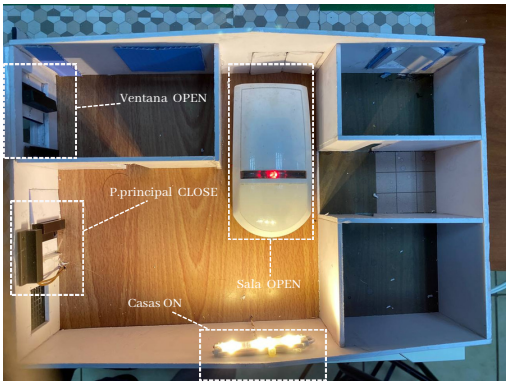

Figura 3.6: Estado de alarma residencial [Fuente Autores].

En la figura 3.6 se observa el estado de cada sensor de forma física;

esto permite verificar el estado abierto o cerrado y comprobar mas adelante su funcionamiento.

La Figura 3.7 muestra el mensaje MQTT recibido en el Broker Mosquitto, mientras que en la Figura 3.8, se detalla la información que aparece en el teclado instalado en la casa con sistema de alarma residencial, así como lo visualizado en la interfaz de gestión de dicha alarma. Esta comparación permite verificar que el sistema está proporcionando coherencia en la información, tal como se evidencia en la Figura 3.6. Además, se observan los datos transmitidos tanto al Broker, como lo presentado en el teclado y en en el aplicativo móvil, son los mismos.

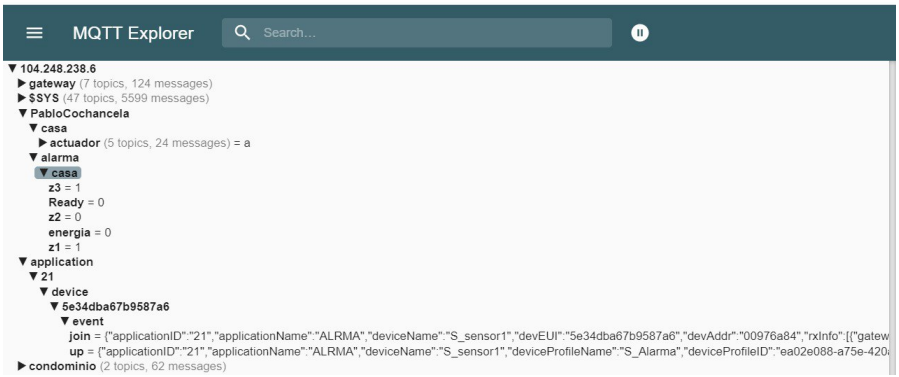

Figura 3.7: Mensaje recibido del estado de las zonas de la alarma residencial en el broker mosquitto MQTT [Fuente Autores].

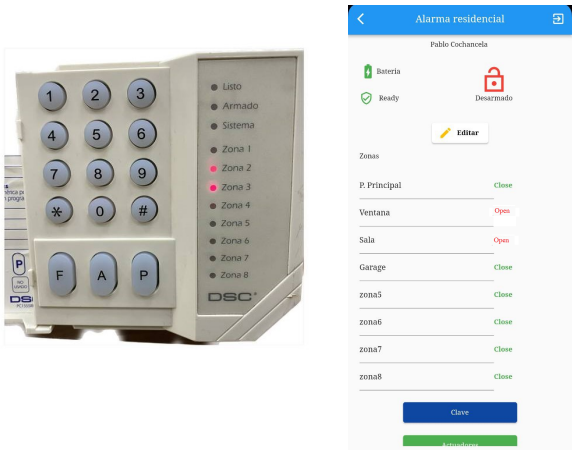

Figura 3.8: Estado de zonas de la alarma residencial en teclado físico y aplicativo móvil [Fuente Autores] .

La gestión de alarma residencial también permite el control de actuadores, como se muestra en la figura 3.6, donde el actuador denominado

"Casas" se encuentra encendido. La figura 3.9 ilustra el estado de los actuadores en el aplicativo móvil.

Figura 3.9: Gestión de actuadores por aplicativo móvil [Fuente Autores].

Al encontrarse el sistema de alarma residencial armado, en caso de existir intrusión, ya sea por apertura de los sensores magnéticos o por movimiento, se genera la activación de una sirena y un aviso a través de la Red LoraWan usando el servidor ChirpStark el cual envía una alerta a través del protocolo MQTT al aplicativo móvil. La figura 3.10 muestra la llegada del mensaje al servidor Chirstack y la visualización de notificaciones push.

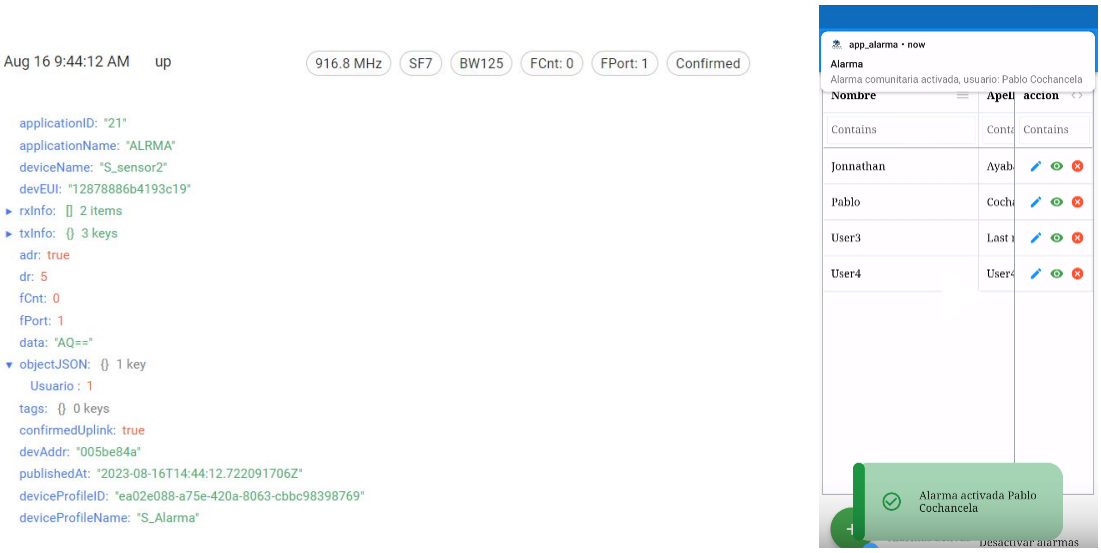

Figura 3.10: Uplink ChirpStack y visualización de notificación en aplicativo móvil [Fuente Autores].

El funcionamiento concluye con la visualización de el nombre del usuario en la pantalla informativa en la maqueta mediante protocolo MQTT.

La figura 3.11 ilustra los mensajes recibidos del protocolo MQTT y la visualización en la pantalla del usuario.

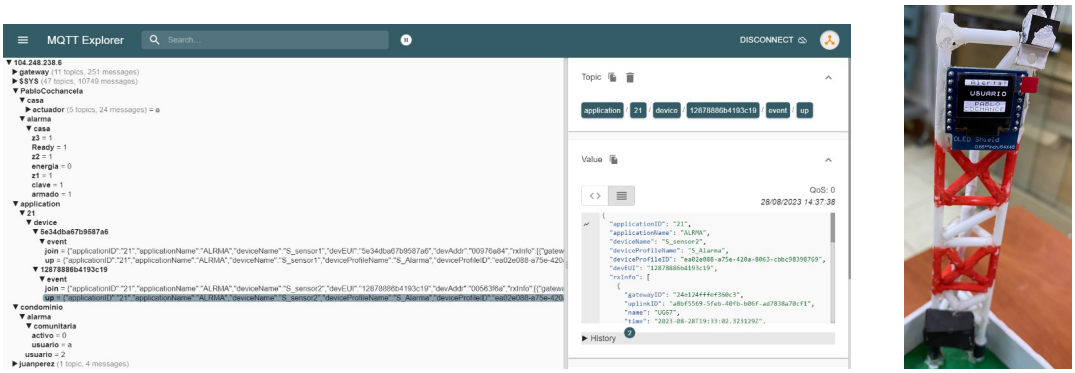

Figura 3.11: Mensaje de alerta recibido por Protocolo MQTT y Visualización de nombre de usuario en pantalla informativa [Fuente Autores].

3.2. Interfaces del Aplicativo móvil

En esta sección se expone en detalle el funcionamiento de cada interfaz que conforma el aplicación móvil. Se compone de tres interfaces fundamentales: interfaz de administración de usuarios, interfaz de gestión de la alarma comunitaria e interfaz de alarma residencial.

A continuación, se describe en cada una de las secciones el funcionamiento de cada parte que integran la aplicación.

3.2.1. Interfaz de administración de usuarios

Esta interfaz permite la visualización, modificación y eliminación de usuarios, ya sean residenciales o comunitarios. Esto se logra a través del inicio de sesión en una cuenta de administración. La Figura 3.12 muestra a continuación e ilustra las interfaces de lo mencionado previamente.

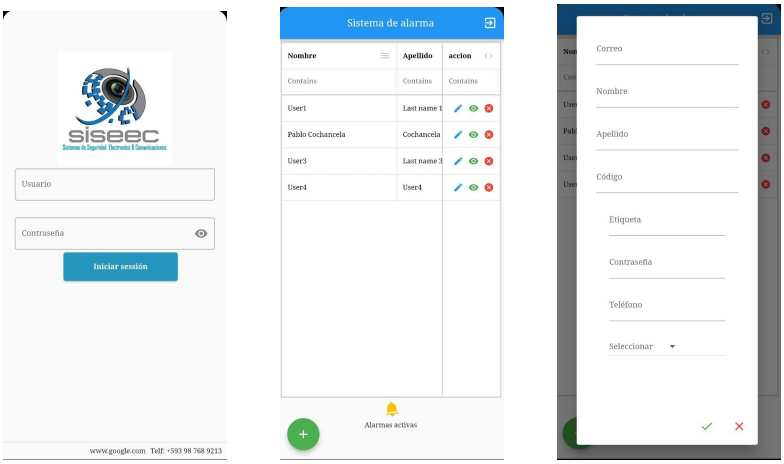

Figura 3.12: Visualización, modificación y eliminación de usuarios [Fuente Autores].

Al iniciar sesión como administrador, también es posible visualizar el historial de alarmas generadas con su respectiva ubicación. Además, si alguna alarma está activa, se puede desactivar únicamente con este usuario, como se muestra en la figura 3.13.

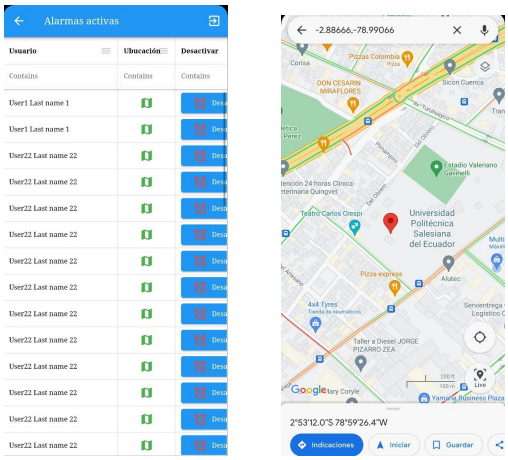

Figura 3.13: Historial de alertas y ubicación [Fuente Autores].

3.2.2. Interfaz de usuarios de alarma comunitaria

El sistema de alarma comunitaria se controla a través de la interfaz de alarma comunitaria a la cual se accede al iniciar sesión ingresando las credenciales validas; se permite activar únicamente el botón de pánico que genera una alerta. La figura 3.14 ilustra las interfaces de inicio de sesión y botón de pánico.

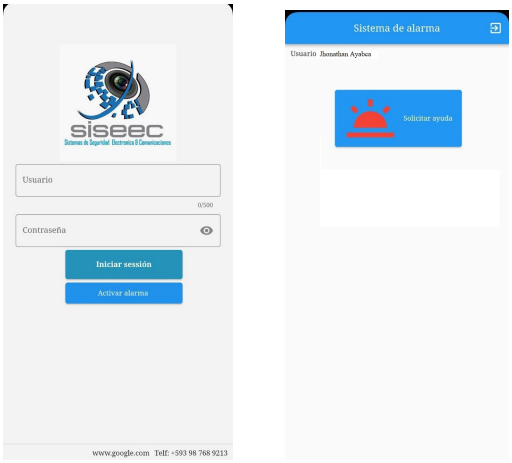

Figura 3.14: historial de alertas y ubicación [Fuente Autores].

De igual manera al seleccionar el botón de alerta, se genera un aviso en la pantalla y se activa la sirena en la maqueta como se muestra en la figura 3.4.

3.2.3. Interfaz de usuarios de alarma residencial integrado a sistema de alarma comunitaria

Para la administración de una alarma residencial, se presentan las siguientes interfaces, como se muestra en la figura 3.15. Estas interfaces posibilitan la verificación de diversos detalles de la alarma residencial, como el proceso de armado y desarme, la identificación de zonas abiertas, historial de alertas y facilita el ingreso de la clave de activación para el armado de la alarma.

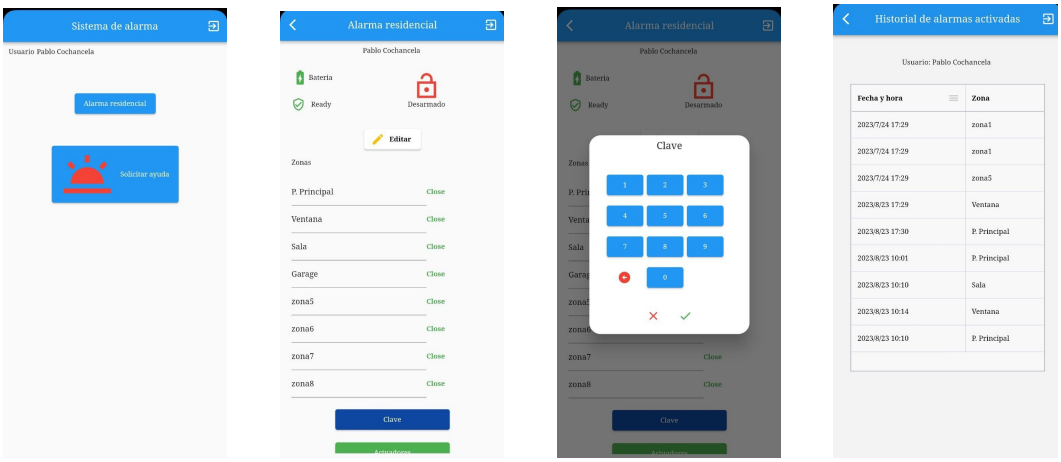

Figura 3.15: interfaces de alarma residencial: botón de pánico alarma comunitaria, gestión y control de alarma residencial.

Al presionar el botón de pánico del aplicativo móvil se genera una alerta en el sistema de alarma comunitaria por medio de una sirena y la visualización del usuario que envió la alerta en la pantalla de información como se mostró en la figura 3.11.

3.3. Análisis de funcionamiento en espacio libre

Con el propósito de validar el desempeño de la plataforma, se lleva a cabo pruebas en un entorno libre para evaluar la eficacia del botón de pánico móvil. Para este propósito, se selecciono un vecindario en las proximidades de la Universidad Politécnica Salesiana sede El Vecino de la ciudad de Cuenca. Se

elige esta ubicación debido a que el gateway de Lorawan se encuentra instalado en dicho lugar, a una altitud de 25 metros. el objetivo es evaluar la cobertura proporcionada en esta área.

Los puntos específicos seleccionados para las pruebas son:

- Parque de la Luz.
- La calle Juan Strobbel.
- Parque Miraflores.
- Bodegas de Juan Eljuri.
- Calle Buena Vista sector Los trigales y la intersección de la Calle Mariano Cueva y Américas.

En los puntos designados, se realizaron activaciones utilizando el botón móvil, y se confirmó la recepción de las alertas en el servidor de aplicaciones ChirpStack. Además, se lleva a cabo una simulación de los enlaces utilizando Radio Mobile, cuyos resultados se detallan a continuación.

3.3.1. Área de cobertura de botón de pánico

Con el objetivo de realizar un análisis exhaustivo de la cobertura del botón de pánico, se llevó a cabo una simulación del sistema que tuvo en cuenta las especificaciones tanto del gateway como del dispositivo móvil. Estos parámetros se introdujeron en el software de simulación Radio Mobile, una herramienta que permite evaluar con precisión la cobertura y el comportamiento del sistema en la ubicación geográfica seleccionada para las pruebas de campo. Para la simulación, se agregaron los puntos seleccionados para pruebas, las cuales se mencionan al inicio de esta sección. S incluye el gateway, el cual está ubicado dentro de la Universidad Politécnica Salesiana. Además, la simulación brinda la capacidad de constatar y verificar si los datos recopilados en el campo coinciden con las predicciones de la simulación, lo que resulta fundamental para garantizar la efectividad y confiabilidad del sistema en situaciones reales.

En la figura 3.16, se puede apreciar la cobertura proporcionada por el botón de pánico. Esta cobertura se derivada de la simulación de la red LoRaWAN en el software Radio Mobile la cual se relaciona mas adelante con las pruebas de campo realizadas en la zona geográfica seleccionada.

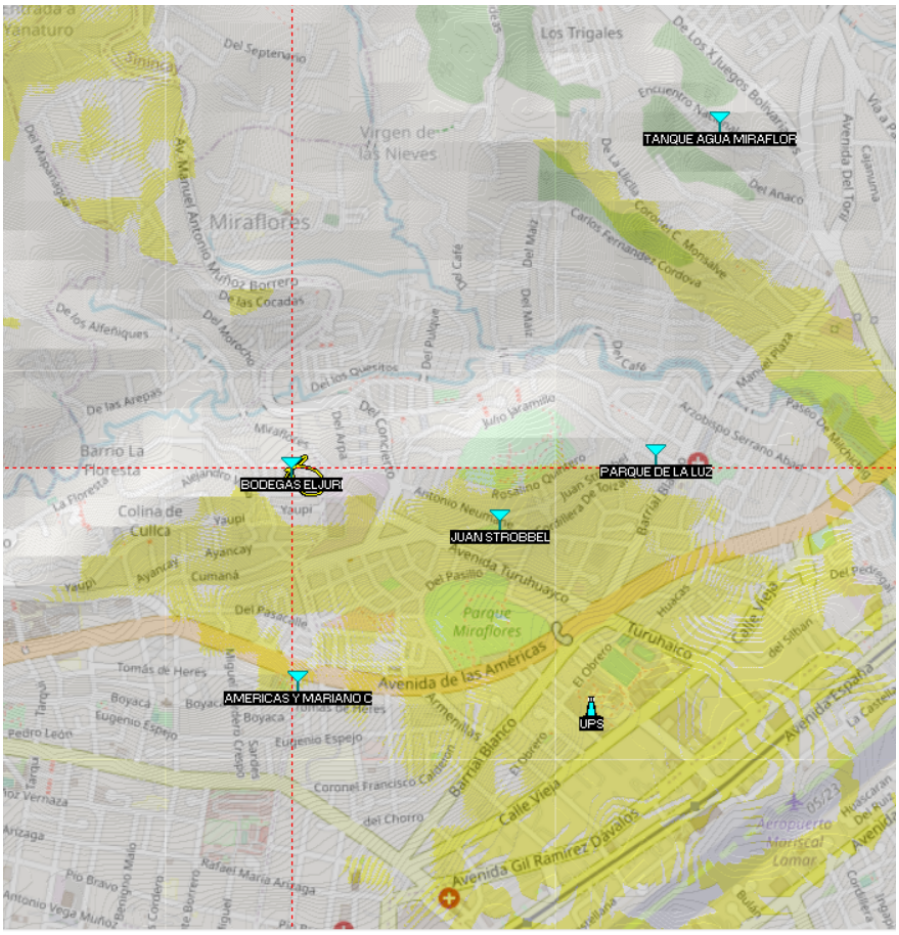

Figura 3.16: Área de cobertura simulado en Radio móbile

3.3.2. Primer escenario: Parque de la luz

La ubicación es el Parque de la Luz, debido a la elevación del terreno en esta zona, la cobertura resultó integral, ya que se estableció una línea de visión directa entre el Gateway y el botón de pánico móvil. Con el fin de llevar a cabo una prueba más realista, se realizo un desplazamiento de ubicación hacia una zona más baja en el sector, la intersección de la Calle Julio Jaramillo y la Cordillera de Toizán. En este punto, pese ala presencia de obstáculos, se logro que la alerta fuese transmitida exitosamente al servidor de aplicaciones, al estar en este escenario el envió no es inmediato y el botón reenvía el mensaje hasta que se confirma la recepción. Se observar la simulación en la figura 3.17.

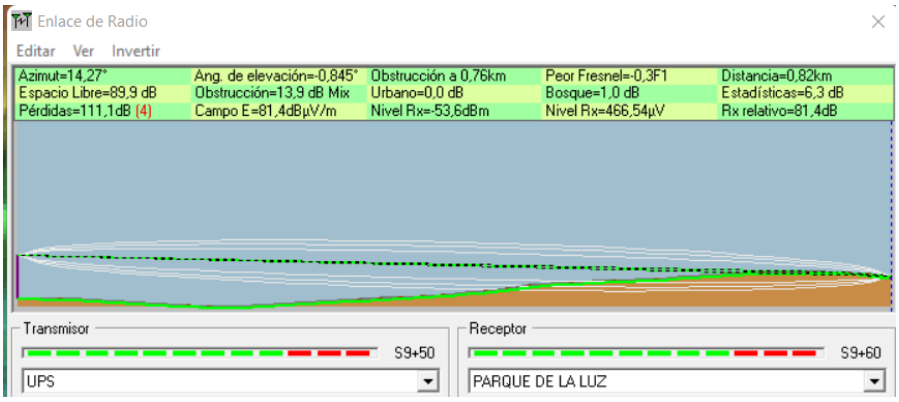

Figura 3.17: Simulación enlace entre Gateway Lorawan y botón móvil sector parque de la Luz.

Uno de los aspectos primordiales que se busca analizar en la simulación es la distancia del enlace, ya que esto proporciona un indicativo clave sobre la cobertura efectiva del botón de pánico. En este escenario particular, la distancia del enlace se registra a 0.82 kilómetros. No obstante, la distancia en sí es solo una parte de la ecuación.

Otra métrica esencial es lo que se conoce como el "peor Fresnel". En este contexto, la evaluación del peor *Fresnel* adquiere importancia, pues revela que se esta lidiando con un valor de 0.3F1. Esto tiene una implicación crítica: denota que aproximadamente un 30 % de la primera zona de *Fresnel*l está obstruida. Esta obstrucción puede derivar de obstáculos físicos en la trayectoria de la señal, como edificios o vegetación. La zona de Fresnel, a su vez, es esencial para mantener la integridad de la propagación de las ondas de radio entre el

emisor y el receptor. Un "peor Fresnel" del 0.3F1 sugiere que, aunque existe una conexión, hay una interferencia sustancial que afecta la calidad de la señal. Esto se traducirse en retrasos en la transmisión de la información. En la práctica, esto podría influir en la rapidez con la que se envían las alertas desde el botón de pánico móvil al servidor de aplicaciones ChirpStack. Dado que la calidad y la velocidad de la transmisión son esenciales en los sistemas de alerta, este dato resalta la necesidad de considerar medidas para mejorar la propagación y minimizar la obstrucción de la zona de *Fresnel*

3.3.3. Segundo escenario: Parque Miraflores

Se explora la ubicación situada a una elevación ligeramente inferior que la del Parque de la Luz, exactamente calle Juan Strobbel y Turuhuayco, caracterizada por una densidad de viviendas más notable. El propósito es evaluar la operación del botón de pánico en este entorno y observar que al enviar las alertas, el botón respondió de manera instantánea, sin evidencia de demoras apreciables. Esto implica que el flujo de alertas hacia el servidor de aplicaciones es fluido y eficiente en este contexto.

Además de las observaciones prácticas, se realizo una simulación del enlace para obtener información más detallada sobre el rendimiento del sistema. Los resultados de esta simulación se presentan en la figura 3.18, brindan una visión más completa de factores cruciales, como la calidad de la señal y la propagación en este entorno específico. Estos datos son esenciales para respaldar la toma de decisiones informadas y optimizar aún más la funcionalidad del botón de pánico en distintos contextos geográficos y urbanos.

En términos de distancia del enlace, se observa que esta se establece en 0.65 kilómetros. La cifra de 2.1F1 implica que existe una zona de *Fresnel* relativamente despejada en gran medida, lo que resulta en una propagación de señal eficaz y confiable entre el botón de pánico y el servidor de aplicaciones. Esta situación es particularmente valiosa en sistemas de alerta, donde la velocidad y la confiabilidad en la entrega de mensajes son esenciales. Así, estos resultados indican que la disposición geográfica de esta ubicación específica, junto con una limitada obstrucción en la zona de *Fresnel*, contribuye de manera

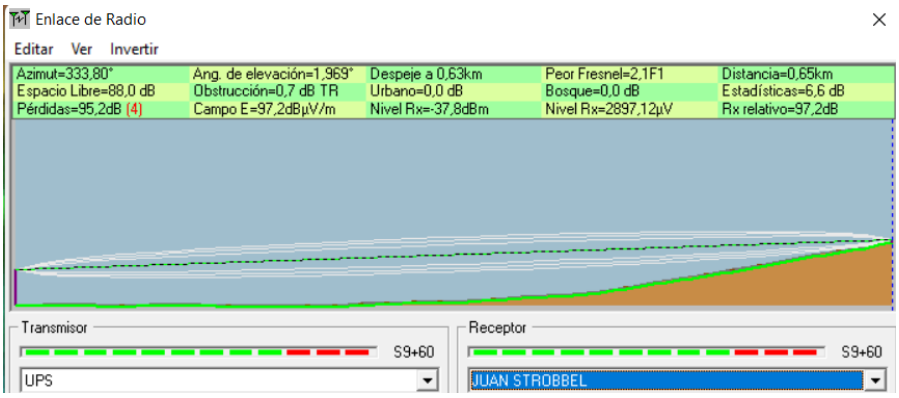

Figura 3.18: Simulación enlace entre Gateway Lorawan y botón móvil sector parque Miraflores

positiva a la eficiencia de la comunicación, sin presentar retrasos notables en el proceso.

3.3.4. Tercer escenario: Bodegas Juan Eljuri

La tercera ubicación es las Bodegas de Juan Eljuri, una ubicación que carece de una línea de vista directa hacia el Gateway. Aquí se llevo a cabo la misma prueba de envío de alertas al servidor de aplicaciones. En este contexto, si bien el envío de la alerta no ocurre de manera instantánea, el botón de pánico implementa una estrategia de retransmisión. Este proceso de retransmisión implica que el botón intenta enviar la alerta en tres ocasiones distintas, persistiendo hasta que se logra confirmar el envío. En este sentido se realiza la simulación del enlace para verificar los datos que obtienen en simulación y observar como se refleja esto en la practica como se ve en la figura 3.19.

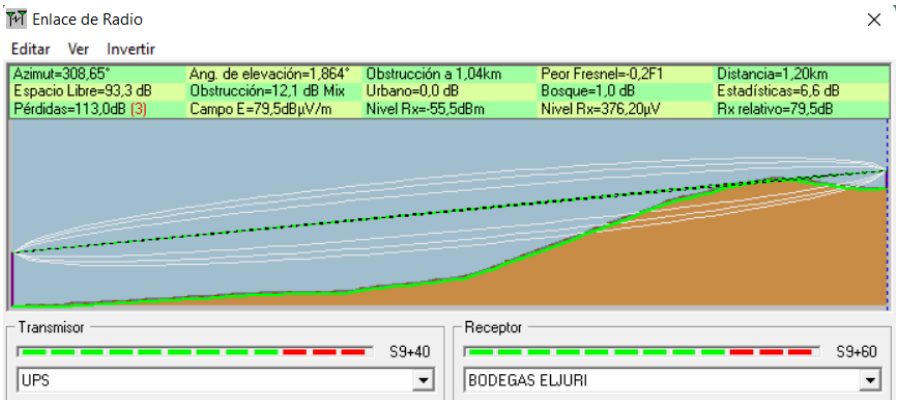

Figura 3.19: Simulación enlace entre Gateway Lorawan y botón móvil sector bodegas El juri

En el análisis de este enlace, se evidencia una distancia de 1.2 kilómetros, lo que resalta la persistente cobertura del botón móvil en este rango. Un factor clave que merece atención es el valor de *Fresnel*, que se registra en 0.2F1. Esta cifra implica que la zona de *Fresnel* se encuentra mayormente despejada, con solo una fracción reducida de obstrucción presente. Este dato es altamente prometedor para la propagación de señales de radio, ya que señala que la mayor parte de la ruta de la señal se encuentra libre de interferencias sustanciales. Asimismo, se destaca que la simulación y la realidad de este enlace no están estrechamente alineadas. En la practica observa una demora más notable en comparación con otros escenarios. Esta discrepancia entre la simulación y la experiencia real puede ser atribuida a obstáculos específicos presentes en el entorno donde se realizó la prueba. Estos obstáculos, intrínsecos a la zona, podrían estar interfiriendo con la propagación de la señal y contribuyendo a los retrasos que observan.

Esta variación entre la simulación y la ejecución en el campo subraya la importancia de considerar no solo los parámetros ideales en un entorno simulado, sino también las condiciones y limitaciones reales del entorno físico. La presencia de obstáculos y características geográficas pueden generar un impacto significativo en el rendimiento de la comunicación inalámbrica y en la velocidad de transmisión de datos. Estas discrepancias entre la simulación y la realidad resaltan la necesidad de pruebas en situaciones reales para obtener una comprensión completa y precisa del rendimiento de los sistemas de comunicación.

3.3.5. Cuarto escenario: Hermano Miguel

Como ultimo escenario la ubicación es Miraflores, concretamente al sector Hermano Miguel, se realiza una evaluación de la función de envío de alerta. Aunque se logra transmitir la alerta con éxito después de varios intentos, se observo que el proceso de envío presentó ciertas dificultades adicionales. Parte de esta complejidad puede atribuirse a la vegetación presente en la zona, la cual ejerce una influencia en la propagación de la señal. En este entorno particular, la vegetación puede actuar como un obstáculo adicional en

la trayectoria de la señal inalámbrica. La densidad y altura de la vegetación pueden causar reflexiones y absorción de la señal, lo que afecta la eficiencia de la transmisión. Estos factores pueden contribuir a los desafíos que observaron en el proceso de envío de alertas. Aunque se logra transmitir la alerta, la presencia de la vegetación generó una mayor complejidad y un esfuerzo adicional en comparación con escenarios más libres de obstrucciones.

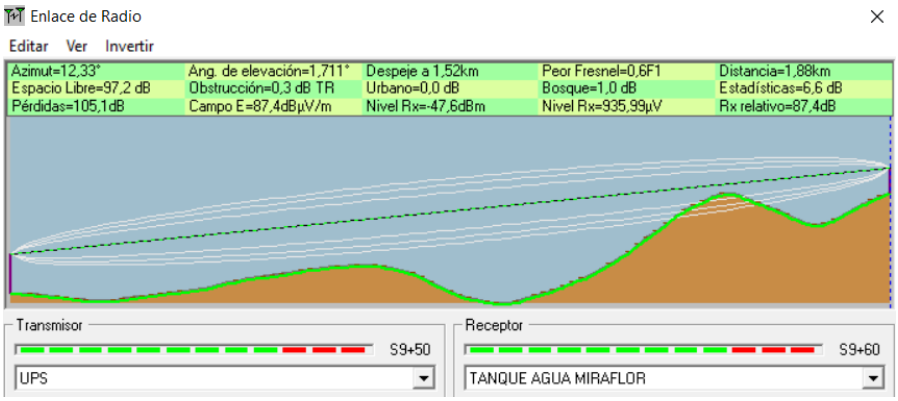

Figura 3.20: Simulación enlace entre Gateway Lorawan y botón móvil sector Hermano Miguel

Esta evaluación adquiere una importancia significativa al considerar que la distancia alcanzada es de 1.88 kilómetros, lo que resalta tanto la robusta cobertura del botón como la impresionante confiabilidad del sistema, incluso en esta distancia considerable. Este resultado subraya la capacidad del sistema para mantener una comunicación efectiva y confiable en una distancia tan extensa.

Un dato relevante obtenido durante la simulación es el valor de 0.6F1 correspondiente al peor *Fresnel*. Este valor es de gran relevancia ya que sugiere que en la zona de *Fresnel* existe una obstrucción apreciable. Este fenómeno puede tener implicaciones en la calidad de la señal y, en casos extremos, se traduce en retrasos en la transmisión. A medida que la obstrucción en la zona de *Fresnel* se acrecienta, las probabilidades de interferencia y pérdida de señal aumentan, lo que podría afectar la confiabilidad de la comunicación inalámbrica entre los dispositivos. Estos aspectos resaltan la importancia de equilibrar la distancia con la calidad de la señal en entornos con obstáculos para garantizar una comunicación sólida y efectiva en todas las condiciones.

3.4. Estimación de Costos para el Desarrollo de la Plataforma

A continuación en esta sección, se presenta una estimación detallada de los costos asociados a los productos utilizados en la ejecución de este proyecto.

3.4.1. Costo general para desarrollo del proyecto

La Tabla 3.1 muestran los costos correspondientes a cada dispositivo empleado en el desarrollo y implementación de la plataforma.

Los dispositivos como gateway, módulos GSM y un dispositivo final CubeCell fueron generosamente proporcionados por el grupo de investigación GITEL. Teniendo en cuenta esta contribución, el costo total de todos los dispositivos utilizados es: **$ 1,207.80**.

Tabla 3.1: Costos de dispositivos

Cantidad	Descripción	V unitario	V total
2	Sirenas	$ 15.00	$ 30.00
2	Cubecell GPS	$ 30.00	$ 60.00
2	ESP8266	$ 9.00	$ 18.00
1	Gateway LoRaWAN	$ 896.00	$ 896.00
1	Alarma DSC	$ 126.00	$ 126.00
2	Banco de relés	$ 4.90	$ 9.80
2	Modulo GSM	$ 13.00	$ 26.00
2	Tarjetas SIM	$ 3.00	$ 6.00
1	Pantalla Oled	$ 9.50	$ 9.50
1	Batería litio 3.3v	$ 12.50	$ 12.50
1	Batería 12v	$ 14.00	$ 14.00
		Total	$ **1,207.80**

A continuación, se detallan los costos por servicios, los cuales incluyen la configuración de dispositivos como CubeCell y ESP8266, además del costo por hora para el diseño e implementación de la aplicación móvil. También se incluye el costo total de la mano de obra requerida para la conexión de los equipos. El total es de **$ 850.00** mostrado en la tabla 3.2.

La Tabla 3.3 presenta los costos mensuales de los servicios en la nube necesarios para el funcionamiento de la plataforma que tiene el costo de **$ 17.00**.

Tabla 3.2: Costos de servicios

Cantidad	Descripción	V unitario	V total
1	Configuración de Equipos	$ 50.00	$ 50.00
40	Horas de diseño de aplicación móvil	$ 15.00	$ 600.00
1	Mano de obra para implementación de la plataforma	$ 200.00	$ 200.00
		Total	$ **850.00**

Es relevante destacar que el costo del servicio de AWS donde se aloja el servidor ChirpStack está cubierto por el grupo de investigación GITEL; mientras que el servicio de Digital Ocean donde esta alojado el servidor Mosquitto MQTT se encuentra dentro de la capa gratuita por 60 días.

Tabla 3.3: Costos de Servicios por mes

Cantidad	Descripción	V unitario	V total
1	Servidor de Amazon	$ 11.00	$ 11.00
1	Servidor de Digital Ocean	$ 6.00	$ 6.00
		Total	$ **17.00**

Los costos varios que se muestran en el Tabla 3.4 incluyen: el total del desarrollo de la maqueta Por último, también se contemplan los gastos relacionados con los cables utilizados para la conexión de todos los equipos.

Tabla 3.4: Costos varios

Cantidad	Descripción	V unitario	V total
1	Maqueta	$ 200.00	$ 200.00
10	Metro de cable Ethernet cat 5e	$ 0.50	$ 5.00
10	Metro de cable gemelo # 18	$ 0.27	$ 2.70
		Total	$ **207.70**

El monto total invertido en el desarrollo del proyecto asciende a **$2,282.50**. Sin embargó, se pueden reducir costos mensuales de alojamiento de los servidores de red ChirpStack y Mosqitto a través de una infraestructura local.

3.4.2. Costo de implementación y funcionamiento de alarma comunitaria

Para el funcionamiento de la alarma comunitaria, se consideran diversos dispositivos y servicios que se detallan en la tabla 3.5. El costo total de este proyecto asciende a **$ 1,362.50**, y será repartido entre todos los usuarios del sistema. Además, los gastos de servicio mensuales serán aportados por los usuarios a través de la cuota del condominio. Es importante destacar que los costos variables dependerán del lugar específico donde se realizará la instalación de los equipos.

Tabla 3.5: Implementación y funcionamiento alarma comunitaria

Cantidad	Descripción	V unitario	V total
	Costo de dispositivos		
1	Sirenas	$ 15.00	$ 15.00
1	ESP8266	$ 9.00	$ 9.00
1	Gateway LoRaWAN	$ 896.00	$ 896.00
1	Modulo GSM	$ 13.00	$ 13.00
1	Tarjetas SIM	$ 3.00	$ 3.00
1	Pantalla Oled	$ 9.50	$ 9.50
	Costos de servicios		
1	Configuración de Equipos	$ 50.00	$ 50.00
1	Mano de obra para implementación de la plataforma	$ 200.00	$ 200.00
	Costos de Servicios por mes		
1	Servidor de Amazon	$ 11.00	$ 11.00
1	Servidor de Digital Ocean	$ 6.00	$ 6.00
	Costos varios		
1	Cables para conexiones de sistemas de alarmas	$ 150.00	$ 150.00
		Total	**$ 1,362.50**

Una vez que se ha determinado el costo de implementación y funcionamiento de la alarma comunitaria, se ofrecen dos tipos de servicio: uno incluye la integración de una alarma residencial, mientras que el otro consiste únicamente en un nodo final para la alarma comunitaria. A continuación, se detalla el costo individual para cada usuario de ambos servicios.

3.4.3. Costo de integración de alarma residencial al sistema de alarma comunitaria

La Tabla 3.6 muestra los costos de integración de una alarma residencial que el usuario ya posee en su domicilio al sistema de alarma comunitaria. El costo es de **$ 132.90**. se toma en cuenta que este costo es por primera ves y consiguiente se le cobrara **$ 6** por mes por el uso del aplicativo móvil.

Tabla 3.6: Integración Alarma residencial

Cantidad	Descripción	V unitario	V total
	Costo de dispositivos		
1	Cube cell GPS	$ 30.00	$ 30.00
1	ESP8266	$ 9.00	$ 9.00
1	Banco de relés	$ 4.90	$ 4.90
1	Modulo GSM	$ 13.00	$ 13.00
	Costos de servicios		
1	Configuración de Equipos	$ 20.00	$ 20.00
1	Monitoreo de aplicativo móvil	$ 6.00	$ 6.00
1	Mano de obra para implementacion del sistema	$ 50.00	$ 50.00
		Total	**$132.90**

3.4.4. Costo de implementación de nodo final para alarma comunitaria

En la Tabla 3.7 se presenta el costo del servicio como nodo final para la alarma comunitaria, el cual asciende a **$ 112.50** por cada usuario que lo desee. Este costo cubre tanto el diseño como la puesta en funcionamiento de un botón de pánico.

Tabla 3.7: Nodo para alarma comunitaria

Cantidad	Descripción	V unitario	V total
	Costo de dispositivos		
1	Cubecell GPS	$ 30.00	$ 30.00
1	batería litio 3.3v	$ 12.50	$ 12.50
	Costos de servicios		
1	Configuración y desarrollo del nodo	$ 70.00	$ 70.00
		Total	**$112.50**

Capítulo 4

Conclusiones, Recomendaciones y Trabajos Futuros

4.1. Conclusiones

Con el objetivo de proyectar un sistema que use una menor cantidad de puertas de enlace entre los nodos de alarma y el acceso al servidor, se eligió como nodo a la tarjeta CubeCell que es de las más baratas en el mercado a la fecha de edición de este trabajo. En este sentido, cada una de las residencias que cuenten con la alarma residencial podrán enviar alertas al sistema de alarma comunitaria usando LoRaWAN. También, ya que los nodos son fijos, LoRa tiene baja latencia de datos en estos casos y ante la bondad de la distancia que permite la tecnología, se puede hablar una cobertura de casas en un radio promedio de 2Km en la ciudad.

El uso de tecnología LoRa y LoRaWAN ha permitido el monitoreo de diversas variables con alcances que superan los 2 Km en entornos urbanos. Este proyecto, realizó pruebas de la cobertura de los botones de pánico móviles, y los resultados fueron notables. Se comprobó que estos dispositivos alcanzaron una distancia de 1.8 Km en línea directa con el Gateway, lo cual es ligeramente inferior a la cobertura mínima de 2 Km establecida en el proyecto, aunque cuando se considera la cobertura radial, se alcanza casi los 4 Km.

El protocolo de comunicación MQTT facilita la integración fluida entre el sistema de alarma residencial y el sistema de alarma comunitaria. Este

protocolo, reconocido por su ligereza y eficiencia en la transmisión de datos, minimiza la carga en la red y optimiza el consumo energético, convirtiéndose en una elección idónea para dispositivos de recursos limitados.

Los servicios en la nube de AWS para la gestión del servidor de aplicación ChirpStack permite recibir y administrar la información de los nodos finales que están conectados al Gateway Milesight. La elección de los servicios en la nube de AWS se basa en su capacidad de escalar de manera eficiente de acuerdo a los requerimientos del proyecto, garantizando así la escalabilidad necesaria.

En el presente trabajo, la arquitectura del sistema permite que todas las alertas sean registradas en una base de datos usando un servidor de aplicación con un Broker MQTT. En este sentido, es posible bridar soluciones de acceso híbridas y no solo LoRa.

Desarrollar una solución híbrida permite abaratar costos, esto debido a que el valor de los nodos LoRa es mayor en contraste con el valor de los nodos WiFi. Así, el nodo de la alarma comunitaria tiene acceso al servidor MQTT a través de una conexión WiFi, esto se aplicó ya que, el escenario que usó como referencia permite que la caseta de guardianía de un barrio disponga de esta conectividad. No obstante, para otras alternativas donde no exista disponible el acceso inalámbrico a Internet, lo mejor sería usar LoRa y su registro en el servidor LoRaWAN sería en forma similar al de las residencias. No se indica la localidad y/o barrio de referencia por asuntos de seguridad.

4.2. Recomendaciones

La colocación adecuada del Gateway en la red LoRaWAN amplía su alcance por supuesto. Al elegir una ubicación estratégica, se mejora la recepción y transmisión de señales, maximizando la cobertura y minimizando interferencias. Es esencial para un despliegue eficiente de la red LoRaWAN.

Es posible incrementar la velocidad de conexión de los dispositivos finales en la red LoRaWAN al optar por cambiar de activación OTAA a ABP, Sin embargo, este cambio implica sacrificar un poco de seguridad. La decisión debe considerar cuidadosamente el equilibrio entre la velocidad y la protección

de la comunicación de los dispositivos en la red.

4.3. Trabajos Futuros

Como posibles trabajos a futuro, se podrían abordar diversas mejoras, como la optimización del diseño del botón de pánico. Esto podría implicar la reducción de su tamaño y, al mismo tiempo, una mejora en el diseño de su case.

Además, para mejorar la implementación del protocolo MQTT, se puede considerar la introducción de un servidor intermedio. Este servidor actuaría como un punto central para la publicación y suscripción a diversos temas (topics), con el objetivo de que los dispositivos finales puedan acceder a la información de manera eficiente.

Glosario

ABP Activation By Personalization – Activación por personalización.

AWS Amazon Web Services – Servicios web de Amazon.

BROKER Se refiere al intermediario que ejecuta órdenes de compra y venta de activos en los mercados financieros.

CMD Command Prompt – Símbolo del Sistema.

GPIO General Purpose Input/Output – Entrada general de entrada y salida.

GPS Global Positioning System – Sistema de Posicionamiento Global.

GSM Global System for Mobile – Sistema global para móviles.

IoT Internet of Things – Internet de las Cosas.

IP Internet Protocol– Protocolo de internet.

ISM Scientific and Medical – Banda de frecuencias para la Industria, la ciencia y el área médica.

LAN Local Area Networks – Red de área local.

LCD Liquid Crystal Display – Pantalla de cristal líquido.

LED Light Emitting Diode – Diodo emisor de luz.

LoRa Long Range – Largo Alcancé.

LoRaWAN Long Range Low PowerWide Area Network — Largo Alcance Red de Área Amplia de Baja Potencia.

LPWAN Low Power Wide Area Network — Multiplexación por División de Frecuencias Ortogonales.

LTE Long Term Evolution – Evolución a Largo Plazo, estándar de comunicaciones móviles.

MQTT Message Queuing Telemetry Transport – Transporte de telemetría.

OSI Open Systems Interconnection Model – Modelo Abierto de Interconexión de Sistemas.

OTAA Over The Air Activation — Activación sobre el Aire..

PAD Punto de Agregación de datos..

PCB Printed Circuit Board – Placa de Circuito Impreso.

RAM Random Access Memory – Memoria de acceso aleatorio.

RF Radio Frequency – Frecuencia de radio.

ROM Read Only Memory – Memoria de sólo lectura.

SIA Security Industry Association – Asociación de la Industria de la Seguridad.

SSH Secure Shell – Cubierta Segura.

TCP/IP Transfer Control Protocol/Internet Protocol – Protocolo de control de transferencia/Protocolo de Internet .

UML Unified Modeling Language – Lenguaje Unificado de Modelado.

WAN Wide Area Network – Red de área amplia.

WSN Wireless Sensor Networks – Redes de Sensores Inalámbricos.

Referencias

[1] Primicias, «Seis de cada diez robos en Ecuador se producen a plena luz del día | Primicias,» *Primicias, Diario nacional*, mayo 2023. dirección: https://www.primicias.ec/primicias-tv/sociedad/aumento-robos-fiscalia-ecuador/.

[2] J. C. V. R. Mario Jaramillo Jaramillo Sandra Patricia Ospina Mendoza, *Aplicación de redes de sensores inalámbricos (WSN) en un sistema de seguridad para los equipos móviles de la Universidad EAFIT*, Mayo 2011. dirección: https://repository.eafit.edu.co/bitstream/handle/10784/2397/Mario_Jaramillo_Sandra_Ospina_2011.pdf?sequence=3&isAllowed=y.

[3] J. B. Tobar Quevedo, *Diseño de un sistema de monitoreo de alarmas para la vigilancia de la policia comunitaria aplicada al sector de Capelo*, 2007. dirección: http://repositorio.espe.edu.ec/handle/21000/697.

[4] M. A. Rengel Rivera Mateo Santiago Jimbo Jérez, *Diseño, construcción e implementación de un dispositivo de seguridad que permite la intercomunicación con audio y video entre dos puntos y la activación remota de elementos de seguridad*, enero 2015. dirección: http://dspace.ups.edu.ec/handle/123456789/7544.

[5] E. G. Yunga Muyulema Alex Fabian Zaruma Quituizaca, *Desarrollo e implementación de un prototipo-interfaz para comunicación de protocolo Contact-ID con un sistema de gestión IP, aplicado a dispositivos de seguridad residencial*, marzo 2019. dirección: https://dspace.ups.edu.ec/handle/123456789/17170.

[6] C. A. Orozco Rojas et al., «Tecnologías de integración en la seguridad electrónica destinadas al servicio de monitoreo de alarmas para los hogares en Cali Valle,» Tesis de mtría., 2017. dirección: http://hdl.handle.net/10654/15890.

[7] C. VILLACRÉS, M. EDUARDO y J. VARGAS OLEAS, «Análisis y diseño de un servicio para monitoreo remoto de residencias usando la red móvil celular (hspa+ y lte) que integra sistemas de alarmas y video vigilancia,» Tesis de mtría., 2016. dirección: http://www.dspace.espol.edu.ec/xmlui/handle/123456789/43723.

[8] B. E. Gómez Cumbajín, «Estudio de sistemas de alarma comunitaria. Caso de estudio conjunto residencial Ruiseñor 2,» Tesis de mtría., 2018. dirección: http://repositorio.puce.edu.ec/handle/22000/15072.

[9] R. Group, «LightSYS2 - Guia rápida Usuario,» dirección: https://www.riscogroup.com/latin-america/products/solution/7330.

[10] H. E. S.A.C., «Manual de instalación Alcom Max LTE,» dirección: https://hagroy.com/wp-content/uploads/Manual-usuario-alcom-max-4G-2023.pdf.

[11] D. S. Controls, «Manual de usuario PowerSeries,» dirección: https://cms.dsc.com/download.php?t=1&id=23893.

[12] Linseg, «Panel de alarma comunitaria Linseg AX 65,» dirección: https://linseg.com.pe/product/panel-de-alarmas-ax65/.

[13] M. E. Reynoza Porras, «Diseño de un sistema de seguridad en la empresa MYLCOM contra la intrusión utilizando alarma y aviso de alerta vía VoIp.,» Tesis de mtría., 2017. dirección: http://repositorio.uch.edu.pe/handle/uch/106.

[14] D. S. Controls, «Manual instalación 585,» dirección: https://cms.dsc.com/download2.php?t=1&id=16456.

[15] N. Aakvaag y J.-E. Frey, «Redes de sensores inalámbricos,» *Revista ABB*, vol. 2, n.º 2006, págs. 39-42, 2006. DOI: 10.1049/el:19961384.

[16] L. Iacono, P. Godoy, O. Marianetti, C. G. Garino y C. Párraga, «Estudio de la Integracón entre WSN y redes TCP/IP,» *Memoria Investigaciones en Ingeniería*, n.º 10, págs. 57-68, 2012. dirección: http://www.revistas.um.edu.uy/index.php/ingenieria/article/view/365/434.

[17] R. W. Inga Morocho y E. R. Pozo Iñamagua, «Diseño, desarrollo e implementación de un operador WSN en la nube,» B.S. thesis, 2019. dirección: https://dspace.ups.edu.ec/handle/123456789/18080.

[18] R. F. Martínez, J. O. Meré, F. J. M. de Pisón Ascacíbar, A. G. Marcos y F. A. Elías, *Redes inalámbricas de sensores: teoría y aplicación práctica*. 2009, pág. 58, ISBN: 9788469230077. dirección: https://www.researchgate.net/publication/260265697_Redes_Inalambricas_de_sensores_teoria_y_aplicacion_practica..

[19] J. Fraden, *Handbook of modern sensors: physics, designs, and applications*. American Association of Physics Teachers, 1998, ISBN: 978-3-319-19302-1. DOI: 110.1007/978-3-319-19303-8. dirección: http://link.springer.com/10.1007/978-3-319-19303-8..

[20] L. Machado y E. Inga, «Optimal Placement of UDAP in Advanced Metering Infrastructure for Smart Metering of Electrical Energy Based on Graph Theory,» *Electronics (Switzerland)*, vol. 11, 11 jun. de 2022, ISSN: 20799292. DOI: 10.3390/electronics11111767.

[21] E. Inga, J. Inga y A. Ortega, «Novel approach sizing and routing of wireless sensor networks for applications in smart cities,» *Sensors*, vol. 21, 14 jul. de 2021, ISSN: 14248220. DOI: 10.3390/s21144692.

[22] M. A. Labrador y P. M. Wightman, *Topology Control in Wireless Sensor Networks: with a companion simulation tool for teaching and research*. Springer Science & Business Media, 2009, ISBN: 9781402095849.

[23] K. Rose, S. Eldridge y L. Chapin, «The internet of things: An overview,» *The internet society (ISOC)*, vol. 80, págs. 1-50, 2015. DOI: 20151014_0. dirección: https://www.academia.edu/download/48790442/ISOC-IoT-Overview-20151014_0.pdf.

[24] M. Kranz, *Internet of Things: Construye nuevos modelos de negocio*. Editorial Almuzara, 2017, pág. 159, ISBN: 978-84-1689-488-8. dirección: https://books.google.es/books?hl=es&lr=&id=Szz1DwAAQBAJ&oi=fnd&pg=PT4&ots=yrPnPnfsJt&sig=1t1vdygpaYElvX1BjVmj8ncHXps.

[25] A. J. González García, «IoT: Dispositivos, tecnologías de transporte y aplicaciones,» Tesis de mtría., 2017. dirección: http://hdl.handle.net/10609/64286.

[26] D. B. Machado, C. A. Calderón y L. P. Moreno, «Propuesta de arquitectura para Internet de las Cosas,» *Universidad Tecnológica de La Habana José Antonio Echeverría, Cuba*, 2016. dirección: https://www.researchgate.net/profile/David-Benitez-Machado/publication/320353907_Propuesta_de_arquitectura_para_Internet_de_las_Cosas/links/59df940c0f7e9b2dba839ede/Propuesta-de-arquitectura-para-Internet-de-las-Cosas.pdf.

[27] A. Espinosa, D. Ponte, S. Gibeaux y C. González, «Estudio de Sistemas IoT Aplicados a la Agricultura Inteligente,» 2020. dirección: https://jadimike.unachi.ac.pa/handle/123456789/717.

[28] K. Laudon y C. G. Traver, *E-commerce*. Pearson educación, 2009.

[29] J. Salazar, «Redes inalámbricas,» *Techpedia. Recuperado de: https://core. ac. uk/download/pdf/81581109. pdf*, 2016.

[30] N. Poursafar, M. E. E. Alahi y S. Mukhopadhyay, «Long-range wireless technologies for IoT applications: A review,» en *2017 Eleventh International Conference on Sensing Technology (ICST)*, IEEE, 2017, págs. 1-6. DOI: 10.1109/ICSensT.2017.8304507.

[31] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta y A. Bhatnagar, «CRAIoT: concept, review and application (s) of IoT,» en *2019 4th international conference on internet of things: Smart innovation and usages (IoT-SIU)*, IEEE, 2019, págs. 1-4. DOI: 10.1109/IoT-SIU.2019.8777467.

[32] K. Mekki, E. Bajic, F. Chaxel y F. Meyer, «A comparative study of LPWAN technologies for large-scale IoT deployment,» *ICT express*, vol. 5, n.º 1, págs. 1-7, 2019. DOI: 10.1016/j.icte.2017.12.005.

[33] M. A. Moya Quimbita, «Evaluación de pasarela LoRa/LoRaWAN en entornos urbanos,» 2018. dirección: https://riunet.upv.es/bitstream/handle/10251/109791.

[34] D. Bankov, E. Khorov y A. Lyakhov, «On the limits of LoRaWAN channel access,» en *2016 International conference on engineering and telecommunication (EnT)*, IEEE, 2016, págs. 10-14. dirección: https://riunet.upv.es/handle/10251/109791.

[35] «What are LoRa and LoRaWAN? | The Things Network.,» 2022. dirección: https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/..

[36] I. Ordóñez Monfort, «Estudio de la arquitectura y el nivel de desarrollo de la red LoRaWAN y de los dispositivos LoRa.,» 2017. dirección: http://hdl.handle.net/10609/64365.

[37] ARCOTEL, «ARCOTEL Bandas UDBL LoRaWAN .,» 2018. dirección: https://www.arcotel.gob.ec/wp-content/uploads/downloads/2018/08/ARCOTEL-2018-0661-2018-08-01-RADIOCOMUNICACIONES-MDBA-MATRIZ.pdf..

[38] J. J. Macías Olives, «Diseño e implementación de un prototipo de control y monitoreo de procesos con sistemas embebidos Arduino y Raspberry Pi para PYMES.,» B.S. thesis, Escuela Superior Politécnica de Chimborazo, 2018. dirección: http://dspace.espoch.edu.ec/handle/123456789/9213.

[39] A. Jonathan Álvarez, «Sistemas embebidos en comunicaciones de red ethernet,» Spanish, *Revista Inventum*, vol. 4, n.º 6, págs. 16-19, ene. de 2009. dirección: https://www.proquest.com/scholarly-journals/sistemas-embebidos-en-comunicaciones-de-red/docview/2018733453/se-2.

[40] A. J. R. Eusebio, «Sistemas Embebidos-EL185-201601,» dirección: http://hdl.handle.net/10757/638289.

[41] P. Beynon-Davies, *Sistemas de bases de datos*. Editorial Reverté, 2018, vol. 1, págs. 13-15, ISBN: 9788429143966. DOI: 6795626.

[42] A. Chauhan, «A review on various aspects of MongoDB databases,» *International Journal of Engineering Research & Technology (IJERT)*, vol. 8, n.º 05, págs. 90-92, 2019. dirección: https://www.academia.edu/download/60661268/a-review-on-various-aspects-of-mongodb-databases-IJERTV8IS05003120190921-10051-a8lu8n.pdf.

[43] F. G. Yazbek Almeida, «Implementación del Backend y Frontend para una empresa de servicio de mantenimiento de equipos de Laboratorio Clínico,» B.S. thesis, 2022. dirección: http://dspace.ups.edu.ec/handle/123456789/23886.

[44] I. F. Santamaría Ballesteros y J. S. Rincón Agredo, «Desarrollo de aplicación móvil para la gestión de servicio técnico de computadores,» 2021. dirección: https://repositorio.uniajc.edu.co/handle/uniajc/645.

[45] H. H. MAXIMILIANO PAUCAR y L. C. PALACIOS CAVASSA, «SISTEMA INFORMATICO PARA LA GESTION BACKEND Y FRONT-END DEL SERVICIO A DOMICILIO DE LIMPIEZA DEL STARTUP CLICKCLEAN–2019.,» 2019. dirección: https://repositorio.utelesup.edu.pe/handle/UTELESUP/846.

[46] V. V. Rodríguez, «Desarrollo de aplicaciones móviles multiplataforma con Flutter,» *Repositorio Universidad de Almeria*, vol. 72, 2018. dirección: https://core.ac.uk/download/pdf/304719890.pdf.

[47] N. L. Hernandez y A. S. Florez-Fuentes, «Computación en la nube,» *Mundo Fesc*, vol. 4, n.º 8, págs. 46-51, 2014. dirección: https://www.fesc.edu.co/Revistas/OJS/index.php/mundofesc/article/view/48.

[48] M. A. Hernández Rala, «Aplicación de aprendizaje profundo: clasificación de imágenes médicas basado en servicios en la nube,» 2021. dirección: https://oa.upm.es/69406/1.haslightboxThumbnailVersion/TFG_MICHAEL_ADRIAN_HERNANDEZ_RALA.pdf.

[49] Amazon, «AWS,» dirección: https://aws.amazon.com/es/?nc2=h_lg.

[50] D. Ocean, «Digital Ocean,» dirección: https://www.digitalocean.com/.

[51] A. A. Garcia, *Ciberseguridad: ¿Por qué es importante para todos?* Siglo XXI Editores México, 2019. dirección: https://books.google.es/books?hl=es&lr=&id=ZqHDDwAAQBAJ&oi=fnd&pg=PT5&dq=ciberseguridad+que+es&ots=yhhd45Xtc4&sig=ESDnCQhSk8K5Gnb2b_fL8gSRsoU#v=onepage&q=ciberseguridad%20que%20es&f=false.

[52] A. G. Segura, *Seguridad en la Internet de las cosas: Propuesta de implementación segura de un sistema de seguridad con dispositivos IoT en PYME*. 2019. dirección: https://openaccess.uoc.edu/bitstream/10609/97447/6/adriansegaTFM0619memoria.pdf.

[53] D. R. González, «Arquitectura y Gestión de la IoT,» *Telemática*, vol. 12, n.º 3, págs. 49-60, 2013. dirección: https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/119/115.

[54] R. A. Light, «Mosquitto: server and client implementatixon of the MQTT protocol,» *Journal of Open Source Software*, vol. 2, n.º 13, pág. 265, 2017. dirección: https://joss.theoj.org/papers/10.21105/joss.00265.pdf.

[55] N Aguirre, N Aranda y N Balich, «Seguridad en el envío de mensajes mediante protocolo MQTT en IoT,» *INNOVA UNTREF. Revista Argentina de Ciencia y Tecnología*, 2019. dirección: https://www.revistas.untref.edu.ar/index.php/innova/article/download/1000/826.

[56] B. Mishra, «Performance evaluation of MQTT broker servers,» en *International Conference on Computational Science and Its Applications*, Springer, 2018, págs. 599-609. dirección: https://www.researchgate.net/profile/Biswajeeban-Mishra/publication/326167412_Performance_Evaluation_

of_MQTT_Broker_Servers/links/5f0abf13a6fdcc4ca4635cc2/Performance-Evaluation-of-MQTT-Broker-Servers.pdf.

[57] W. Pipatsakulroj, V. Visoottiviseth y R. Takano, «mumq: A lightweight and scalable mqtt broker,» en *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, IEEE, 2017, págs. 1-6. dirección: https://ieeexplore.ieee.org/iel7/7963870/7972123/07972165.pdf?casa_token=790-zIJ91ooAAAAA:YYkqARiHhALxr0PIchN4Xo-WYVQ1Kw0VmZOLRQc4Mi8MU21pPuyW0XrhKdQPTewobNQeeOh5Gu5EGg.

[58] R. A. D. Vásquez y A. R. L. Yacelga, «Alternativa tecnológica de seguridad comunitaria en tiempos actuales,» *Universidad y Sociedad*, vol. 14, n.º S2, págs. 337-343, 2022. dirección: https://rus.ucf.edu.cu/index.php/rus/article/view/2790.

[59] K. M. ORTEGA y S. L. PINO, «Impacto social y económico de los factores de riesgo que afectan la seguridad ciudadana en Ecuador.,» *Revista Espacios*, vol. 42, n.º 21, 2021. dirección: https://revistaespacios.com/a21v42n21/a21v42n21p04.pdf.

[60] E. Telégrafo, «Robos Incrementaron En El Ecuador Durante 2021, Respecto al Año de Confinamiento | El Telégrafo,» *El Telégrafo, Diario nacional*, junio 2021. dirección: https://www.eltelegrafo.com.ec/noticias/judicial/12/robosincrementaron-en-el-ecuador-durante-2021.s.

[61] N. Choudhary, «DSC Keybus Interface,» dirección: https://github.com/taligentx/dscKeybusInterface.