ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN PROTOTIPO DE RED LPWAN CON TECNOLOGÍA LoRa PARA LA DETECCIÓN DE INTRUSOS EN LAS VIVIENDAS DE UNA ZONA RESIDENCIAL

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

EDWIN FABRICIO AVILA CUEVA

edwin.avila@epn.edu.ec

MIGUEL ÁNGEL PARRA ORDÓÑEZ

miguel.parra@epn.edu.ec

DIRECTOR: Sang Guun Yoo. PhD.

sang.yoo@epn.edu.ec

CODIRECTOR: Jhonattan Barriga A. MSc.

jhonattan.barriga@epn.edu.ec

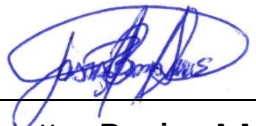
Quito, octubre 2020

AVAL

Certificamos que el presente trabajo fue desarrollado por **Edwin Fabricio Avila Cueva** y **Miguel Ángel Parra Ordoñez** bajo nuestra supervisión.

Sang Gunn Yoo PhD.

DIRECTOR DE PROYECTO



Jhonattan Barriga A MSc.

CODIRECTOR DE PROYECTO

DECLARACIÓN DE AUTORÍA

Nosotros, **Edwin Fabricio Avila Cueva** y **Miguel Ángel Parra Ordóñez**, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Edwin Fabricio Avila Cueva

Miguel Ángel Parra Ordóñez

DEDICATORIA

A mis padres Washington Avila y Angela Graciela, quienes han inculcado en mí el ejemplo de esfuerzo, dedicación y humildad, para no rendirme frente a las adversidades hasta cumplir con mis objetivos.

A mis hermanos Bryan Avila, Erika del Pilar Avila y en especial a mi hermano Javier Avila quién me animó en este campo de estudio y, durante todos los años de universidad, supo apoyarme, guiarme y aconsejarme, ya que con sus palabras de aliento me motivaron a concluir mi carrera

Edwin.

DEDICATORIA

A mis amados padres, Carmen y Rodrigo, quienes son mi inspiración y ejemplo a seguir, todo lo que soy se lo debo a ustedes.

A mis hermanos, Dario, Pamela e Isaac, quienes a pesar de nuestras diferencias siempre han estado brindándome su ayuda cuando la he necesitado. Los adoro hermanos míos, deseo que triunfen y tengan éxito en su vida.

A mi sobrina, Samy, por alegrarme la vida con sus ocurrencias. Mi chinita te quiero inmensamente.

Miguel.

AGRADECIMIENTO

Agradezco en primer lugar a Dios por ser mi guía, mi fortaleza y por bendecirme a mí y a toda mi familia.

A mi padre y a mi madre, que me han brindado su amor, cariño y su apoyo incondicional, en aquellos momentos de dificultad y de debilidad.

A mis hermanos por llenarme de alegría día tras día y por todos los consejos brindados.

A mi compañero y buen amigo Miguel, por la amistad que formamos en esta carrera, por brindarme todo su apoyo y por compartir sus conocimientos he infundir en mí el compañerismo y los ánimos de triunfar.

Y finalmente quiero agradecer a los ingenieros Sang Guun Yoo y Jonathan Barriga, por brindarnos parte de su conocimiento y las pautas para llevar a cabo el desarrollo y la culminación de esta tesis.

Para ellos va dedicado todo el trabajo y esfuerzo puesto en la realización de este proyecto de titulación.

Edwin.

AGRADECIMIENTO

A mis padres, por haber sacrificado mucho de sus vidas para que tanto yo como mis hermanos pudiéramos construir las nuestras, gracias por todo su amor, dedicación y por siempre anhelar lo mejor para nosotros. Ustedes son el motor de mi vida. ¡los amo con todo el corazón!

A mi amada novia, Pamela Valencia, por todo lo que hace por mí, por su apoyo incondicional y sobre todo por compartir su vida conmigo, gracias por ser el complemento que tanto deseaba en la vida.

A mi compañero de tesis, Edwin, que ha demostrado ser un gran amigo y ha sido parte fundamental para la culminación del proyecto, gracias por tu colaboración y por los gratos momentos vividos en nuestra etapa universitaria.

A mi familia y amigos, por la confianza depositaba en mí, y quienes esperaban con ansias este logro.

A ti abuelita querida, aunque partiste de este mundo siempre te llevo presente en mis recuerdos y corazón, gracias por acogerme en esa etapa de mi vida y cuidarme como a uno más de tus hijos. Quisiera que estuvieras aquí sintiéndote orgullosa de mí.

A los ingenieros Sang Guun Yoo y Jonathan Barriga, director y codirector del presente trabajo de titulación, por darnos a conocer este interesante tema y el apoyo proporcionado en cada fase que involucro su finalización.

Finalmente, también agradecer a todas las personas que en algún momento de mi carrera universitaria me han brindado su apoyo para culminarla con éxito.

Miguel.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

La constante expansión de las redes inalámbricas y el crecimiento de la conectividad de los dispositivos inteligentes ha motivado la búsqueda de nuevas soluciones de comunicación que permitan contrarrestar problemas sociales como la inseguridad ciudadana, siendo el robo en las viviendas uno de los más relevantes. Una solución para este tipo de problemas es automatizar el hogar usando sensores que permitan detectar intrusiones en los inmuebles.

Este trabajo se centra en el desarrollo de un prototipo de red LPWAN, que mediante el uso de las tecnologías LoRa y LoRaWAN combinadas con MQTT y una API REST, notifican actos de intrusión en una vivienda tanto al personal de seguridad de la zona residencial como al propietario del inmueble. El personal de seguridad utiliza una aplicación web para gestionar la autenticación de usuarios y supervisar, dentro de un mapa en tiempo real, el estado de los nodos de cada residencia. Por otra parte, el propietario del inmueble puede manejar la activación y desactivación de los nodos a través de una aplicación móvil. Las notificaciones push se han habilitado cada vez que se produce una intrusión o se desconecta un nodo. Los elementos hardware del prototipo trabajan de acuerdo con la especificación 1.0.2 del protocolo LoRaWAN en las bandas ISM 902_928.

Los resultados obtenidos han demostrado que la solución propuesta funciona adecuadamente en entornos urbanos cumpliendo con las características de una red LPWAN, ya que proporciona comunicación de largo alcance y notificaciones inmediatas.

Palabras clave: LPWAN, LoRaWAN, LoRa, seguridad de la vivienda, detección de intrusos.

ABSTRACT

Wireless networks constant expansion and the growth of smart devices connectivity has motivated the search of new communication solutions to provide benefits for social problems like citizen insecurity where home robbery is one of the most relevant. A solution this type of problem is home automation, by using sensors that can detect intrusions at a house.

This work focuses on the development of a prototype of an LPWAN network by using LoRa and LoRaWAN technologies combined with MQTT and an API REST to generate notification alerts of intrusion into a home for the security personal of residential area and the house owner. The security personal uses web application for manage user authentication and monitor, within a real-time map, the status of the nodes of each home. Otherwise, the house owner uses a mobile application for handle activation and deactivation of the nodes. Push notifications has been enabled whenever an intrusion occurs, or a node is disconnected. The elements hardware of the prototype work according to the specification 1.0.2 of the LoRaWAN protocol in the ISM 902_928 bands. The results obtained have shown that the solution proposed works appropriately in urban environments complying with the characteristics of an LPWAN network, due to provides communication of long range and immediate notifications

Keywords: LPWAN, LoRaWAN, LoRa, Home Security, Intrusion detection

PRESENTACIÓN

El presente documento se encuentra estructurado de la siguiente manera:

El Capítulo I presenta el problema que la solución propuesta pretende contrarrestar, los objetivos del proyecto, su alcance, la metodología de trabajo utilizada para su desarrollo y una breve descripción de la zona residencial en donde el prototipo fue puesto a prueba.

El Capítulo II define los fundamentos teóricos del trabajo realizado, así como una breve revisión literaria relacionada al tema de seguridad del hogar y a la tecnología LoRa.

El Capítulo III presenta una descripción detallada de los componentes del prototipo de red LPWAN junto con el diseño de su arquitectura, el software utilizado para el desarrollo del proyecto y la aplicación del marco de trabajo ágil SCRUM.

En el Capítulo IV se presenta la implementación de la solución propuesta, detallando su desarrollo y haciendo uso de los temas expuestos en el Capítulo III.

El Capítulo V muestra las distintas pruebas efectuadas al prototipo y los resultados obtenidos que permitieron determinar la factibilidad de la solución.

Para finalizar, el Capítulo VI presenta las conclusiones y recomendaciones obtenidas al haber finalizado este trabajo, así como una posible implementación del mismo.

CAPÍTULO I

1. INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

Durante los últimos años, la inseguridad ciudadana se ha convertido en uno de los temas más importantes de tratar a nivel mundial. Entre los principales problemas de seguridad que enfrentan los ciudadanos, resaltan aquellos que involucran robos en sus hogares. Según una investigación realizada por la empresa norteamericana Safeatlast [1], solo en Estados Unidos existe un total de 2,5 millones de robos por año, de los que el 66% corresponde a asaltos a hogares. Además, también se menciona que las viviendas que carecen de un sistema de seguridad presentan un 30% más de posibilidad de ser invadidas por delincuentes. Por lo tanto, al expandir estas cifras a nivel mundial, el problema se transforma en un tema real de preocupación. Esto es corroborado por otro estudio, esta vez realizado en Quito por parte de Johanna Espín M, docente de la Facultad Latinoamericana de Ciencias Sociales (FLACSO) [2], que afirma que los delitos contra la vivienda constituyen uno de los mayores problemas de inseguridad que presenta la ciudad.

Actualmente existen sistemas robustos que permiten reforzar la seguridad en una vivienda, sin embargo, no todas las personas tienen acceso a estos por tema de costos, por lo que, llegar a implementarlos en todas las viviendas perteneciente a un determinado sector (barrio, cooperativa, recinto, etc.) resulta complicado. Debido a esto existe otro tipo de sistemas que, aunque no refuerzan en gran medida la seguridad de un hogar, permiten que otras personas (vecinos, guardias, policías, etc.) puedan percatarse que alguien del sector está sufriendo un asalto dentro de su vivienda, permitiéndoles acudir rápidamente a ayudar a la víctima de este delito.

Entre los sistemas que controlan la seguridad de las viviendas en algunas zonas residenciales del país se tienen: botones de seguridad o de pánico, alarmas comunitarias y sistemas de videovigilancia. No obstante, estos sistemas no se encuentran presentes en todo lugar e involucran en gran medida la intervención humana para activar las notificaciones de auxilio. También en muchos lugares existe la presencia de las Unidades de Policía Comunitaria (UPC) que refuerzan y controlan la seguridad del sitio, pero de igual manera no son eficientes ya que exige de la presencia, a toda hora y en todo lugar, del personal de seguridad, llevando de nuevo a lo citado anteriormente, a la necesidad de la presencia de un humano para realizar el control de la seguridad del lugar.

Como se menciona en [3], el asalto en hogares de cualquier sector de la ciudad está presente en todo momento, independientemente si existe un control comunitario o si las infraestructuras se encuentran protegidas, habitadas o no. Por tal motivo, para fortalecer y automatizar el control de intrusos en las viviendas de una determinada zona residencial, se pretende hacer un mejor uso de la tecnología, ya que como se ha observado en los últimos años esta poco a poco ha ido evolucionando hasta el punto de convertirse en algo indispensable en el diario vivir del ser humano, brindándole no solo entretenimiento sino soluciones a los problemas que se le presentan.

Por lo tanto, en este proyecto se plantea desarrollar un prototipo de red de área amplia de bajo consumo, LPWAN (en inglés, Low Power Wide Area Network), usando tecnología LoRa, mismo que será sometido a pruebas utilizando 2 tipos de nodos dentro del barrio Divino Niño ubicado al sur la ciudad de Quito. Actualmente, las redes LPWAN son ideales para solventar el tipo de problema mencionado ya que ofrecen: bajo consumo de energía, coste reducido, conexiones a grandes distancias (aproximadamente de 2 km a 5 km para zonas urbanas densas y de 15 km para zonas rurales) y gran solidez frente a las interferencias [4]. De esta forma, se monitoreará en tiempo real las viviendas de la zona residencial, de modo que notifique principalmente al personal de seguridad si se llega a detectar la presencia de algún intruso en cualquiera de las viviendas, permitiéndoles llegar inmediatamente al sitio a ayudar. La solución tendrá nodos implementados con tecnología LoRa colocados en los lugares más propensos a ser vulnerados. Estos nodos transmitirán los datos captados por los sensores a un Gateway LoRa, que a la vez los reenviará a un servidor LoRaWAN que se encargará de procesarlos para realizar la notificación oportuna en el caso de detectarse un intruso dentro del inmueble.

1.2. OBJETIVOS

1.2.1. Objetivo general

Desarrollar un prototipo de red LPWAN con tecnología LoRa para la detección de intrusos en las viviendas de una zona residencial.

1.2.2. Objetivos específicos

- Identificar las características relacionadas a las redes LPWAN y dispositivos LoRa.
- Analizar y definir los requerimientos necesarios para el diseño de una red LPWAN usando tecnología LoRa.
- Diseñar un prototipo de red LoRaWAN que por medio de nodos LoRa detecte intrusos en las viviendas de una zona residencial, y que permita enviar mensajes

de alerta tanto al personal que administra la aplicación web como a los usuarios que utilicen la aplicación móvil.

- Desarrollar una aplicación web que permita al personal de seguridad gestionar usuarios y monitorear el estado de los nodos.
- Desarrolla una aplicación móvil que permita al usuario dueño de la vivienda gestionar la activación y desactivación de los nodos.
- Validar el funcionamiento del prototipo red LPWAN, utilizando dos nodos con diferentes tipos de sensores, dentro del barrio Divino Niño de la ciudad de Quito.

1.3. ALCANCE

El presente proyecto incorporará dos tipos de sensores: magnético y movimiento, que permitirán detectar intrusos en una vivienda; una aplicación web, que permitirá al personal de seguridad recibir notificaciones y monitorear, a través de un mapa en tiempo real, las viviendas de la zona residencial; y una aplicación móvil, que permitirá al propietario del inmueble recibir notificaciones y gestionar la activación y desactivación de un dispositivo final LoRa. El prototipo contará con dos nodos y será puesto a prueba en el barrio Divino Niño de la ciudad de Quito.

La red permitirá emitir mensajes de estado de los nodos finales, esto quiere decir que, si el dispositivo LoRa tiene problemas de conexión con la red LoRaWAN, esto será reflejado dentro del mapa presente en el aplicativo web y notificado tanto al personal de seguridad como al dueño de la propiedad. Finalmente, la aplicación web incorporará un módulo para la creación de nuevos usuarios y la gestión confiable de los mismos.

1.4. METODOLOGÍA

Para la realización de este proyecto se utilizó una metodología de cuatro fases: investigación, diseño, implementación y pruebas. Las cuales permitieron desarrollar el prototipo de manera eficiente y ordenada con el fin de obtener los resultados deseados.

Fase de investigación

Esta fase consistió en realizar una investigación general acerca de las redes LPWAN, de la tecnología de comunicación inalámbrica LoRa y del protocolo LoRaWAN. Así como, efectuar una revisión y análisis de soluciones relacionadas a la detección de intrusos en hogares, mismas que permitieron tener un marco de referencia para sustentar la aplicación de las tecnologías planteadas en el prototipo. La finalidad de esta fase es tener el conocimiento necesario para determinar los elementos que formarán parte de la arquitectura de la red LPWAN y la metodología utilizada para su implementación.

Fase de diseño

Esta fase permitió diseñar la arquitectura de red LoRaWAN, definiendo a detalle cada elemento que la conforma junto con la razón por lo que ha sido escogido. Del mismo modo, en base a la aplicación del marco de trabajo ágil SCRUM, se recopilaron los requerimientos del proyecto y definieron los distintos Sprints que permitieron cumplir los objetivos propuestos y culminar su implementación dentro de un tiempo establecido.

Fase de implementación

Esta fase consistió en implementar, configurar e integrar los componentes hardware y software del prototipo, es decir, contiene la parte práctica del proyecto. Todo este proceso se lo realizó en base a las historias de usuario y Sprints establecidos previamente con el enfoque ágil Scrum.

Fase de pruebas

En esta fase se llevaron a cabo las pruebas que permitieron comprobar el funcionamiento de la red implementada. El propósito de esta fase fue determinar el rango de comunicación de los dispositivos, el tiempo que le toma al prototipo enviar un mensaje de detección de intrusos desde los dispositivos finales hacia el backend del sistema y el consumo de corriente de los nodos implementados. Todo esto con el afán de comprobar que tan factible es implementar este tipo de redes en un entorno urbano.

1.5. DESCRIPCIÓN DE LA ZONA RESIDENCIAL

El sistema de seguridad esta ideado para ser implementado en diferentes tipos de viviendas, tales como casas y departamentos. En este caso, la propuesta toma como referencia las residencias del barrio Divino Niño localizado al sur de la ciudad, ver Figura 1. Esta zona residencial se encuentra conformada por 36 manzanas. En cada manzana existen entre dos a seis bloques de tres pisos, mismos que contienen 2 departamentos por piso.

Figura 1. Ubicación del barrio Divino Niño

Los departamentos tienen 70 m^2, de los cuales 52 m^2 son de construcción y 18 m^2 que, dependiendo del caso, son para el patio y la lavandería (departamentos del primer piso) o para la terraza (departamentos del segundo y tercer piso). La construcción se encuentra distribuida en dos dormitorios con closet, sala, baño, comedor y cocina, ver Figura 2. Algo importante de destacar es que la estructura del departamento se encuentra construida con hormigón armado, cuyos dormitorios tienen piso flotante y su área social posee piso de cerámica.

Figura 2. Plano de la vivienda
Fuente [5]

CAPÍTULO II

2. REVISIÓN LITERARIA Y FUNDAMENTOS TEÓRICOS

2.1. TRABAJOS RELACIONADOS

Este apartado presenta una breve revisión de soluciones relacionadas tanto a seguridad para el hogar como a las tecnologías LoRa y LoRaWAN dentro del contexto de hogares inteligentes.

2.1.1. Estudios previos

La solución propuesta en [6], implementa un sistema de seguridad basado en las tecnologías Wi-Fi y GSM. Aquí se utiliza dos tipos de sensores que permiten detectar algún movimiento o incendio dentro del hogar. Para notificar dicho evento cada uno se conecta a una placa NodeMCU, que mediante Wi-Fi transporta el mensaje a un Raspberry Pi (master node). El envío de la notificación la realiza el módulo GSM que también se encuentra conectado al Raspberry. La notificación puede ser a través de un correo electrónico, un SMS o una llamada al teléfono de los usuarios previamente establecidos, es decir, al propietario del inmueble, a los bomberos, a la policía, etc. El usuario también tiene la posibilidad de acceder a los datos de los nodos y a la cámara del sistema a través de internet colocando la IP del Raspberry.

Por otro lado, el trabajo denominado Design and Implementation of a New Smart Home Control System Based on Internet of Things [7], describe el uso de una red Zigbee combinada con tecnología de red 3G / 4G para el monitoreo remoto del hogar. El sistema tiene varias aplicaciones que permiten garantizar la seguridad familiar dentro de la vivienda, entre ellas está la videovigilancia, implementando una cámara de video que permite al usuario monitorear el inmueble a través de su teléfono móvil o un aplicativo web.

Las soluciones [8] y [9] se enfocan en el uso de Bluetooth. En [8], se lo utiliza para establecer conexión con una PC, la cual actúa como un módulo de control principal, y que junto con Internet notifica, mediante SMS o correo electrónico, al dueño de la propiedad si alguien forcejea la puerta o ingresa a su vivienda. Para detectar dichas acciones se emplea dos tipos de sensores (vibración y movimiento). Otras funciones del sistema son: bloquear la puerta, encender una cámara y activar la alarma antirrobo. Ahora bien, en [9] se describe un sistema de hogar inteligente que utiliza servicios web basados en Bluetooth y REST como capa interoperable. Este sistema tiene características como autenticación de usuario mediante de una aplicación móvil, conectividad Bluetooth e Internet, control automatizado de electrodomésticos, sistema

de seguridad y vigilancia contra incendios e intrusos. Este último incorpora dispositivos que sirven para detectar un incendio o intrusión en el inmueble, si ha ocurrido alguno de estos dos eventos, el sistema activa una sirena y notifica al usuario por correo electrónico.

En cuanto a LoRa y LoRaWAN, se han encontrado soluciones que utilizan estas tecnologías para detectar fugas de gas y controlar la iluminación de una vivienda. Por ejemplo, en [10] se presenta una arquitectura LoRaWAN que utiliza un sensor MQ2 junto con una placa TTGO para detectar y notificar al gateway la presencia de gas LPG en el entorno. La comunicación entre la placa y el gateway se realiza mediante LoRa. Cabe destacar que en este proyecto se propone implementar a futuro un Cluster Controller, que se encargará de gestionar la información de los sensores para que los guardias de seguridad puedan monitorearlos. Otro trabajo [11], implementa una red LoRaWAN que gestiona un termostato y las luces de una vivienda. En esta solución se implementó un gateway utilizando una Raspberry Pi3 B y un shield LoRaGo PORT. El encendido y apagado tanto de las luces como del termostato se lo puede realizar mediante un aplicativo web, móvil o un asistente de voz. Además, se utiliza MQTT para publicar los mensajes en el servidor de aplicaciones usado por LoRaWAN.

2.1.2. Análisis

Los sistemas revisados anteriormente proponen varias formas de implementar una solución con respecto a la seguridad del hogar, enfatizando la detección de intrusos. Sin embargo, algunos inconvenientes que podrían llegar a presentar se relacionan con la tecnología utilizada para la comunicación y notificación de eventos detectados por los sensores, la usabilidad y la tranquilidad que brindan a los usuarios. Por ejemplo, en [6–9] utilizan Wi-Fi, Zigbee y Bluetooth como medio de comunicación entre nodos y elementos centrales, estos últimos, debido a la tecnología de corto alcance, tienen que ser ubicados en una zona específica, sin ningún tipo de obstáculo para que puedan establecer conexiones óptimas con el resto de los componentes.

Por otro lado, en [6,8] se especifica el uso de SMS como medio de notificación al propietario del inmueble. Implementar este tipo de tecnología podría llegar a ser costosa dependiendo de los mensajes que se envíen. Ahora bien, la solución [6] y [10] no presentan una interfaz gráfica que permita al usuario gestionar el estado de los nodos conectados al sistema.

Es importante señalar que ninguna solución cuenta con un aplicativo que brinde al personal encargado de la seguridad la posibilidad de monitorear el estado de los sensores que posee cada vivienda, ni notificarles directamente cuando una intrusión

ocurra dentro de la misma, por lo que no se asegura que alguien acudirá al sitio a verificar lo sucedido. Como último punto, en las soluciones [10] y [11] se puede evidenciar que la tecnología LoRa junto con LoRAWAN se adaptan fácilmente a escenarios relacionados al contexto de hogares inteligentes.

2.2. MARCO TEÓRICO

2.2.1. SMART CITY

Una Smart City es aquella ciudad que haciendo uso de las TIC (Tecnologías de la Información y Comunicación) mejora la calidad de vida de sus habitantes, ofreciéndoles sostenibilidad e innovación a largo plazo. Las Smart Cities impulsan el desarrollo urbano actual atendiendo las necesidades básicas inmersas en el ámbito social, económico y ambiental [12]. Entre las necesidades que cubren se tiene: mejorar el sistema de transporte público, ahorro energético, seguridad ciudadana, sostenibilidad, iniciativas ambientales, mejorar el turismo, tener infraestructura donde la tecnología sea la base, etc.

2.2.1.1 Elementos que componen una Smart City

Para que el modelo urbano de una Smart City funcione se debe precisar de diversos elementos que permitan, en conjunto, formar un sistema eficiente y sostenible capaz de solventar las necesidades presentes y futuras. Entre los principales componentes se tiene [13]:

- **Sensores** ubicados en lugares claves de la ciudad para la captura de datos.
- **Base de datos** que permitan recopilar y almacenar la gran cantidad de datos obtenidos de manera organizada.
- **Servidores** que se encarguen de gestionar la red, así como del procesamiento de los datos.
- **SIG (Sistemas de Información Geográfica)** útiles en la contextualización de los datos recopilados, unificando datos relevantes con la información espacial.
- **IoT (Internet of Things)** para interconectar tanto los sistemas como los dispositivos utilizados.

La interrelación de estos componentes permite obtener información adecuada que no solo ayuda a los gobernantes a tomar decisiones óptimas, sino que permite automatizar aún más los procesos del sistema de la ciudad, todo esto a favor del bienestar y de la prosperidad de sus habitantes.

2.2.1.2. Dimensiones de una Smart City

De acuerdo con el informe “Mapping Smart Cities in the EU” del Parlamento Europeo [14], presentado en enero del 2014, una Smart City puede ser clasificada o identificada como tal de acuerdo con 6 dimensiones o ejes principales, mismos que son detallados en la Figura 3.

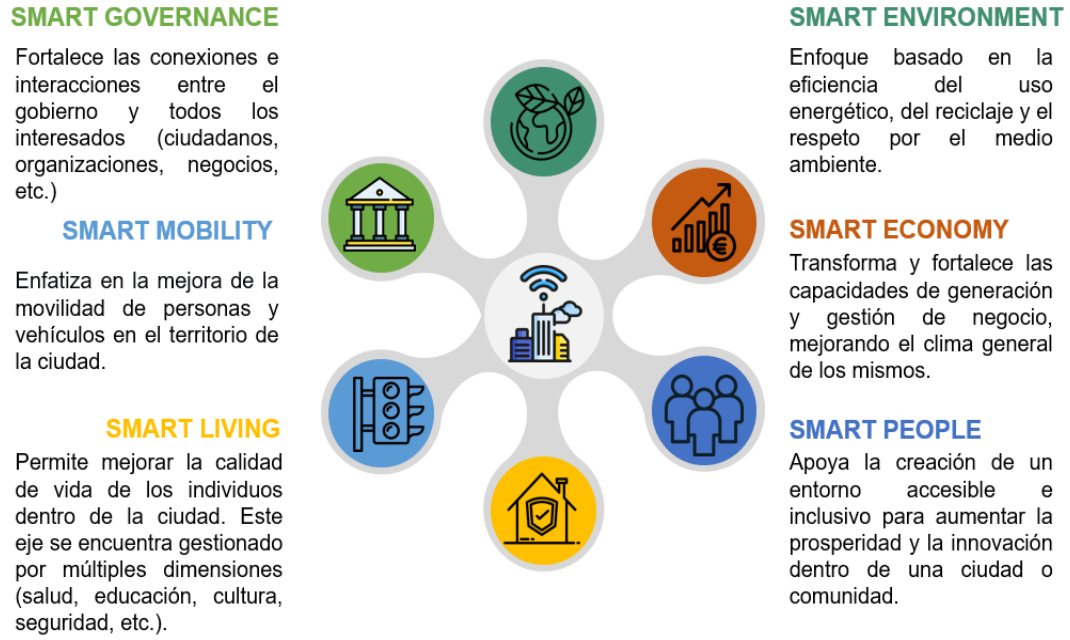


Figura 3. Dimensiones de una Smart City

2.2.2. INTERNET DE LAS COSAS

En el año de 1999 el británico Kevin Ashton, en una presentación hecha a Procter & Gamble (P&G), utiliza la frase “Internet de las cosas” (IoT) para describir una red que conecta, mediante el uso de sensores, los objetos presentes en el mundo físico a Internet [15]. Desde entonces el término se ha difundido de tal manera que en la actualidad se considera al IoT como una de las tecnologías más importantes de la vida cotidiana, que continuará ganando impulso a medida que las personas y empresas se percaten del potencial que tiene.

Actualmente no existe un concepto único de Internet de las cosas, pero en términos simples, se lo puede definir como una red de dispositivos físicos que conectados al Internet envían y reciben datos con una mínima participación humana. En otras palabras, IoT es una colección de dispositivos electrónicos que pueden compartir información entre ellos a través de la red. Incluso, en algunos casos estos dispositivos pueden “aprender” al examinar patrones presentes tanto en las preferencias del usuario como en el historial de sus datos [16].

La idea clave de IoT es hacer que no solo los dispositivos tradicionales se conecten a Internet sino también aquellos que quizá no se imagina que pudieran hacerlo, pero que al ser dotados de tecnología ganan la capacidad de comunicarse e interactuar dentro de una red. Normalmente, estos se les denomina dispositivos “inteligentes” porque recopilan datos mediante sensores, y ofrecen servicios en función del análisis de esa información y de acuerdo con los parámetros definidos por el usuario. Dentro de estos se tiene: medidores inteligentes (temperatura, humedad, presión, luz), wearables o tecnología vestible (pulseras, anillos, gafas, ropa), electrodomésticos (refrigeradoras, consolas, lavadoras), telemática vehicular, rastreadores de actividad física, entre otros.

2.2.2.1. Aplicaciones del IoT dentro de una Smart City

Las Smart Cities utilizan el IoT para recopilar y analizar datos en tiempo real, además, aprovechan sus características para monitorear, administrar y controlar dispositivos de forma remota. La información obtenida en este proceso permite tanto a gobernantes como a ciudadanos tomar decisiones, rápidamente, que permitan mejorar las condiciones de vida, la infraestructura y los servicios públicos dentro de la ciudad.

Entre algunos usos del IoT dentro de una Smart City se tiene [17]:

- **Gestión del tráfico:** los sensores integrados en el pavimento envían actualizaciones en tiempo real del flujo de tráfico a una plataforma central que analiza los datos y ajusta automáticamente los semáforos en segundos.
- **Estacionamiento inteligente:** los sensores en el suelo informan al conductor, a través de su teléfono inteligente, donde puede encontrar un espacio de estacionamiento disponible.
- **Gestión inteligente de residuos:** el contenedor de residuos recibe un sensor ultrasónico para medir la distancia de los desechos, cuando se alcanza un cierto umbral, la plataforma de gestión envía una notificación al teléfono inteligente de un conductor del camión recolector.
- **Iluminación inteligente:** en base a sensores de movimiento solo se debería dar luz cuando alguien realmente pasa junto a ellos, además, permite establecer niveles de brillo y rastrear el uso diario del consumo energía eléctrica.
- **Gestión de la seguridad:** El IoT también puede encontrar aplicaciones en el campo de la seguridad y la vigilancia, por ejemplo, vigilancia de espacios, seguimiento de personas y activos, seguridad de viviendas, mantenimiento de infraestructura y equipos, alarmas, etc.

2.2.2.2. Seguridad inteligente

Como se menciona en la anterior sección, el IoT tiene un uso significativo dentro del tema de seguridad inteligente, y es en el apartado de seguridad de las viviendas donde se aprecia una gran utilidad debido a que permite crear sistemas capaces de alertar robos e invasiones dentro de los inmuebles. Estos sistemas poseen dispositivos de seguridad interconectados que se adaptan a la infraestructura de la propiedad, dicha interconexión les permite trabajar conjuntamente frente a una situación de riesgo, permitiéndoles incluso transmitir, mediante la red, alertas importantes no solo al dispositivo móvil del propietario del inmueble sino también al personal de seguridad [18].

2.2.2.2.1. Tipos de dispositivos de seguridad

Para la construcción de un sistema seguridad inteligente se usan distintos tipos de dispositivos, entre los más comunes se tiene [18]:

- **Cámaras:** han evolucionado al grado de poseer tecnología capaz de ofrecer una visión óptima en sitios oscuros, incluso permiten al personal autorizado conectarse en la red para monitorear lo que sucede dentro de la vivienda.
- **Sensores**: los sensores magnéticos, de movimiento, entre otros se encargan de detectar algún inconveniente dentro del inmueble, además, si ese fuera el caso, tienen la capacidad de enviar una notificación al smartphone del propietario a través de la red.
- **Cerraduras:** pueden ser controladas para permitir tanto la apertura como el cierre de puertas y ventanas de la vivienda.
- **Alarmas:** son programadas para emitir alertas en situaciones de riesgo o para disuadir intrusos en el caso de que estos ingresen al inmueble.

2.2.2.3. Redes de comunicación IoT

Dentro del contexto del IoT, una red de comunicación conecta entre sí los distintos dispositivos que conforman un sistema IoT con la finalidad de que estos puedan intercambiar información a través del internet. El tamaño y el alcance de una red se pueden expresar por el área geográfica que ocupan y la cantidad de dispositivos que forman parte de esta. Es decir, pueden abarcar desde un puñado de dispositivos presentes en una sola habitación hasta millones de dispositivos repartidos por todo el mundo. Además, cada una de ellas poseen diversas tecnologías de comunicación inalámbrica que intentan cubrir las necesidades claves de la mayor parte aplicaciones IoT que son: largo alcance de comunicación, bajo consumo de energía, altas o bajas tasas de transferencia de datos dependiendo de la aplicación y bajo coste de implementación [19].

En la Figura 4, se presentan las distintas redes utilizadas dentro de IoT con ejemplos de las tecnologías de comunicación inalámbrica que cada una de ellas poseen. Cabe recalcar que estas se encuentran relacionadas en función de su alcance, velocidad y potencia.

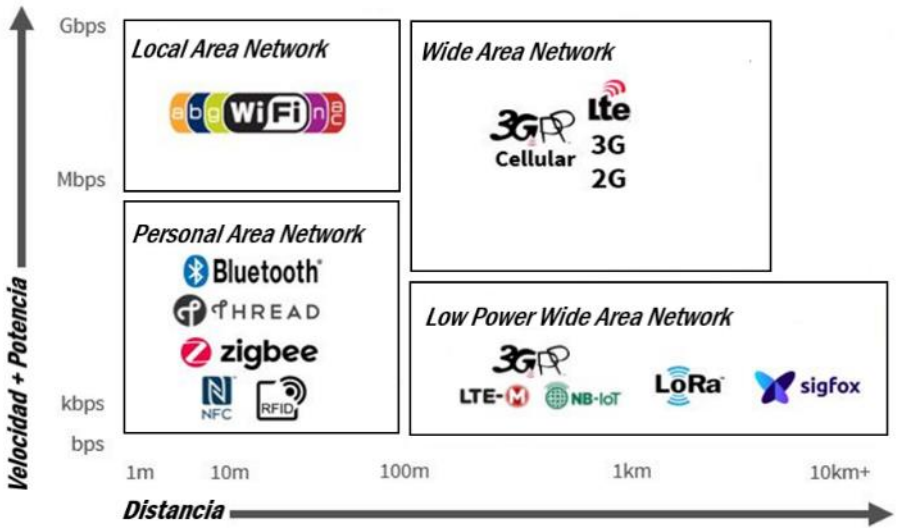


Figura 4. Comparación de redes de comunicación usadas en IoT

Fuente [20]

2.2.3. LPWAN

Las redes de área amplia de baja potencia (en inglés, Low-Power Wide Area Network), como su nombre lo indica, no solo permiten cubrir un área amplia, sino que también funcionan a un costo menor y con una mayor eficiencia energética que las redes tradicionales. Esta tecnología de comunicación surge para dar solución a problemas presentes en las tecnologías inalámbricas convencionales, tal es el caso de Wi-Fi, Bluetooth, Zigbee, etc., que, a pesar de ser soluciones de radio altamente utilizadas, no pueden ser aplicadas a situaciones que requieran transmisiones de largo alcance. Por otro lado, las tecnologías celulares 2G, 3G y 4G, aunque permiten cubrir grandes áreas, el excesivo consumo de energía es su principal desventaja [21].

Teóricamente, las tecnologías usadas en LPWAN permiten conectar dispositivos pequeños que pueden permanecer hasta 10 años funcionando con una sola carga de batería. Esto es idóneo para sensores remotos y otras aplicaciones que presentan requerimientos mínimos de datos, inclusive para aquellos que necesitan ser localizados bajo tierra o en interiores. Además, en ambientes adecuados, una sola estación base o puerta de enlace que se ejecute dentro de una LPWAN es capaz de proporcionar servicio a un área extensa, desde pocos kilómetros en zonas urbanas a más de 15 kilómetros en zonas rurales [22].

2.2.3.1. Características de LPWAN

Entre las principales características de LPWAN se tiene [22]:

- **Baja potencia:** posee componentes optimizados para el consumo de energía. Los transceptores LPWAN pueden trabajar con baterías con una vida útil de hasta 10 años, lo que reduce en gran parte costos relacionados a mantenimiento.
- **Largo alcance:** el alcance funcional teórico de LPWAN va desde unos pocos kilómetros (1 – 10 km) en zonas urbanas hasta más de 15 km en entornos rurales. Además, permite la comunicación de datos de forma efectiva en lugares subterráneos y dentro de edificios.
- **Baja tasa de datos:** admiten transferencias de datos pequeñas (10 a 10.000 bytes de datos a una velocidad de hasta 1 Mbps), lo que las hace perfecto para sensores remotos y otras aplicaciones con demandas mínimas de datos.
- **Bajo costo**: los protocolos de LPWAN reducen la complejidad en el diseño del hardware lo que reduce el costo de fabricación del dispositivo. Por otra parte, el largo alcance junto con la topología en estrella proporciona una reducción de costos de infraestructura. Adicionalmente, usar bandas sin licencia y con licencia propia, permite reducir costos de red.

Estas características convierten a LPWAN en un tipo de red adecuada para cubrir muchas de las necesidades presentes en las aplicaciones IoT, mismas que fueron descritas en secciones anteriores.

2.2.3.2. Tecnologías LPWAN

No todas las LPWAN se crean de la misma manera. Hay una serie de tecnologías comparables que operan tanto en partes del espectro con licencia como sin licencia. Si bien esta variedad es buena para los diseñadores de productos, ahora es fundamental considerar cuidadosamente las opciones. Con la tecnología adecuada, los dispositivos LPWAN podrán conectarse sin problemas y de manera eficiente.

Según una investigación realizada al mercado LPWAN por IoT Analytics en el 2019 [23], existen cuatro tecnologías que representan aproximadamente el 92% de la base global instalada de dispositivos conectados a LPWAN. Por tal razón, se considera conveniente detallar a cada una de ellas en el siguiente apartado.

2.2.3.2.1 Sigfox

Sigfox es una red LPWAN propietaria desarrollada en 2010 por la empresa Sigfox (en Toulouse, Francia). Actualmente, se trata de una de las redes LPWAN más grandes del mundo y opera en las frecuencias sin licencia en las bandas ISM (433 MHz en Asia, 868

MHz en Europa y 902 MHz en América). Según esta empresa, sus redes IoT ahora se encuentran presentes en 70 países [24], con una importante presencia en Europa. El modelo de negocio de Sigfox se basa en suscripciones anuales pagadas por los clientes para conectarse a su servicio. El costo actual de la conectividad depende de dos factores principales: la cantidad de mensajes que se debe enviar todos los días y el número de nodos que se desea conectar. Según su sitio oficial, en Estados Unidos el costo de suscripción anual por dispositivo es de $8 por el envío de 2 mensajes al día y de $18.50 por transmitir 140 mensajes diarios [25].

La red de SigFox puede entregar mensajes a distancias de 30 a 50 km en áreas rurales y de 3 a 10 km en entornos urbanos. Esta red trabaja con mensajes de enlace ascendente livianos (12 bytes, excluyendo encabezados de carga útil), que, como se indicó, van desde 2 hasta 140 por día. Sin embargo, el número de mensajes a través del enlace descendente está limitado a 4 diarios. El ciclo de vida de un mensaje de Sigfox es el siguiente [26]:

1. Un dispositivo se despierta y envía un mensaje mediante su antena de radio, para ello se debe tener un módulo compatible con la red de Sigfox.
2. Múltiples estaciones base Sigfox en el área receptan el mensaje.
3. Las estaciones base reenvían el mensaje a la nube de Sigfox.
4. Sigfox Cloud se encarga de enviar el mensaje a la plataforma de backend del cliente.

La Figura 5, presenta el ciclo de vida expuesto.

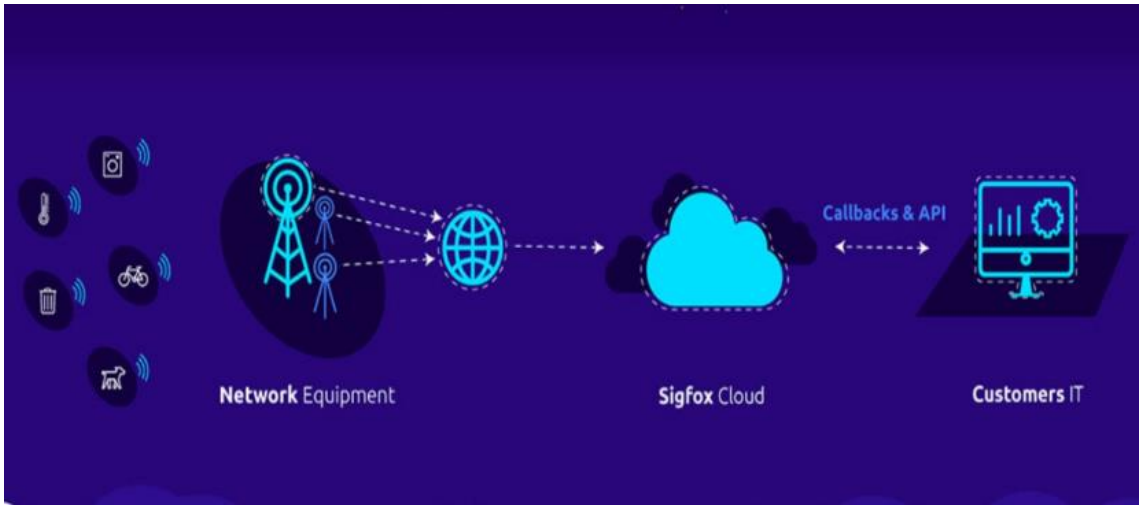


Figura 5. Transmisión de un mensaje en la red de Sigfox

Fuente [26]

Como se puede observar Sigfox posee una arquitectura cerrada, es decir, el control de la red es completamente de la empresa, lo que puede llegar a ser un inconveniente para las personas que deseen realizar una implementación propia, además, estas necesitan que la infraestructura de Sigfox esté presente en el territorio para poder contratar el servicio.

2.2.3.2.2. NB-IoT (Narrowband IoT)

Estandarizado por 3GPP (3rd Generation Partnership Project), NB-IoT es una tecnología LPWAN que minimiza el consumo de energía de los dispositivos conectados, al tiempo que aumenta la capacidad del sistema y la eficiencia espectral. Además, posee altos niveles de penetración, lo que la hace llamativa especialmente en lugares que no pueden ser cubiertos fácilmente por las tecnologías celulares convencionales [27].

NB-IoT puede coexistir con GSM (The Global System for Mobile Communications) y LTE (Long Term Evolution) bajo bandas de frecuencia con licencia (por ejemplo, 700, 800 y 900 MHz), ocupando un ancho de banda de frecuencia de 200 KHz. La velocidad de datos está limitada de 20 a 250 Kbps para el enlace ascendente y a 200 Kbps para el enlace descendente. El tamaño máximo de carga útil para cada mensaje es de 1600 bytes y la batería de un dispositivo puede alcanzar 10 años de vida al transmitir 200 bytes por día en promedio [21].

Muchos operadores diferentes han puesto a prueba la tecnología NB-IoT en una amplia gama de casos de uso, como medición, estacionamiento y agricultura inteligente, implementando estos servicios en Asia, América y Europa. Según GSMA (Asociación GSM), NB-IoT cuenta con 92 redes en todo el mundo [28]. Actualmente, el costo de un plan NB-IoT de 12 MB de datos en el operador T-Mobile es de $6 por dispositivo al año [29]. Un factor diferencial de NB-IoT es que, al trabajar en redes celulares públicas a través del espectro de radio con licencia, tanto su despliegue como su explotación a nivel comercial se encuentran casi asegurados debido a que hacen uso de la infraestructura del operador de red móvil existente. Sin embargo, si el operador presente en el país aún no ofrece este servicio no existiría la posibilidad de que se pueda acceder al mismo.

2.2.3.2.3. LTE-M (Long Term Evolution category M1)

LTE-M es un tipo de tecnología LPWAN desarrollada por 3GPP. Esta tecnología permite tener altas velocidades de datos y comunicaciones de baja latencia, lo que podría ser crucial si un usuario necesita obtener una respuesta inmediata de la estación base, recolectar archivos grandes de sus dispositivos IoT o enviar actualizaciones a los nodos finales. Con el fin de acatar el requerimiento de “baja potencia” de una red LPWAN, LTE-

M utiliza chips 4G que son menos costosos de fabricar y que cuentan con un modo especial de economizar la batería del dispositivo. Esto permite que los dispositivos pueden entrar en un modo de “sueño profundo” llamado PSM (Power Saving Mode) y despertarse periódicamente mientras están conectados [30].

LTE-M ofrece una velocidad de datos de 1Mbps tanto para enlace ascendente como para el descendente usando la versión 13 de 3GPP, limitando el ancho de banda máximo del sistema a 1,4 MHz, debido a esto la tecnología admite funciones que involucran voz. Al ser compatible con las redes LTE existentes, LTE-M no necesita que los operadores construyan una nueva infraestructura para implementarla [31]. Es decir, utiliza las torres celulares presentes en el país para transportar los datos. Esto también genera algunas desventajas ya que la red presentará los mismos puntos ciegos que tengan las redes celulares, además si la conexión de estas se interrumpe por mantenimiento, límite de suscripción, emergencias o alguna otra razón, el servicio LTE-M también sufrirá interrupciones.

Según GSMA, LTE-M cuenta con 42 redes en todo el mundo [28]. Actualmente, el costo de los planes ofrecidos por el operador AT&T Mobility van desde $1 dólar al mes por 500KB de datos hasta $30 dólares al año por datos ilimitados [32]. Una de las razones por las que LTE-M es una opción de conectividad más costosa es porque varios jugadores grandes tienen patentes sobre las tecnologías subyacentes, por tal motivo los usuarios deben pagar regalías a estas empresas por el uso de su propiedad intelectual.

La principal diferencia que tiene LTE-M con respecto a NB-IoT es la forma en cómo se comunican los datos cuando los dispositivos están en movimiento. Por ejemplo, si un dispositivo se mueve, cruzando celdas de redes diferentes, el dispositivo LTE-M no interrumpirá la conexión debido a que se comporta de igual manera que un celular, conectándose de torre a torre a medida que se mueve. Por otro lado, un dispositivo NB-IoT no transfiere la conexión, por lo que debe establecer una nueva cada vez que detecta otra torre [33].

2.2.3.2.4 LoRaWAN

LoRaWAN es una especificación de red LPWAN desarrollada y mantenida por LoRa Alliance, una asociación abierta creada en el 2015, y que hasta el momento consta con más de 500 compañías comprometidas a implementar a gran escala redes LPWAN, mediante el uso del estándar abierto LoRaWAN. Dentro de estos miembros se encuentran compañías reconocidas como: CISCO, HP, IBM, Semtech, entre otras [34]. Según LoRa Alliance, las implementaciones de LoRaWAN (incluidas públicas, privadas y de comunidad abierta) ahora están presente en 157 países [35].

LoRaWAN permite que dispositivos de baja potencia puedan comunicarse con aplicaciones conectadas a Internet mediante conexiones inalámbricas de largo alcance. Esta especificación se encarga de definir tanto el protocolo de comunicación MAC (Media Access Control) como la arquitectura del sistema para la red [36]. Además, utiliza la modulación LoRa (Long Range) de Semtech en bandas ISM sin licencia para permitir el enlace de comunicación de largo alcance entre dispositivos finales y puertas de enlace. La Figura 6, presenta las principales características de LoRaWAN.

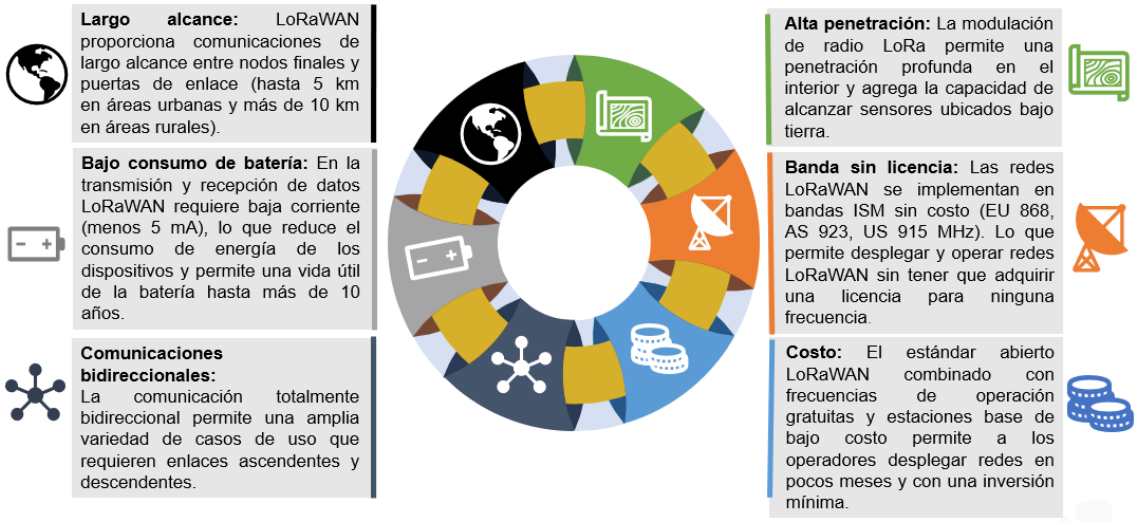


Figura 6. Características de LoRaWAN

2.2.3.2.4.1. LoRa

LoRa una modulación de radiofrecuencia desarrollada por Cycleo, y adquirida en el 2012 por Semtech, un reconocido fabricante de semiconductores y fundador de la LoRa Alliance [37]. Aunque Semtech posee la patente y se encarga de la elaboración de módulos LoRa, también ha concedido licencias a empresas como: Microchip, STMicroelectronics y HopeRF para que puedan hacer uso de esta tecnología y la incorporen en sus módulos. La patente proporciona información útil acerca de la capa física, especialmente de la modulación utilizada por LoRa que se basa en una dispersión del espectro.

LoRa está diseñada para utilizar todo el ancho de banda del canal para la transmisión, lo que le permite ser más robusta a las interferencias ocasionadas por otras frecuencias o por el ruido. LoRa es conocido por mejorar la sensibilidad del receptor y tener un mayor rango de comunicación para transmitir datos, cubriendo hasta unos 20 km en buenas condiciones. Esto la hace una tecnología perfecta para soluciones de redes en áreas amplias o rurales [38]. Mientras que otras tecnologías de IoT son propietarias, la gestión de la red LoRa es abierta, por lo que cualquier persona tiene la posibilidad de implementar sus propias redes y ofrecer servicios siempre que se respete las

regulaciones de uso del espectro. Las capas superiores de LoRa pueden ser patentadas o estandarizadas. El estándar más popular es LoRaWAN.

LoRa es una implementación de capa física (PHY), o de “bits”, según lo definido por el modelo de red de siete capas OSI. En lugar de cable, se usa el aire como medio para llevar las ondas de radio LoRa desde un dispositivo IoT a un gateway (enlace ascendente) y viceversa (enlace descendente) [39]. En la Figura 5, se observa como LoRa se encuentra a nivel de capa física empleando su modulación en las bandas ISM regionales. De la misma manera, se aprecia al estándar LoRaWAN haciéndose cargo de la capa MAC mediante la implementación de tres clases de dispositivos.

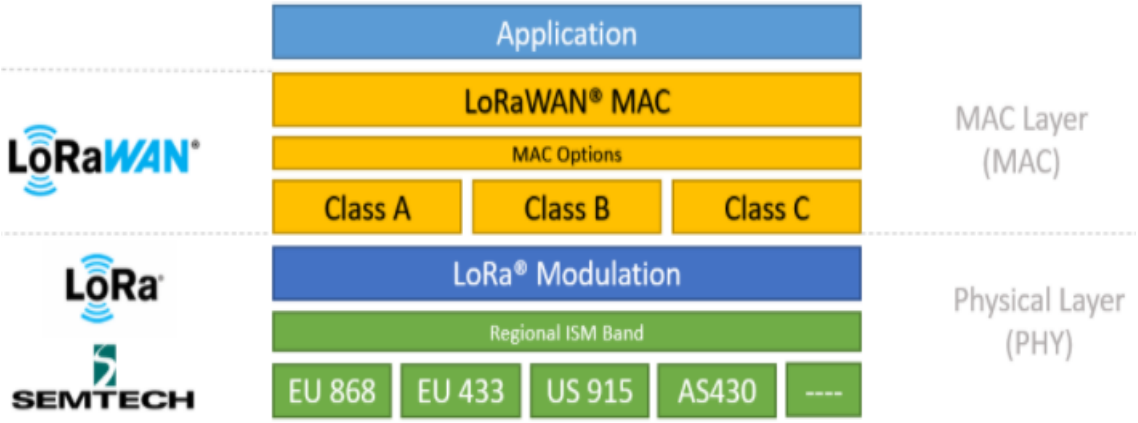


Figura 7. Pila de la tecnología LoRaWAN

Fuente [39]

2.2.3.2.4.1.1 Banda ISM 902-928

LoRa opera en bandas industriales, científicas y médicas sobre las frecuencias 433 MHz en Asia, 868 MHz en Europa y 915 MHz en América, utilizando tres rangos de ancho de banda: 125 KHz, 250 KHz y 500 KHz. El uso de estos rangos depende de la región o del plan de frecuencia que se aplique en el país. LoRa Alliance emite un documento de Parámetros Regionales [40], que contiene los planes de canales aprobados para varias regiones globales, y que sigue las restricciones regulatorias establecidas en cada región. La Unión Internacional de Telecomunicaciones (UIT), divide al mundo en tres Regiones para atribuir las bandas de frecuencia del espectro radioeléctrico, tal como se muestra en la Figura 8. Según este organismo los países de América pertenecen a la Región 2, por tal razón, en Ecuador se hace uso de la banda ISM de 902-928 MHz (con frecuencia central de 915 MHz) para trabajar con LoRa.

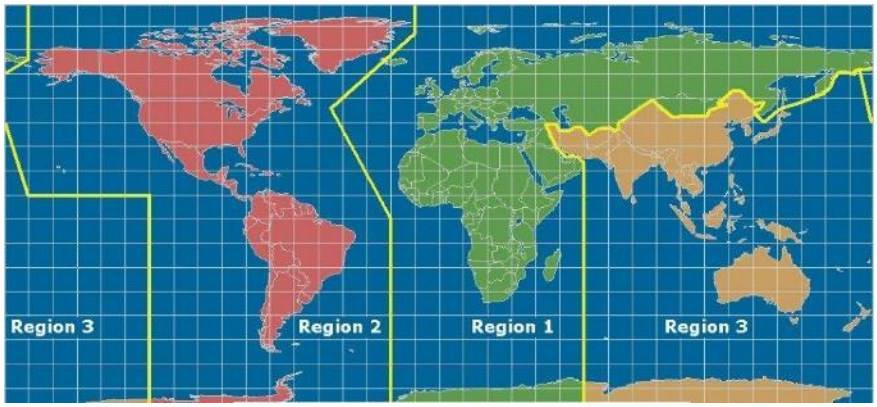


Figura 8. Regiones según la Unión Internacional de Telecomunicaciones

Fuente [41]

De acuerdo con el documento de Parámetros Regionales, los dispositivos LoRa que trabajan en la banda ISM US902-928 deben usar un canal de ancho de banda fijo de 125 KHz o 500 KHz en enlaces ascendentes, y de 500 KHz en enlaces descendentes. La banda ISM de 915 MHz presenta el siguiente plan de canales [40]:

- **Upstream 125 KHz:** 64 canales (0 al 63), utilizando un ancho de banda de 125 KHz, variando de DR0 (Data Rate) a DR3, comenzando en 902.3 MHz (canal 0) e incrementándose en 200 KHz hasta llegar a 914.9 MHz (canal 63).
- **Upstream 500 KHz:** 8 canales (64 al 71), utilizando un ancho de banda de 500 KHz en DR4, comenzando en 903.0 MHz (canal 64) e incrementándose en 1.6 MHz hasta llegar a 914.2 MHz (canal 71).
- **Downstream 500 KHz:** 8 canales (0 al 7), utilizando un ancho de banda de 500 KHz, variando de DR8 a DR13, comenzando en 923.3 MHz (canal 0) e incrementándose en 600 MHz hasta llegar a 927.5 MHz (canal 7).

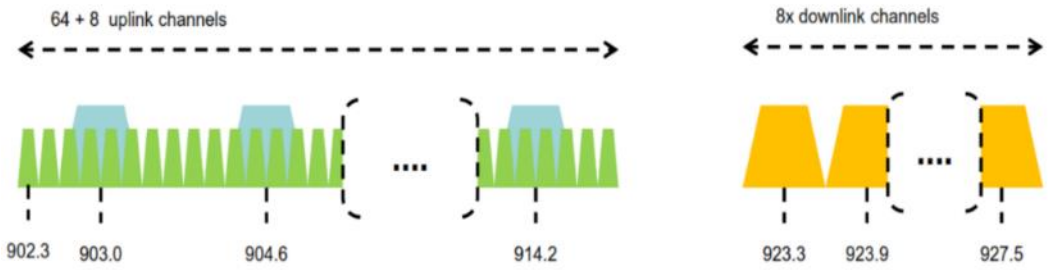


Figura 9. Frecuencias del canal US902-928

Fuente [40]

La Tabla 1 presenta otra manera de entender las características del plan de frecuencias mencionado.

Data Rate (DR)	Spreading Factor (SF)	Frecuencia del canal	Enlace ascendente y descendente	Tasa de bit (bits/s)	Tamaño máximo de carga útil del usuario (Bytes)
0	SF10	125KHz	E. ascendente	980	11
1	SF9	125KHz	E. ascendente	1760	53
2	SF8	125KHz	E. ascendente	3125	125
3	SF7	125KHz	E. ascendente	5470	242
4	SF8	500KHz	E. ascendente	12500	242
5 – 7					
8	SF12	500KHz	E. descendente	980	53
9	SF11	500KHz	E. descendente	1760	129
10	SF10	500KHz	E. descendente	3125	242
11	SF9	500KHz	E. descendente	5470	242
12	SF8	500KHz	E. descendente	12500	242
13	SF8	500KHz	E. descendente	21900	242

Tabla 1. Características de modulación LoRa en la banda ISM US902-928

Fuente [39]

2.2.3.2.4.1.2. Chirp Spread Spectrum (CSS)

Chirp Spread Spectrum es una tecnología ampliamente utilizada para el sonar en la industria marítima y el radar en la aviación. LoRa adopta la técnica CSS, basada en pulsos de chirp modulados en frecuencia lineal, para codificar la información. Un chirp, a menudo llamado señal de barrido, es una onda sinusoidal que aumenta o disminuye linealmente en frecuencia a lo largo del tiempo [42].

En la Figura 10, se puede apreciar la forma de onda de un chirp lineal ascendente y descendente, así como la evolución de frecuencia de cada uno de ellos a través del tiempo.

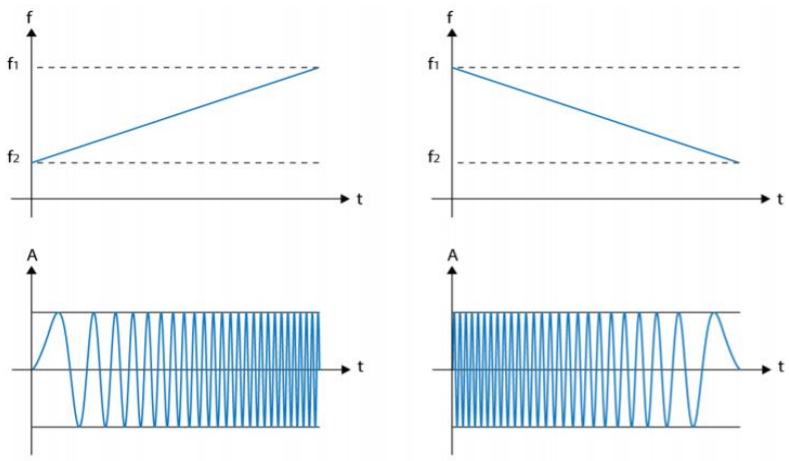


Figura 10. Forma de onda y evolución de frecuencia de un chirp ascendente y descendente

Fuente [43]

LoRa utiliza los chirps como señales portadoras para codificar el mensaje, esto puede ser apreciado en la Figura 11, en la que, mediante el uso de un espectrograma, se logra observar un mensaje modulado con LoRa.

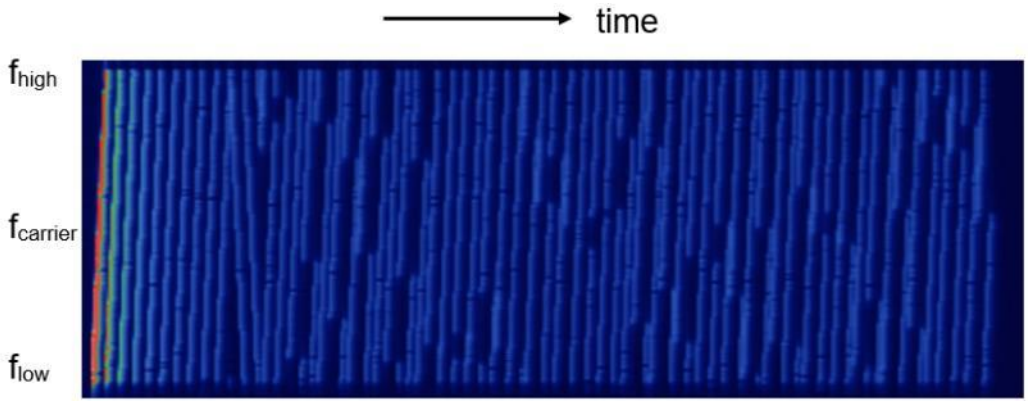


Figura 11. Mensaje codificado en señales de chirp

Fuente [42]

En la figura anterior se aprecia cómo cada chirp va variando linealmente su frecuencia con el transcurso del tiempo. Si bien al inicio de la transmisión se observa uniformidad en la señal (preámbulo), después esta empieza a presentar ciertos desplazamientos o saltos, es justo aquí donde la información de cada símbolo transmitido se encuentra codificada. Para entender con más detalle lo mencionado, se hace uso de la Figura 12, la cual muestra el proceso de demodulación de una señal LoRa.

Figura 12. Demodulación de una señal LoRa

Fuente [44]

El dispositivo receptor se encarga de generar un chirp inverso para multiplicarlo con la señal modulada recibida. Como resultado de este proceso se obtiene los símbolos decodificados. Cada desplazamiento existente en la señal entrante es la manera de identificar que se ha trasmitido un símbolo diferente.

2.2.3.2.4.1.3. Spreading Factor (SF)

El spreading factor corresponde a la cantidad de código de dispersión aplicada a una señal de datos original. La modulación LoRa define un total de seis factores de dispersión (SF7 a SF12) para controlar la tasa de bits, mejorar el rango de comunicación

y disminuir el consumo de batería. Un incremento en el factor de dispersión seleccionado duplicará el tiempo del símbolo y disminuirá la sensibilidad del receptor en un rango aproximado de 3 Db [39]. Es decir, cuanto mayor sea el spreading factor utilizado, más lejos podrá viajar la señal y ser recibida sin errores por el receptor de radiofrecuencia, esto a expensas de una menor velocidad de datos y un mayor consumo de energía.

Una descripción general de la duración de un símbolo con respecto a diferentes factores de difusión puede ser apreciada en la Figura 13.

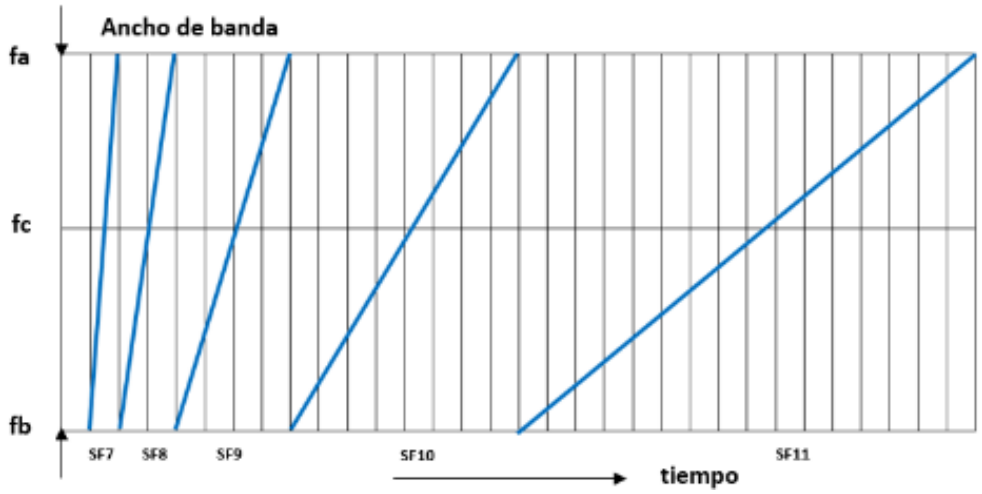


Figura 13. Relación entre el spreading factor y la duración del símbolo

Fuente [42]

La Tabla 2 muestra los seis factores de dispersión usados por LoRa en un canal de 125 KHz con una carga útil de 10 bytes.

Spreading Factor	Chips / symbol	Límite SNR	Tiempo en el aire (ToA)	Tasa de bit	Sensibilidad (dBm)
7	128	-7,5	56 ms	5469 bps	-123
8	256	-10	103 ms	3125 bps	-126
9	512	-12,5	205 ms	1758 bps	-129
10	1024	-15	371 ms	977 bps	-132
11	2048	-17,5	741 ms	537 bps	-134,5
12	4096	-20	1483 ms	293 bps	-137

Tabla 2. Factores de dispersión LoRa

Fuente [44]

Para determinar la tasa de bits (Rb) se hace uso de la siguiente fórmula [42]:

$$Rb\left(\frac{bits}{seg}\right) = SF \times \frac{BW}{2^{SF}} \times \frac{4}{(4+CR)}$$

Donde:

Rb = Velocidad de bits (bits/seg),

SF = Spreading factor (7 – 12),

BW = Ancho de banda (125 KHz, 250 KHz o 500 KHz),

CR = Tasa de codificación (1 – 4)

2.2.3.2.4.2. Arquitectura

Una red LoRaWAN normalmente se encuentra implementada en una topología de tipo estrella. Los elementos claves que conforman su arquitectura son: nodos finales, puertas de enlaces, un servidor de red y un servidor de aplicación [45]. La Figura 14 presenta la interacción de estos componentes.

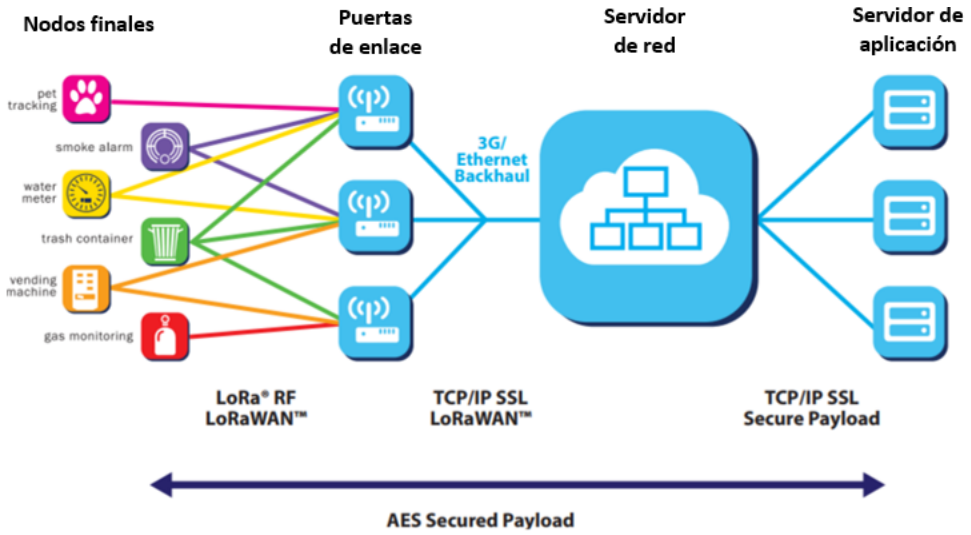


Figura 14. Arquitectura de la red LoRaWAN

Fuente [36]

El funcionamiento básico de la red LoRaWAN es el siguiente:

1. Un nodo final, provisto de sensores, envía un mensaje a una o varias puertas de enlace mediante una comunicación inalámbrica LoRa.
2. Las puertas de enlace, que se encuentren dentro del alcance del nodo, reciben el mensaje y mediante una conexión IP lo reenvían hacia el servidor de red.
3. El servidor de red recepta los paquetes de las puertas de enlace y gestiona la deduplicación de los mismos. Además, se encarga de determinar que puerta de enlace posee el mejor indicador de intensidad de señal recibida (RSSI).
4. Una vez filtrado el paquete, el servidor de red lo envía al servidor de aplicaciones adecuado, quien se hace cargo de preparar los datos para que el usuario final pueda usarlos en su implementación.

5. De manera opcional, el servidor de aplicación podría requerir enviar una respuesta al nodo que emitió el mensaje. Para ello, el servidor de red utilizará la puerta de enlace escogida en el punto 3.

2.2.3.2.4.2.1. Nodo final

Un nodo final es un dispositivo físico operado por batería que, haciendo uso de la comunicación inalámbrica LoRa, se conecta a la red LoRaWAN a través de una o más puertas de enlace. Generalmente estos dispositivos presentan capacidades de detección, procesamiento y modulación [45]. Entre los elementos que lo conforman se tiene: uno o varios sensores, que captan los eventos o estímulos del entorno; un microcontrolador (por ejemplo, tensilica LX6), usado para procesar los datos obtenidos del sensor; un módulo de radio LoRa (por ejemplo, SX1278), encargado de codificar la información usando los pulsos de chirp; y una antena, para enviar el mensaje inalámbricamente.

Al momento de fabricar un nodo, a este se le asigna un identificador único, el cual es utilizado para la activación y administración del dispositivo dentro de la red [39]. Normalmente se construyen utilizando un shell de radio LoRa y un microcontrolador. Actualmente, existen en el mercado placas de desarrollo LoRa, en las que el microcontrolador y el módulo ya se encuentran integrados en la placa de circuito impreso.

Algo importante a tomar en consideración es que LoRaWAN, dentro de su especificación [46], presenta tres clases de dispositivos finales, las cuales han sido creadas para resolver diferentes necesidades presentes en una variedad de aplicaciones. El funcionamiento de cada una de ellas puede ser visualizado en la Figura 15.

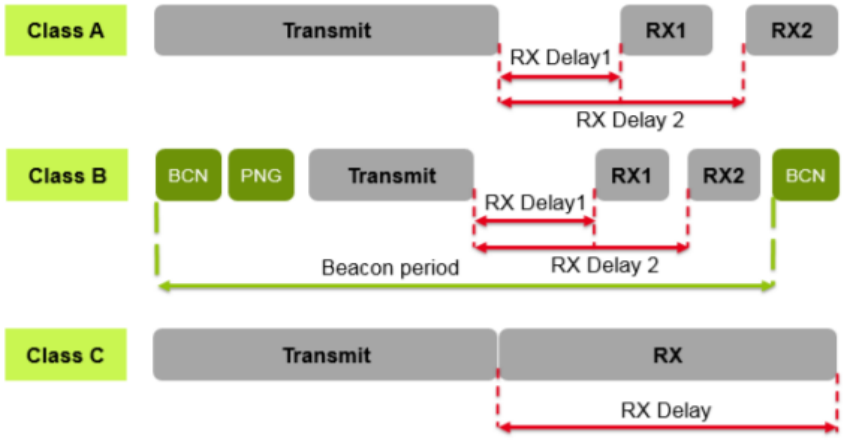


Figura 15. Clases de dispositivos LoRaWAN

Fuente [47]

2.2.3.2.4.2.1.1. Dispositivos de Clase A

Son los dispositivos de menor consumo energético debido a que la mayor parte de tiempo permanecen inactivos; solamente se despiertan cuando necesitan informar algo al servidor. Todos los dispositivos finales deben implementar las características de esta clase, la cual es útil en situaciones donde se necesita realizar un envío de datos en intervalos de tiempo determinados [45], pero no en aquellas donde se requiere que los nodos funcionen como actuadores.

Se debe tomar en cuenta que aquí el nodo final es el único que puede iniciar la comunicación dentro de la red. Después de transmitir el mensaje de enlace ascendente, el nodo abre dos ventanas, cada una con diferente tiempo de retraso, para recibir datos del servidor; brindando de esta manera una comunicación bidireccional. Si la respuesta es recibida en la primera ventana, el dispositivo ya no abrirá la segunda [46]. En caso de que el servidor no responda en estas ventanas, este deberá esperar hasta la siguiente transmisión de enlace ascendente para enviar algún dato al nodo.

2.2.3.2.4.2.1.2. Dispositivos de Clase B

Aparte de las dos ventanas de recepción abiertas por el modo de operación de clase A, estos dispositivos abren otras adicionales en horas establecidas. Esto, además de brindar al nodo una mayor capacidad para enlaces descendentes, genera un mayor consumo energético en su batería. La comunicación funciona al emplear un proceso denominado balizamiento, el cual consiste en el envío, por parte de las puertas de enlace, de beacons o balizas sincronizadas en el tiempo hacia los nodos finales, con el afán de permitirle al servidor conocer en qué momento el dispositivo final se encuentra activo [39]. Es así como, se evita tener que esperar a que el nodo envié una transmisión de enlace ascendente para conocer el momento en el que puede recibir datos del servidor.

Algo importante de señalar aquí, es que un dispositivo de clase B ya cuenta con la capacidad de ejercer el rol de actuador dentro de la red.

2.2.3.2.4.2.1.3. Dispositivos de Clase C

Un nodo de clase C casi siempre se encuentran escuchando trasmisiones de enlace descendente, únicamente cierran la ventana de recepción cuando necesitan enviar un nuevo mensaje. Estos dispositivos abren dos ventanas, tal como lo hacen los de clase A, solo que en esta ocasión la segunda ventana permanece abierta indefinidamente para receptar mensajes del servidor. A diferencia de las anteriores clases, un dispositivo de clase C ofrece una menor latencia a coste de un mayor consumo de energía [46]. Razón por la que es recomendable usarlos en lugares donde se tenga acceso continúo

a una conexión eléctrica; evitando así problemas referentes al uso de baterías cuando el dispositivo se encuentren en funcionamiento.

Los dispositivos de clase C son muy útiles en aplicaciones como: control de iluminación, seguimiento de automóviles, control industrial, etc. Al igual que un dispositivo de clase B, estos también pueden funcionar como actuadores dentro de la red.

2.2.3.2.4.2.2. Puertas de enlace

Las puertas de enlace son los dispositivos encargados de recibir los mensajes LoRa de todos los nodos finales, que se encuentren dentro de su rango, y reenviarlos hacia el servidor de red. Estas deben estar conectadas a Internet, bien sea por Ethernet, Wi-Fi o celular, para poder enviar los mensajes al servidor LoRaWAN. Una puerta de enlace usa un chip transceptor (SX1301, generalmente) para decodificar la señal modulada y extraer la información, permitiéndole incluso, escuchar múltiples frecuencias en diferentes spreading factors simultáneamente [42]. Dependiendo del tipo, pueden usarse tanto en ambientes internos (indoor) como al aire libre (outdoor).

2.2.3.2.4.2.3. Servidor de red

El servidor de red LoRaWAN es el responsable de administrar toda la red, encargándose de la deduplicación y gestión de los paquetes recibidos de las puertas de enlace. Esta gestión permitirá reenviar los datos a la aplicación adecuada. A más de las funciones descritas, un servidor de red LoRaWAN también es responsable de [39]:

- Controlar la tasa de transmisión de datos de cada dispositivo final.
- Autenticar cada nodo dentro de la red y verificar la integridad de los paquetes recibidos.
- Reenviar los mensajes provenientes del servidor de aplicación a los nodos finales pertinentes.
- Determinar la puerta enlace adecuada para establecer un enlace descendente hacia un nodo especifico.
- Responder cualquier solicitud de capa MAC realizada por un nodo.
- Reenviar los mensajes que intercambian el servidor de unión con los nodos finales en el proceso de unión de un dispositivo a la red LoRaWAN.

2.2.3.2.4.2.4. Servidor de aplicación

El servidor de aplicación LoRaWAN es el responsable de gestionar e interpretar la carga útil enviada por los distintos nodos que conforman la red. Este interactúa directamente con el servidor de red para recibir los paquetes provenientes de los dispositivos finales registrados en una aplicación especifica [45]. Si se desea enviar un mensaje de enlace descendente hacia un nodo determinado, es el servidor de aplicación que, dependiendo

del caso, se encarga de generar y encriptar la carga útil o payload necesario a ser enviado [39]. Se debe tomar en cuenta que los usuarios finales deben hacer uso de los servicios que ofrezca este servidor para integrarlo a sus soluciones; permitiéndoles así, utilizar los datos que el servidor gestiona.

2.2.3.2.4.3. Seguridad

LoRaWAN establece dos factores claves para la seguridad de su red [48]:

- Procedimiento de unión, que permite tanto al nodo como a la red LoRaWAN autenticarse mutuamente. Esto garantiza que únicamente los nodos autorizados puedan unirse a una red autorizada.
- Autenticación de mensajes de aplicación y MAC, estos mensajes se: autentican en origen, protegen por integridad y encriptan desde el nodo hacia el servidor de aplicaciones LoRaWAN, y viceversa.

Adicional a esto, LoRaWAN utiliza contadores de trama para detectar y bloquear ataques de repetición. Al activarse un dispositivo en la red, se establecen a cero los contadores de trama de enlace ascendente y descendente. El contador correspondiente a estos enlaces se va incrementando conforme se acepten los mensajes. En el caso de que el nodo o la red recibieran un mensaje con un contador de tramas inferior al último aceptado, este sería ignorado [49].

2.2.3.2.4.3.1. Procedimiento de unión

LoRaWAN en su especificación establece dos métodos para activar un dispositivo en la red: activación por personalización (ABP) y activación por aire (OTAA). Una gran diferencia entre los métodos mencionados es que ABP descarta el procedimiento de unión, vinculando directamente al nodo en la red. Esto, aunque permite una conexión rápida, debilita la seguridad ya que las claves de cifrado se encuentran preconfiguradas en el dispositivo final [50]. Por otro lado, los nodos que utilizan el método de activación OTAA obligatoriamente siguen el procedimiento de unión para intercambiar datos con el servidor de red LoRaWAN. Antes de empezar este procedimiento se debe personalizar en el dispositivo un ID de 64 bits denominado DevEUI, que identifica de manera única al nodo final, y una clave de aplicación AES-128, conocida como AppKey, asignada de forma específica al nodo por el dueño de la aplicación [51].

El procedimiento de unión se resume en lo siguiente [39]:

1. Un nodo envía al servidor de unión un mensaje MAC de solicitud de unión para autenticarse dentro de la red.

2. Si se autentica al nodo en la red, el servidor de unión responderá al dispositivo con un mensaje MAC de aceptación.
3. Con los parámetros enviados en el mensaje de aceptación y los ya personalizados en el dispositivo, este generará dos claves de sesión localmente. El servidor de unión, de igual manera, derivará las claves de sesión a partir de los parámetros personalizados y los presentes en los mensajes de solicitud y aceptación de unión.
4. Finalmente, el servidor de unión compartirá las claves de sesión a los servidores de red y de aplicación LoRaWAN correspondientes.

2.2.3.2.4.3.2. Integridad y protección de mensajes

Como se menciona en el anterior apartado, se necesita dos claves de sesión para proteger el tráfico de control y de datos entre los dispositivos finales y los servidores LoRaWAN. A continuación, se detalla cada una de estas [51]:

- Clave de sesión de red AES-128 o denominada NwkSKey: permite calcular y verificar el código de integridad del mensaje (MIC). Este proceso comprueba que la integridad del paquete no haya sido comprometida. Tanto el nodo final como el servidor de red comparten una misma NwKSKey.
- Clave de sesión de aplicación AES-128 o denominada AppSKey: permite cifrar y descifrar la carga útil de cada mensaje, asegurando que ni la puerta de enlace ni el servidor de red lean el payload del paquete transmitido. Tanto el nodo final como el servidor de aplicaciones comparten una misma AppSKey.

2.2.3.3. Comparación de tecnologías LPWAN

Cada tecnología LPWAN tienen características útiles que, dependiendo del caso, pueden ser aprovechadas en distintas aplicaciones. En la Tabla 3, se presenta una comparación de las tecnologías anteriormente descritas [52].

Tecnologías LPWAN	Descripción	Regulación	Alcance de comunicación	Potencia máxima de salida	Seguridad	Ventajas	Desventajas	Para su uso se requiere	Cálculo de costos
Sigfox	Sigfox es una red y un protocolo patentado, eso quiere decir, que el control de la red es completamente de la empresa. Esta red trabaja con mensajes de enlace ascendente livianos. Es de baja velocidad y baja potencia, pero también de largo alcance.	Banda ISM sin licencia, pero las estaciones base solo las administra Sigfox.	Hasta 30 a 50 km en áreas rurales y de 3 a 10 km en entornos urbanos.	0,025 W	"De forma predeterminada, los datos se transmiten a través de la interfaz aérea sin ningún cifrado". En la capa de aplicación existe un método de autenticación de extremo a extremo basado en una clave secreta además de brindar a cada dispositivo una identificación estática única.	• Lectura remota o monitorización de cualquier tipo de variable, como por ejemplo de agua, gas, electricidad u otros. • Se utiliza principalmente donde no es necesario tener mensajes de enlace descendente al dispositivo. • Monitoreo de funcionamiento en instalaciones de producción, clima, etc.	• Sigfox limita la cantidad de mensajes que se pueden enviar, a un mensaje cada 10 minutos en un período de 24 horas. • La velocidad lenta como un sms puede tardar un minuto en enviarse, por lo que no es una opción para los productos de IoT que requieren un ciclo de retroalimentación inmediata, como el monitoreo de la salud. • No es ideal para rastrear un vehículo en tiempo real.	La infraestructura de Sigfox esté presente en el territorio para poder contratar el servicio, así como una suscripción para usar la red sigfox.	El cálculo de costo se realizaría mediante los planes de suscripción.
NB-IoT	NB-IoT se implementa en el espectro de radio de la telefonía móvil y se acopla a canales GSM antiguos no utilizados, o espacio libre entre canales LTE.	Necesita una frecuencia / bajo bandas de frecuencia con licencia.	Los dispositivos que implementan NB-IoT dependen de la cobertura de redes móviles, por lo que funcionarían bien en interiores y en áreas urbanas densas.	0,2 W (máximo)	NB-IoT hereda la autenticación y el cifrado de LTE.	• Permite comunicaciones eficientes y alta durabilidad de la batería. •Hace uso de las actuales estaciones base de 4G (redes móviles), no es necesario realizar nuevas instalaciones. •Actualización de sistemas basados en GSM. Principalmente para lecturas de sensores, seguimiento y gestión de flotas.	• Es difícil implementar la transferencia de archivos. • Algunas de las especificaciones de diseño hacen que sea difícil enviar grandes cantidades de datos a un dispositivo. • Es más adecuado para dispositivos principalmente estáticos, como medidores y sensores en una ubicación fija, en lugar de activos en itinerancia.	Una suscripción con un proveedor de telefonía móvil.	Suscripción, tarjeta SIM, costos de datos, hardware.
LTE Cat-M	LTE que proporcionan un protocolo de baja velocidad, baja potencia y largo alcance para la transmisión de pequeñas cantidades de datos. Debido a que se ejecuta sobre estaciones base LTE, la implementación es más económica ya que no se necesita hardware dedicado.	Necesita una frecuencia / bajo bandas de frecuencia con licencia.	Los dispositivos que implementan LTE Cat-M1 dependen de la cobertura de redes móviles (4G), por lo que funcionarían bien en interiores y en áreas urbanas densas.	0,2 W (máximo)	LTE Cat-M1 hereda la autenticación y el cifrado de LTE.	• Utiliza chips 4G que son menos costosos de fabricar y cuentan con un modo especial de economizar la batería del dispositivo. • LTE-M no necesita que los operadores construyan una nueva infraestructura para implementarla. Es decir, utiliza las torres celulares presentes en el país para transportar los datos.	• Utiliza las torres celulares presentes en el país para transportar los datos esto genera algunas desventajas ya que la red presentará los mismos puntos ciegos que tengan las redes celulares, además si la conexión de estas se interrumpe por mantenimiento, límite de suscripción, emergencias o alguna otra razón, el servicio LTE-M también sufrirá interrupciones.	Una suscripción con un proveedor de telefonía móvil.	Suscripción, tarjeta SIM, costos de datos, hardware.
LoRaWAN	Es un protocolo de comunicación de baja potencia, largo alcance y bajo consumo energético. Es de especificación abierta, es decir, cualquiera es libre de implementar el protocolo.	Banda ISM sin licencia.	Hasta 5 km en aréas urbanas y 10 km en areás rurales.	0,025 W	Basado en sesiones, donde cada sesión se inicia con claves estáticas, pero después de un intercambio de claves se utiliza un conjunto único de claves AES.	• Implementación de una red aislada o privada en zonas urbanas o rurales. • En la transmisión y recepción de datos LoRaWAN es ideal, debido a que requiere baja corriente (menos de 5 mA),esto quiere decir que, reduce el consumo de energía de los dispositivos y permite una vida útil de la batería hasta más de 10 años. • Gracias a la modulación LoRa que implementa LoRaWAN, permite una penetración profunda, es decir, agrega la capacidad de alcanzar sensores ubicados bajo tierra.	• Las limitaciones en la banda de frecuencia utilizada pueden causar una alta latencia en los mensajes entregados. Por lo tanto, no es una opción para los productos de IoT que requieren un ciclo de retroalimentación inmediata, como el monitoreo del estado.	Invertir en la propia red con estaciones base. Sin embargo, una estación base necesitará conexión a Internet y alimentación. Bandas ISM, depende de la región o del plan de frecuencia que se aplique en el país.	Los costos de invertir en la creación de su propia red se compensarán al desplegar redes con frecuencias de operación gratuitas y estaciones base de bajo costo en pocos meses y con una inversión mínima.

Tabla 3. Comparación de tecnologías LoRaWAN

Fuente [52]

2.2.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE

Una metodología de desarrollo de software hace referencia a un marco de trabajo o framework cuya finalidad es la de optimizar, estructurar, planificar y controlar el proceso y desarrollo de un proyecto software [53].

Utilizar una metodología de desarrollo de software de forma correcta, permite al equipo de trabajo mejores estimaciones, comunicación continua con el cliente, entrega de sistemas estables, entendimiento claro de las tareas que tienen que realizar e identificación de dificultades o problemas que retrasan la finalización del proyecto en cuestión, garantizando elevar los estándares y el aseguramiento de éxito de un proyecto software [54]. Por otro lado, si no se hace uso de una metodología de desarrollo de software el resultado final del producto será impredecible y no se podrá controlar el avance del proyecto [55].

En la actualidad, se debe de tomar en cuenta qué metodología de desarrollo es la más adecuada para obtener un proyecto software de calidad. Existen algunas características a considerar como: la velocidad de finalización, el tamaño del sistema y el nivel de colaboración e interacción entre los miembros del equipo de desarrollo del software. Es por eso que, dentro de las metodologías de desarrollo, existen dos enfoques principales que son: las tradicionales y ágiles [56].

La metodología de desarrollo tradicional trabaja de manera lineal, rigurosa y disciplinada, esto quiere decir, que no se adapta adecuadamente a los cambios donde los requisitos pueden variar y se centra especialmente en planificar todo el trabajo a realizar mediante una documentación exhaustiva. Una vez que está todo detallado comienza el ciclo de desarrollo del proyecto software. Por otro lado, la metodología de desarrollo ágil surge para solventar los problemas que ocasiona la implementación de una metodología tradicional debido a que se centra especialmente en ser iterativa e incremental, adaptable y flexible a los cambios donde los requisitos no pueden predecirse o pueden variar y presentar constante comunicación entre cliente y el equipo de desarrollo durante el proceso [53]**.**

La metodología ágil se enfoca en la construcción de un proyecto satisfactorio en el menor tiempo posible, es por eso que se convierte en la más usada en el mundo del software. Dentro de la metodología ágil existen diferentes marcos de trabajos, tal es el caso de Scrum, el cual es un marco de trabajo ágil que permite construir y obtener un producto útil en poco tiempo, para que el cliente pueda probarlo en situaciones reales de consumo, permitiendo la mejora continua [53]. Por tal motivo, este marco de trabajo es adecuado para efectuar el desarrollo de la solución propuesta, misma que busca entregar un prototipo útil que se encuentra sujeto a cambios y pruebas constantes, permitiendo entregas continuas con el fin de atender mejor las necesidades y deseos del usuario final.

2.2.4.1. Marco de trabajo Scrum

Scrum es un marco de trabajo ágil que se enfoca especialmente en desarrollar, entregar y mantener de manera productiva productos complejos con el mayor valor posible. Este marco de trabajo se compone del equipo Scrum junto a los roles, eventos, artefactos y reglas.

2.2.4.1.1. Roles de Scrum

El equipo de trabajo Scrum define tres tipos de roles [57]:

- **Product Owner:** es la persona que va a estar en contacto con el cliente y el asegura que el equipo de trabajo comprenda de manera clara y concisa lo que se va a desarrollar.
- **Scrum Master:** es la persona que coordina al equipo de trabajo y el que guía el uso correcto de estándares Scrum en el proceso de desarrollo.
- **Development Team:** son las personas con habilidades especializadas para el desarrollo, que se encargarán de realizar las tareas definidas por el Product Owner.

2.2.4.1.2. Artefactos

Son herramientas que sirven para la implementación del proyecto software. Siendo estos los siguientes [57]:

- **Product Backlog (PB):** es una lista creada mediante las Historias de Usuario (H.U.) por el Product Owner, el cual la ordena por prioridades. Siendo una historia de usuario una descripción corta y esquemática de las funcionalidades que debe incorporar el proyecto software. Para la presente solución se ha propuesto el siguiente formato para la recopilación de historias de usuario, Tabla 4.

Historia de usuario	**Identificación**
Título:	
Descripción:	
Prioridad:	

Tabla 4. Formato de historias de usuario

El formato consta de los siguientes campos:

- **Identificación:** es el código que identifica a una historia de usuario.
- **Título:** es una descripción global del requerimiento del cliente.
- **Descripción:** es una pequeña descripción de la funcionalidad que se desea realizar.

- **Prioridad:** esta característica indica el nivel de importancia de la historia de usuario. Puede contener tres niveles: alta, media o baja.

- **Sprint Backlog:** es la descomposición de una H.U del P.B en tareas más pequeñas para alojarse en el Scrum board donde se va a ir reflejando el avance de cada tarea.
- **Incremento:** es el producto completo que vamos logrando Sprint por Sprint. Tomando en cuenta que los Sprints son ciclos de trabajo con una duración fija que va desde una a cuatro semanas, con el fin de obtener un producto final, utilizable y potencialmente liberable.

2.2.4.1.3. Eventos [58]

Son actividades que se realizan dentro del ciclo principal de Scrum llamado Sprint. Entre estos se tiene:

- **Sprint Planning:** es una reunión donde participa todo el equipo Scrum, con la finalidad de detallar, aclarar y delegar las características del producto a desarrollar.
- **Daily Scrum:** es una reunión que se lleva a cabo todos los días durante el Sprint y que tiene una duración de 15 minutos, con el fin de comprobar el progreso o inconvenientes que se tiene al realizar cada tarea asignada a los miembros del equipo de desarrollo.
- **Sprint Review:** es una reunión que dirige el Scrum Master, en la cual asisten el equipo de desarrollo junto a las partes interesadas, con fin de revisar y analizar si se logró cumplir con el objetivo del Sprint durante el trabajo realizado.
- **Sprint Retrospective:** es una reunión que se lleva a cabo con todo el equipo scrum, con el fin de conocer e inspeccionar aquello que produjo problemas durante el Sprint, de modo que se permita proponer mejoras para iniciar con un nuevo Sprint.

CAPÍTULO III

3. DISEÑO DE LA ARQUITECTURA DEL PROTOTIPO DE RED LPWAN

3.1. ELEMENTOS DE LA RED LORAWAN

En base a la arquitectura que debe cumplir una red LoRaWAN, se ha indagado y establecido los elementos necesarios a utilizar para el desarrollo del prototipo de red propuesto, siendo estos los siguientes:

3.1.1. Nodos detectores de intrusión

Este elemento se encarga de detectar y notificar algún acto de intrusión en el lugar de la vivienda donde se encuentre instalado. Por lo que, su función principal es emitir dicha notificación a la puerta de enlace haciendo uso de la tecnología LoRa. Debido a la diversidad de sensores existentes actualmente y a los diferentes escenarios en los que un ladrón podría ingresar a una vivienda, se ha considerado usar dos tipos de ellos: un sensor de movimiento (PIR HC-SR60), que permitirá detectar la presencia de un intruso en el inmueble; y un sensor magnético (MC-38), capaz de detectar la apertura de la puerta o ventana en la que se encuentre colocado. Cada sensor se conecta a una tarjeta WiFi LoRa 32 (V2) diferente, mismas que se encargan de enviar, mediante LoRa, las detecciones realizadas por los sensores al Gateway.

Para tener un contexto claro de los elementos que componen a los nodos en cuestión, a continuación, se procede a detallarlos:

3.1.1.1. Placa WiFi LoRa 32 (V2)

Es una placa de desarrollo, elaborada por Heltec Automation, que usa un chip SX1276 de Semtech integrado para manejar las comunicaciones LoRa en la banda 868/915 MHz, el cual es gestionado por el microcontrolador ESP32, y que se encuentra ubicado debajo de la pantalla OLED de 0.96”. En lo que se refiere a consumo energético posee un diseño de baja potencia que mejora la administración de energía (800 µA en sueño profundo). Una gran ventaja de este módulo es que cuenta con una propia biblioteca de la pila LoRaWAN, la cual se integra fácilmente con Arduino, convirtiéndola en una placa óptima para poder implementar nodos funcionales basados en este protocolo. En la Figura 16, se puede visualizar la tarjeta de desarrollo en cuestión, así como los distintos elementos que la conforman.

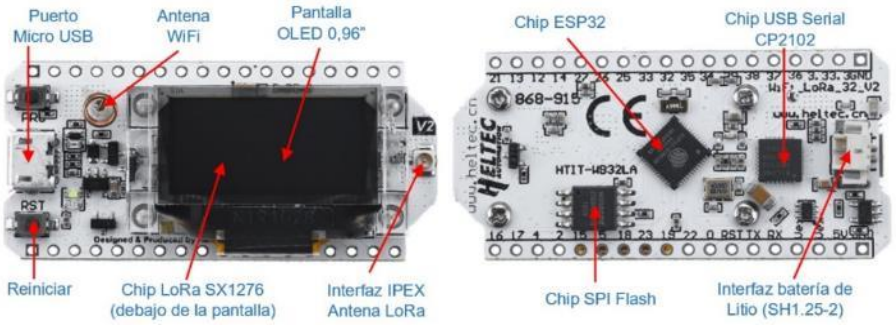


Figura 16. Elementos que conforman la placa WiFi LoRa 32 (V2)

Cabe señalar que la tarjeta ocupa 6 pines GPIO (General Purpose Input/Output) para la comunicación interna, mediante SPI (Serial Peripheral Interface), entre el ESP32 (maestro) y el chip SX1276 (esclavo). En el Anexo 1, se muestra el diagrama de pines y esquemático de la placa, mismos que han sido extraídos de su página oficial. Según sus especificaciones se puede alcanzar ~ 3 km de comunicación LoRa en área abierta. Además, cuenta con una interfaz IPEX para la conexión de la antena omnidireccional, incluida en la compra del módulo, y que otorga una ganancia aproximada de 2.5 dBm (decibelio-milivatio). La Tabla 5, presenta un resumen de las principales características técnicas de esta placa de desarrollo [59].

Recurso	Descripción
Microcontrolador	ESP32 (240MHz Tensilica LX6 dual-core + 1 ULP, 600 DMIPS)
Comunicación inalámbrica	Wi-Fi: 802.11 b/g/n (802.11n hasta 150 Mbps)
	Bluetooth: V4.2 y BLE
	LoRa: comunicación nodo a nodo o LoRaWAN
Chip LoRa	SX1276
Potencia de salida máxima LoRa	19Db ± 1 Db
FLASH	8MB (64M-bits) SPI FLASH
RAM	520KB SRAM interno
Interfaz	1 micro USB; 1 IPEX Antena LoRa; 36 pines
Tamaño máximo	51 x 25.5 x 10.6 mm
Chip USB a Serial	CP2102
Batería	3.7V Litio (SH1.25 x 2 socket)
Baja potencia	800µs en sueño profundo
Tamaño del display	OLED de 0.96 pulgadas
Temperatura de funcionamiento	-40 hasta 80°C

Tabla 5. Características de la tarjeta de desarrollo WiFi LoRa 32 (V2)

3.1.1.2. Sensor PIR HC-SR501

Es un elemento electrónico que contiene un sensor PIR (Passive Infrared) ubicado en la parte superior de la placa, el cual es cubierto por un lente de Fresnel que le permite ampliar su campo o rango de visión hasta 110°. Este sensor no irradia o genera energía, sino que

se encarga de recibir y detectar la radiación infrarroja que emiten los objetos presentes dentro su campo visual. Al cambiar dicha radiación (por ejemplo, cuando una persona ingresa al lugar o se mueve) el sensor se activará, arrojando 3.3v por su pin de salida.

El módulo PIR está conformado por tres pines (GND, OUT-3,3v y VCC-5v) que pueden ser conectados fácilmente a las entradas de cualquier microcontrolador. También posee 2 potenciómetros integrados en la placa que le permiten al usuario ajustar, según sean sus necesidades, los parámetros tanto del tiempo de activación (3 segundos a 5 minutos) como la sensibilidad de detección (3 a 7 metros) del sensor PIR. La Figura 17, presenta la disposición de los componentes descritos.

Figura 17. Elementos que conforman al sensor de movimiento PIR HC-SR501

Con el fin de obtener un conocimiento específico del sensor PIR, se utiliza la Tabla 6, la cual muestra un resumen de sus principales características técnicas [60].

Recurso	Descripción
Sensor PIR	LHI778
Controlador	BISS0001
Voltaje de alimentación	5 – 12 voltios de corriente directa
Consumo en estado de reposo	~50µA
Sensibilidad de detección	3 a 7 metros (ajustable)
Tiempo de activación	3 a 200 segundos (ajustable)
Angulo de detección	90 a 110°
Tiempo de bloqueo	3 segundos
Temperatura de funcionamiento	-20 a 70°C
Dimensiones	32 24 x 18 mm

Tabla 6. Características del sensor PIR HC-SR501

3.1.1.3. Sensor MC-38

Es un tipo de sensor magnético conformado por 2 elementos: un imán y un interruptor de lengüeta (reed switch) de 14 mm de largo, ambos cubiertos por plástico ABS (Acrylonitrile Butadiene Styrene), un material caracterizado por su dureza y resistencia a impactos. El interruptor de lengüeta posee, a la vez, 2 láminas ferrosas en su interior, mismas que se

unen al ser expuestas a un campo magnético, y que se alejan cuando este desaparece. La Figura 18, presenta los elementos en cuestión, así como el funcionamiento del interruptor reed.

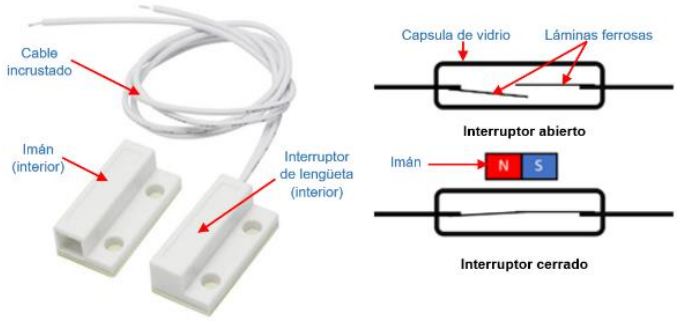


Figura 18. Elementos que conforman el sensor magnético MC-38

El funcionamiento del sensor es muy simple ya que actúa como uno de tipo N.C (normalmente cerrado). Es decir, cuando ambas piezas del sensor entran en contacto, o se encuentran a una distancia aproximada de 15 a 25 mm, el circuito se cierra permitiendo el flujo de corriente; sin embargo, cuando estas se alejan, el circuito se abre debido a la ausencia del campo magnético producido por el imán. Este funcionamiento es ideal para determinar la apertura de puertas o ventanas dentro de las viviendas, además, debido a su simplicidad de uso resulta conveniente utilizarlo junto con algún microcontrolador, ya que permitirá enviarle una señal digital cuando esto ocurra.

Algo importante que se debe tomar en cuenta es que el sensor no podrá ser colocado en puertas elaboradas con materiales ferromagnéticos debido a que alterarían su funcionamiento y sensibilidad. La Tabla 7, presenta un resumen con las principales especificaciones técnicas del sensor MC-38 [61].

Recurso	Descripción
Máximo voltaje	100V
Máxima corriente	500Ma
Distancia de activación	15 a 25 mm
Material	Plástico ABS y metal
Vida útil	1 millón de contactos
Tipo de contacto	Normalmente cerrado
Longitud del cable	25cm
Temperatura de funcionamiento	Aprox. -26 a 70°C
Dimensiones	34 x 41 x 6,5 mm

Tabla 7. Características del sensor MC-38

3.1.2. Puerta de enlace

Este elemento es el responsable de recibir y retransmitir los paquetes enviados por los nodos detectores de intrusión hacia los servidores LoRaWAN, y viceversa. Para llevar a

cabo dicho procedimiento el dispositivo debe contar principalmente de un chip LoRa y tener conectividad a internet para poder retransmitir el paquete mediante una comunicación TCP/IP.

Actualmente, en el mercado existe diversos modelos de puertas de enlace ofrecidas por distintos fabricantes, incluso se puede implementar una utilizando un Raspberry Pi. Sin embargo, se debe considerar que entre ellas existen 2 tipos: puertas de enlace de un solo canal y puertas de enlace “completas” o multicanal. La gran diferencia entre estas es que las primeras, aunque son más económicas, no cumplen totalmente con el estándar LoRaWAN debido a que únicamente escuchan en una sola frecuencia y en un determinado spreading factor. Además, la mayor parte de estas no tiene soporte para enlaces descendentes [62], funcionalidad indispensable dentro del prototipo de red propuesto, ya que estos enlaces son utilizados para establecer tanto la autenticación OTAA como para activar y desactivar el envío de mensajes de intrusión por parte del dispositivo final.

Aunque existen módulos LoRa que pueden integrarse con un Raspberry Pi para crear puertas de enlace multicanal, resultan más costosas que algunas soluciones ya fabricadas. Es por ende que, para el desarrollo del prototipo de red propuesto, se ha considerado utilizar el Gateway RAK7258 debido a sus características y precio.

3.1.2.1. RAK7258

Es una puerta de enlace completa de 8 canales basada en el protocolo LoRaWAN. Es capaz de establecer comunicaciones bidireccionales y permite la transmisión de datos mediante Ethernet. Además, cuenta con conectividad Wi-Fi para su fácil configuración. Posee un reenviador de paquetes (software encargado del reenvío de paquetes LoRa hacia un servidor LoRaWAN a través de un enlace IP/UDP) configurable desde la interfaz gráfica de usuario, misma que también se utiliza para configurar los parámetros relacionados a la red y al protocolo LoRaWAN.

En cuanto a LoRa, la puerta de enlace integra un concentrador RAK2247 que mediante el chip SX1301 de Semtech es capaz de admitir 8 canales de enlace ascendente y 1 canal de transmisión de enlace descendente LoRa. Según su fabricante [63], este elemento proporciona radios de comunicación de hasta 15 km con línea de visión y de 2 km en sitios urbanizados. Además, es compatible con nodos de clase A y C, y posee una sensibilidad de recepción de -142 dBm junto con una potencia de transmisión de 27 dBm.

La Figura 19, muestra la puerta de enlace descrita, así como los distintos elementos que la conforman.

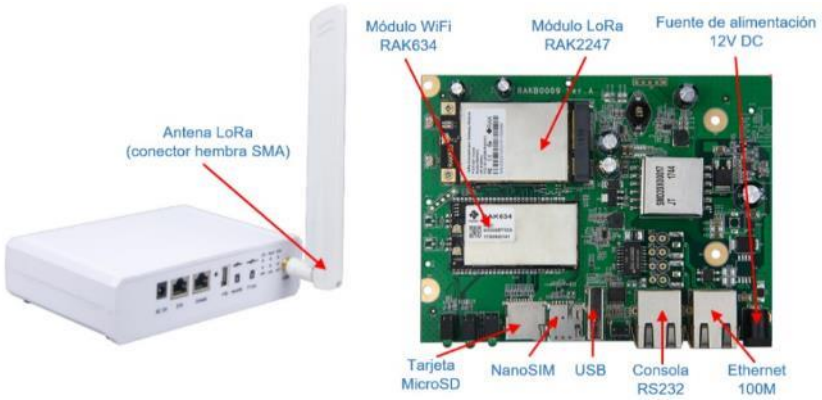


Figura 19. Elementos que conforman la puerta de enlace RAK7258

Cabe señalar que el Gateway es de tipo indoor, es decir, está diseñado para trabajar exclusivamente en ambientes internos. Sin embargo, al ser compatible con PoE (Power over Ethernet) puede ser fácilmente instalado en paredes o techos sin necesidad de usar conexiones eléctricas extras. La Tabla 8, presenta un resumen de las principales características técnicas de esta puerta de enlace [64].

Recurso	Descripción	
Características LoRa	**Chip**	SX1301
	Canales	8
	Sensibilidad de recepción	-142 dBm
	Potencia de transmisión	27 dBm
	Frecuencias	EU433 / CN470 / EU868 / US915 / AS923 / AU915 / IN865 / KR920
	Clases soportadas	A y C
	Versión LoRaWAN	1.0.2
Características Wi-Fi	**Frecuencia**	2,400 -2,4835 GHz
	Estándar	802.11b/g/n
	Sensibilidad de recepción	-95 dBm
	Potencia de transmisión	20 dBm
Fuente de poder	12V corriente directa / 1 A	
	PoE (IEEE 802.3af), 42-57VDC	
Consumo de energía	12W	
Antena	**LoRa**	conector hembra SMA
	Wi-Fi	Antena interna
Dimensiones	180 x 120 x 43 mm	
Temperatura de funcionamiento	-10 a 55°C	

Tabla 8. Características de la puerta de enlace RAK7258

3.1.3. Servidores LoRaWAN

Son los elementos responsables de administrar el prototipo de red LPWAN planteado. Entre sus principales funciones se encuentran:

- Gestionar tanto a los nodos detectores de intrusión como a la puerta de enlace RAK7258.
- Activar a los nodos y autorizar su unión a la red LoRaWAN.
- Recibir los paquetes enviados por el RAK7258, verificando su integridad y descifrándolos.
- Gestionar el enlaces ascendente y descendente para el intercambio de datos entre las aplicaciones y los nodos detectores de intrusión.
- Permitir la integración con la aplicación cliente con el afán de que esta pueda visualizar o utilizar los datos que el servidor LoRaWAN maneja.

En la actualidad, existen varias organizaciones que implementan servidores LoRaWAN, entre estas se encuentran: The Things Newtwork (TTN) y ChirpStack (anteriormente denominado LoRa Server). La primera se caracteriza por ofrecer un servidor LoRaWAN gratuito dentro de una red pública, permitiendo que los usuarios de la plataforma puedan usar cualquier puerta de enlace registrada en la misma; siempre y cuando esta se encuentre dentro del rango de comunicación de los nodos. Aunque esto se considera una ventaja en ciertas situaciones, también resulta un inconveniente en otras, por ejemplo, no se podría implementar una red completamente privada, ya que como se mencionó, al registrar un gateway este se comparte con el resto de los usuarios. Para resolver el inconveniente presentado, TTN ofrece una versión comercial denominada “The Things Industries”; sin embargo, esta tiene un costo, que varía dependiendo del plan a contratar.

Otra limitante de TTN es que admite únicamente 10 mensajes de enlace descendente por día [65]. Lo que genera un inconveniente dentro de la solución propuesta, debido que se desconoce el número de veces que el usuario necesitará usar este tipo de mensajes. Ahora bien, ChirpStack no presenta dichos problemas, y es por esto que se lo considera como la mejor opción a utilizar para el desarrollo del presente proyecto. Por ende, a continuación, se procede a detallar su funcionamiento y los elementos que lo conforman.

3.1.3.1. ChirpStack

Es un proyecto de código abierto, desarrollado por Orne Brocaar, que provee los elementos necesarios para crear redes LPWAN privadas basadas en el estándar LoRaWAN, mismos que pueden usarse con fines comerciales bajo la licencia MIT. Es un software fácil de instalar y configurar debido a que posee una documentación muy completa. Entre sus principales características se tiene:

- Compatibilidad con las versiones 1.0 y 1.1 del protocolo LoRaWAN
- Admite dispositivos finales de clase A, B y C

- Registro de tramas en vivo para cada nodo y puerta de enlace.
- Posee una interfaz web que permite la administración de aplicaciones, gateways y dispositivos finales.
- Ofrece un API RESTful que puede utilizarse al obtener un token a través de la consola de la API.
- Permitir integraciones con base de datos y programas de visualización.

Ahora bien, para tener un contexto claro de ChirpStack, a continuación, se presenta un breve resumen del funcionamiento de sus componentes.

- **ChirpStack Gateway Bridge:** se encuentra entre el reenviador de paquetes y el servidor de red de ChirpStack. Su principal función es transformar, en formato JSON, los mensajes enviados por el reenviador de paquetes para publicarlos en el bróker MQTT Mosquito.
- **Servidor de red ChirpStack:** este servidor se suscribe al mismo tópico donde el reenviador de paquetes publica los mensajes. Su principal función es gestionar el estado de la red, permitiendo la activación de los nodos y la administración de los mensajes de enlace ascendente y descendente. Es decir, al recibir los datos se encarga de enviarlos como carga útil al servidor de aplicaciones, de igual manera, si este servidor necesita enviar algún paquete a un nodo, elige la puerta de enlace más adecuada para llevar a cabo el envío.
- **Servidor de aplicaciones ChirpStack:** gestiona la carga útil enviada por los nodos, cifrándola o descifrándola según sea el caso. Además, ofrece una API RESTful JSON, servicios MQTT, entre otros, para su integración con aplicaciones clientes. Dentro de su interfaz web se puede administrar a puertas de enlace, aplicaciones y dispositivos finales, así como ver los datos que estos últimos envían.

La Figura 20, muestra la conexión de los componentes ChirpStack mencionados.

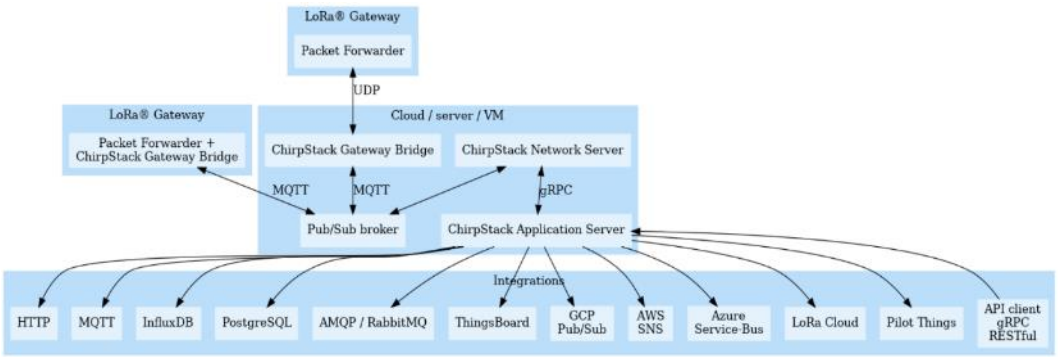


Figura 20. Arquitectura de ChirpStack

Fuente [66]

3.1.4. Aplicación cliente

Es la encargada de recibir los datos de los nodos detectores de intrusión mediante la suscripción al tópico del bróker MQTT, integración presente en el servidor de aplicación ChirpStack. Además, permite activar o desactivar a estos dispositivos haciendo uso de los métodos provistos en la consola de la API RESTful del servidor. El principal propósito de la aplicación cliente en este prototipo, es mostrar notificaciones de actos de intrusión tanto al propietario de la vivienda como al personal de seguridad de la zona residencial. El software utilizado para la creación de la aplicación es descrito en la sección 3.3.

3.2. ARQUITECTURA DEL PROTOTIPO DE RED LPWAN

Una vez establecidos los elementos a utilizar dentro de la solución propuesta, resulta conveniente presentar la arquitectura final de la misma, es por ende que en la Figura 21 se muestra la relación de todos los componentes anteriormente descritos.

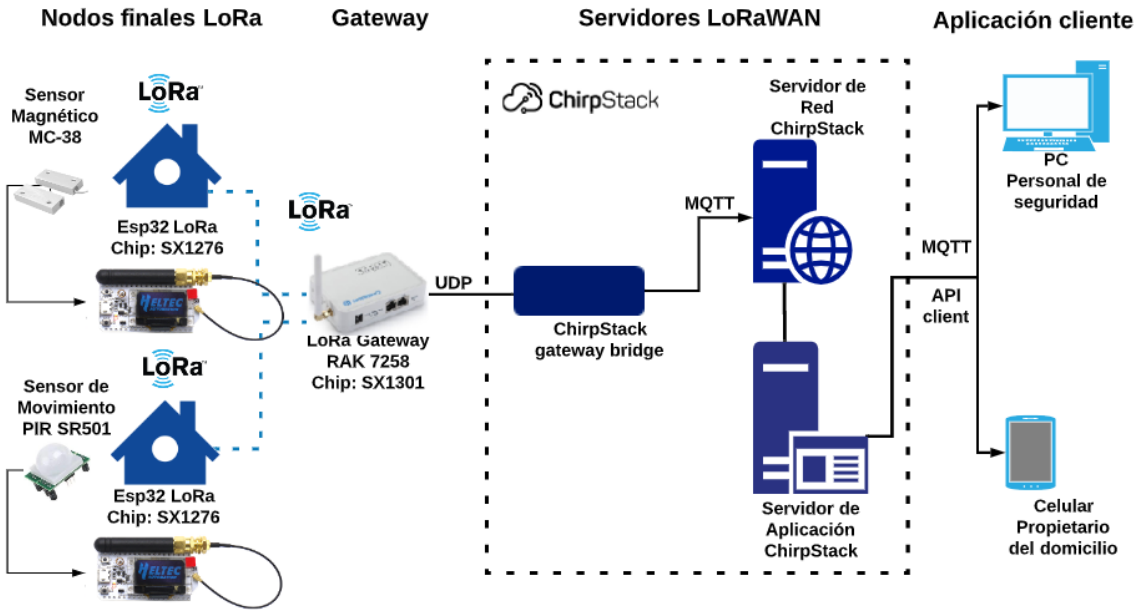


Figura 21. Arquitectura del prototipo de red LPWAN

3.3. HERRAMIENTAS DE DESARROLLO

Para la implementación de las aplicaciones web y móvil del sistema de detección de intrusos se usaron las herramientas presentadas en las Tablas 9-16.

3.3.1. Lenguajes de programación

Nombre	Descripción	Aplicado en	Logo
HTML5	Es un lenguaje de marcado de hipertexto (HTML, por sus siglas en inglés) es el componente más básico de un sitio web y permite estructurar y organizar el contenido del sitio web. HTML5 es la última versión que brinda nuevas características para que los sitios o aplicaciones web sean más diversos y de gran alcance [67].	Aplicación web y móvil	
CSS	Es un lenguaje de hojas de estilo en cascada (CSS, por sus siglas en inglés) usado para describir la presentación de un documento escrito en HTML o XML. Permite diseñar páginas web, por ejemplo, para modificar el color, tamaño y estilo de fuente y otras características [68].	Aplicación web y móvil	
JavaScript	Es un lenguaje de programación, dinámico, sencillo, ligero e interpretado, que admite estilos orientados a objetos y programación funcional. Hoy en día ha crecido de manera acelerada y es un lenguaje que se puede usar en distintos entornos, tales como Node.js, Apache CouchDB, entre otros [69].	Aplicación web y móvil	
TypeScript	Es un lenguaje de programación de código abierto, que se basa en JavaScript, permite agregar definiciones de tipo estáticos, dando como resultado una mejor documentación y validez para que el código funcione correctamente. El código TypeScript se puede transpilar en código JavaScript por medio del compilador de TypeScript o Babel, el nuevo código JavaScript que surge es limpio y simple que se puede ejecutar en un navegador, servidor Node.js o en aplicaciones, donde se ejecute JavaScript. [70].	Aplicación web y móvil	
C	Es un lenguaje de programación estructurado, que proporciona gran flexibilidad de programación, pero muy baja comprobación de incorreciones, esto quiere decir, que es responsabilidad del programador revisar la sintaxis de programación o acciones que otros lenguajes realizan por sí mismos [71].	Aplicación Nodo LoRa	

Tabla 9. Descripción de lenguajes de programación

3.3.2. Frameworks de desarrollo

Nombre	Descripción	Aplicado en	Logo
Angular	Es un framework de diseño de aplicaciones web y creación de aplicaciones de una sola página (SPA, por sus siglas en inglés), gratuito y de código abierto [72]. Usa como lenguaje de programación TypeScript, conjunto de JavaScript/ECMAScript que facilita mucho el desarrollo [73].	Aplicación web	
IONIC	Es un framework gratuito y de código abierto, que ofrece una biblioteca de herramientas y componentes optimizados para dispositivos móviles y con el objetivo de crear aplicaciones web y nativas multiplataforma rápidas, interactivas y de alta calidad [74].	Aplicación móvil	
Sails.js	Es un framework utilizado para la creación de aplicaciones modernas relacionadas con Node.js y que trabaja con el patrón de arquitectura de software, modelo vista controlador (MVC, por sus siglas en ingles). Sails.js permite la integración con diferentes bases de datos proporcionando una capa de acceso a datos simple. De igual forma, permite la integración con herramientas de vanguardia como los WebSocket y es compatible a nivel de interfaz con: Angular, React, Ios, Android y Windows Phone [75].	Aplicación web y móvil	
Node JS	Es un entorno ideado para la ejecución de JavaScript y para la creación de aplicaciones network escalables, construido con el motor de JavaScript V8 de Chrome y orientado a eventos asíncronos [76].	Aplicación web y móvil	

Tabla 10. Descripción de frameworks de desarrollo

3.3.3. Librerías

Nombre	Descripción	Utilizado	Logo
Leaflet	Es una biblioteca JavaScript de código abierto, que sirve para desarrollar mapas eficientes e interactivos para todas las plataformas de escritorio y móviles. Brinda a los desarrolladores una API bien documentada de fácil uso, con funciones de mapeo y código fuente simple y legible [77].	Aplicación web	
MOSCA	Es una librería desarrollada en JavaScript, de código abierto para Node.js basado en el protocolo MQTT. Puede ser empleado como aplicación independiente o embebido en diferentes proyectos Node.js. Sirve para la comunicación entre dispositivos IOT mediante eventos que permiten suscribirse a tópicos o realizar publicaciones [78].	Aplicación web	
EEPROM	Memoria de solo lectura, borrable eléctricamente (EEPROM, por sus siglas en inglés) permite leer y escribir bytes de la EEPROM de Arduino, es decir, sirve para gestionar el uso de memorias en las placas, mediante los métodos update, get, put, read y write. Esta librería forma parte del IDE de Arduino, así que no hay necesidad de descargarla [79].	Red LORAWAN	

Tabla 11. Descripción de librerías de implementación

3.3.4. Servicios de software

Nombre	Descripción	Aplicado en	Logo
AWS	AWS (Amazon Web Service), es una plataforma en la nube que brinda servicios de integración de datos a nivel mundial, ofrece tecnologías de infraestructura de cómputo, almacenamiento y base de datos. De igual forma, brinda tecnologías innovadoras como inteligencia artificial, análisis de datos e internet de las cosas. AWS permite la integración en la nube a este tipo de tecnologías con aplicaciones existentes de manera rápida, fácil y rentable [80].	Red LORAWAN	
ChirpStack	Es un servidor LoRaWAN de código abierto, que proporciona los componentes para la creación de redes LPWAN basadas en el protocolo LoRaWAN, incorpora una interfaz amigable con el usuario fácil de usar, para administrar dispositivos y APIs Grpc y REST para integrar con infraestructuras externas [81].	Red LORAWAN	
OneSignal	Es un servicio de envío de notificaciones push por medio de diferentes plataformas que estén previamente registrados. OneSignal, posee una documentación API REST fácil de entender, para que los desarrolladores realicen integraciones sin ningún inconveniente [82].	Aplicación web y móvil	
Docker	Es una aplicación que sirve para la creación y uso compartido de contenedores ligeros y portables con el objetivo de ejecutarse en las máquinas MacOS y Windows [83].	Aplicación web y móvil	
Putty	Es un software de código abierto, que sirve para conectarse a servidores remotos para administrarlos, por medio de una sesión que se realiza por el cliente SSH y Telnet [84].	Red LORAWAN	

Tabla 12. Descripción de servicios de software

3.3.5. Base de datos

Nombre	Descripción	Aplicado en	Logo
MySQL	Es un gestor de bases de datos relacional de código abierto, permitiendo así disponer de esta herramienta para la realización de pequeños proyectos o prototipos de manera fiable, ordenada y estandarizada. MySQL se basa en la arquitectura cliente-servidor permitiendo la comunicación entre sí, para crear y administrar bases de datos basados en el modelo relacional [85].	Aplicación web y móvil	

Tabla 13. Descripción de la Base de datos MySQL

3.3.6. Entorno de desarrollo

Nombre	Descripción	Aplicado en	Logo
Visual Studio Code	Es un editor de texto o código fuente muy ligero y potente disponible para ejecutarse en los principales sistemas operativos Windows, MacOS y Linux. Posee varias extensiones para el desarrollo diferentes lenguajes de programación como: C ++, C #, Java, Python, PHP, Go) e incorpora soporte para JavaScript, TypeScript y Node.js [86].	Aplicación web y móvil	
Arduino IDE	Es un software de código abierto escrito en Java y basado en Processing, sirve para facilitar la escritura de código y su carga en la placa Arduino. Está disponible para ejecutarse en los principales sistemas operativos Windows, MacOS X y Linux. Este software puede ser usado con cualquier placa Arduino [87].	Red LORAWAN	

Tabla 14. Descripción de entornos de desarrollo

3.3.7. Control de versiones

Nombre	Descripción	Aplicado en	Logo
Git	Es un software de código abierto y gratuito, que permite realizar el control de versiones distribuido de código fuente, de igual manera está diseñado para manejar las versiones de proyectos pequeños y grandes, con velocidad y eficiencia [88].	Aplicación web y móvil	
GitHub	Es un repositorio online que sirve para alojar y gestionar proyectos, permite el trabajo colaborativo mediante el sistema de control de versiones Git [89].	Aplicación web y móvil	
GitHub Desktop	Es una aplicación de escritorio desarrollado por GitHub que permite administrar proyectos, realizar seguimiento mediante un historial y realizar cambios de versiones de manera gráfica sin hacer uso de la línea de comandos que brinda Git [90]**.**	Aplicación web y móvil	

Tabla 15. Descripción de herramientas de control de versiones

3.3.8. Recursos y Diagramación

Nombre	Descripción	Aplicado en	Logo
Lucidchart	Es un software web que permite al usuario bosquejar y compartir diagramas de flujo profesional de forma colaborativa en tiempo real. Además, permite la elaboración de organigramas, pizarras, mapas, entre otros [91].	Creación de ilustraciones	
Flaticon	Es un sitio web donde se recopila y se aloja iconos en formato PNG, SVG, EPS, PSD y Base64, para poder usar en proyectos de desarrollo o ilustraciones [92].	Aplicación web y móvil	
Adobe Illustrator	Es un software de diseño vectorial que permite a los diseñadores e ilustradores crear logotipos, iconos, gráficos y redimensionar ilustraciones y vallas publicitarias para pantallas celulares y sitios web, sin afectar su nitidez y características [93].	Aplicación web y móvil	

Tabla 16. Descripción de herramientas adicionales

3.4. APLICACIÓN DE LA METODOLOGÍA DE DESARROLLO

3.4.1. Definición de roles

En base a lo descrito dentro del marco de trabajo Scrum se establecieron los roles necesarios para la implementación del prototipo, siendo el equipo de desarrollo los autores el presente proyecto, tomando en cuenta que todo lo realizado en estas secciones se lo hizo bajo la supervisión de su director y codirector.

3.4.2. Definición de historias épicas

La Tabla 17, muestra las historias épicas consideradas para el desarrollo del proyecto propuesto.

Código	Título	Prioridad
HE01	Implementación y comunicación de los componentes LoRa y LoRaWAN.	Alta
HE02	Desarrollo de un backend para la integración con los componentes LoRaWAN.	Alta
HE03	Desarrollo de la aplicación web y móvil que permita recibir y consumir los servicios desde una API RESTful.	Media

Tabla 17. Detalle de historias épicas

3.4.3. Pila de producto (Product Backlog)

A continuación, se presenta el listado de historias de usuario en las Tablas 18-26 necesarios para la implementación del proyecto.

HISTORIA DE USUARIO	**HE01-01**
Título: Notificación de intrusiones en una vivienda.	
Descripción: Como elemento detector de intrusos, necesito enviar mensajes de los eventos capturados por el sensor magnético y de movimiento utilizando la tecnología LoRaWAN, para que la infraestructura de backend pueda notificar dicha acción a los usuarios suscritos al sistema**.**	
Prioridad: Alta	

Tabla 18. Historia de usuario HE01-01

HISTORIA DE USUARIO	HE03-03
Título: Monitoreo del estado de los nodos de una vivienda.	
Descripción: Como administrador del sistema, necesito visualizar un mapa en tiempo real que presente el estado de los nodos de cada vivienda, para que el personal de seguridad pueda tomar las acciones necesarias dependiendo de estos estados.	
Prioridad: Media	

Tabla 19. Historia de usuario HE03-03

HISTORIA DE USUARIO	HE03-02
Título: Gestión del acceso a la aplicación web y móvil.	
Descripción: Como módulo de control de acceso, necesito autenticar a través de un login las credenciales ingresadas por el usuario, para permitirle hacer uso de las funciones de la aplicación web o móvil.	
Prioridad: Media	

Tabla 20. Historia de usuario HE03-02

HISTORIA DE USUARIO	HE01-02
Título: Configuración de los componentes de la red LoRaWAN.	
Descripción: Como administrador de red, necesito llevar a cabo diferentes configuraciones basadas en el estándar LoRaWAN y en regulaciones establecidas en el país referentes al uso de bandas libres, para que los elementos dentro de la red puedan comunicarse entre sí.	
Prioridad: Alta	

Tabla 21. Historia de usuario HE01-02

HISTORIA DE USUARIO	HE02-01
Título: Procesamiento de información.	
Descripción: Como backend de aplicación, necesito capturar, interpretar y procesar los datos obtenidos tanto del aplicativo web y móvil como de los elementos de la red, para gestionarlos adecuadamente.	
Prioridad: Alta	

Tabla 22. Historia de usuario HE02-01

HISTORIA DE USUARIO	HE03-01
Título: Gestión de la información de un usuario final.	
Descripción: Como administrador del sistema, necesito realizar operaciones CRUD en la aplicación web, para gestionar la información de cada usuario final registrado en el sistema.	
Prioridad: Media	

Tabla 23. Historia de usuario HE03-01

HISTORIA DE USUARIO	HE02-02
Título: Envío de avisos de desconexión y de actividad sospechosa.	
Descripción: Como backend de aplicación, necesito enviar notificaciones referentes a la desconexión del nodo o a la detección de intrusos tanto a la aplicación web como al dispositivo móvil, para que el personal de seguridad pueda acudir en ayuda.	
Prioridad: Alta	

Tabla 24. Historia de usuario HE02-02

HISTORIA DE USUARIO	HE03-04
Título: Gestión del estado de los nodos.	
Descripción: Como usuario final, necesito activar o desactivar los nodos a través del aplicativo móvil**,** para que el sensor vinculado a la vivienda emita mensajes únicamente si existe alguna intrusión en el inmueble.	
Prioridad: Media	

Tabla 25. Historia de usuario HE03-04

HISTORIA DE USUARIO	HE03-05
Título: Registro de alertas de detección de intrusos.	
Descripción: Como administrador del sistema, necesito tener un registro, dentro del aplicativo web, de las notificaciones emitidas al detectarse intrusos en las viviendas registradas en el sistema, para constatar que las alertas fueron recibidas por el personal de seguridad.	
Prioridad: Baja	

Tabla 26. Historia de usuario HE03-05

3.4.4. Planificación de los Sprints (Release planning)

Mediante el release plannig presentado en la Tabla 27, se especifica historias de usuario que serán desarrolladas dentro de cada sprint.

RELEASE PLANNING			
SPRINT 1	**SPRINT 2**	**SPRINT 3**	**SPRINT 4**
HE01-01	HE02-01	HE03-01	HE03-04
HE01-02	HE02-02	HE03-02	HE03-05
		HE03-03	

Tabla 27. Release plannig

Cabe señalar que para ponderar el esfuerzo de cada historia de usuario se utilizó la técnica Plannig Poker, la cual consistió en la participación activa de los miembros del equipo Scrum. Esta técnica hace uso de una baraja modificada donde las cartas están enumeradas bajo la serie de Fibonacci (0, 1, 2, 3, 5, 8, 13). Se hicieron rondas con estas para determinar la estimación de cada historia de usuario.

3.4.4.1. Sprint 1

3.4.4.1.1. Objetivo del Sprint

Conectar los dispositivos LoRaWAN mediante LoRa y TCP/IP con el afán de comunicar actos de intrusión al servidor de ChirpStack.

3.4.4.1.2. Historias de usuario

El primer sprint se desarrolló en base a las historias de usuario presentadas en las Tablas 28 y 29, las cuales muestran el esfuerzo y los criterios de aceptación relacionados a cada una de ellas.

HISTORIA DE USUARIO	**HE01-01**
Título: Notificación de intrusiones en una vivienda.	
Descripción: Como elemento detector de intrusos, necesito enviar mensajes de los eventos capturados por el sensor magnético y de movimiento utilizando la tecnología LoRaWAN, para que la infraestructura de backend pueda notificar dicha acción a los usuarios suscritos al sistema**.**	
Prioridad: Alta	**Esfuerzo:** 13
Criterios de Aceptación:	
• Cuando los nodos detecten un acto de intrusión, el mensaje tiene que llegar con un payload de 1 al servidor de aplicaciones LoRaWAN de ChirpStack	

Tabla 28. Historia de usuario HE01-01 para el Sprint 1

HISTORIA DE USUARIO	HE01-02
Título: Configuración de los componentes de la red LoRaWAN.	
Descripción: Como administrador de red, necesito llevar a cabo diferentes configuraciones basadas en el estándar LoRaWAN y en regulaciones establecidas en el país referentes al uso de bandas libres, para que los elementos dentro de la red puedan comunicarse entre sí.	
Prioridad: Alta	**Esfuerzo:** 8
Criterios de Aceptación:	
• Al enviar un paquete mediante la comunicación LoRa este debe utilizar el plan de frecuencias ISM 902_928 • Los paquetes de enlace ascendente únicamente deben utilizar SF del 7 al 10 y el Data Rate del 0 al 4.	

Tabla 29. Historia de usuario HE01-02 para el Sprint 1

3.4.4.2. Sprint 2

3.4.4.2.1. Objetivo del Sprint

Implementar un backend que permita gestionar los datos recibidos de los componentes LoRaWAN y de las aplicaciones web y móvil para presentarlos al usuario de manera adecuada.

3.4.4.2.2. Historias de usuario

En las Tablas 30 y 31 se presentan las historias de usuario, que se tomaron para el segundo sprint junto con sus respectivos criterios de aceptación:

HISTORIA DE USUARIO	HE02-01
Título: Procesamiento de información.	
Descripción: Como backend de aplicación, necesito capturar, interpretar y procesar los datos obtenidos tanto del aplicativo web y móvil como de los elementos de la red, para gestionarlos adecuadamente.	
Prioridad: Alta	**Esfuerzo:** 8
Criterios de Aceptación:	
• El backend del sistema debe hacer uso del framework Sails.js. • Al recibir los datos de la aplicación web o móvil, estos deben ser almacenados en una base de datos. • Cuando un mensaje de enlace ascendente llegue al servidor de ChirpStack, el backend debe capturar estos datos mediante la subscripción a un tópico de MQTT.	

Tabla 30. Historia de usuario HE02-01 para el Sprint 2

HISTORIA DE USUARIO	HE02-02
Título: Envío de avisos de desconexión y de actividad sospechosa.	
Descripción: Como backend de aplicación, necesito enviar notificaciones referentes a la desconexión del nodo o a la detección de intrusos tanto a la aplicación web como al dispositivo móvil, para que el personal de seguridad pueda acudir en ayuda.	
Prioridad: Alta	**Esfuerzo:** 8
Criterios de Aceptación:	
• Cuando se detecta una intrusión o desconexión del nodo, el usuario o personal de seguridad recibirá una notificación que será desplegada en una interfaz con la siguiente información: - Título de la notificación: Alerta de seguridad o Problemas de conexión. - Apellidos del propietario. - Mensaje detallando el tipo de inconveniente. - Información general de la vivienda. - Dirección de la vivienda. - Mapa con la ubicación de la vivienda.	

Tabla 31. Historia de usuario HE02-02 para el Sprint 2

3.4.4.3. Sprint 3

3.4.4.3.1. Objetivo del Sprint

Desarrollar una aplicación web y móvil que permita tanto al personal de seguridad como al usuario final ingresar al sistema con el fin de hacer uso de las funcionalidades relacionadas al monitoreo de los nodos y a la detección de intrusos.

3.4.4.3.2. Historias de usuario

Las Tablas 32, 33 y 34 se detallan las historias de usuario junto con sus respectivos criterios de aceptación para el desarrollo del tercer sprint:

HISTORIA DE USUARIO	HE03-01
Título: Gestión de la información de un usuario final.	
Descripción: Como administrador del sistema, necesito realizar operaciones CRUD en la aplicación web, para gestionar la información de cada usuario final registrado en el sistema.	
Prioridad: Media	**Esfuerzo:** 5
Criterios de Aceptación:	

• Cuando el administrador del sistema proporcione todos los campos de los diferentes formularios (usuario, vivienda y nodo) requeridos por la aplicación web, se creará un nuevo registro en la base de datos. • La aplicación web mostrará la información general de la vivienda vinculada al usuario mediante una interfaz, al dar clic sobre el botón “DETALLES”. • La aplicación web permitirá editar toda la información general de la vivienda vinculada al usuario mediante una interfaz, al dar clic sobre el botón “EDITAR”. • El administrador del sistema eliminará dentro de la aplicación web un usuario registrado en la base de datos, al dar clic sobre el botón representado por una “X” en la tabla de usuarios.

Tabla 32. Historia de usuario HE03-01 para el Sprint 3

HISTORIA DE USUARIO	**HE03-02**
Título: Gestión del acceso a la aplicación web y móvil.	
Descripción: Como módulo de control de acceso, necesito autenticar a través de un login las credenciales ingresadas por el usuario, para permitirle hacer uso de las funciones de la aplicación web o móvil.	
Prioridad: Media	**Esfuerzo:** 5
Criterios de Aceptación:	
• Cuando el usuario o el administrador del sistema proporcione credenciales erróneas en la interfaz del login de la aplicación web o móvil, el sistema negará el acceso desplegando un mensaje de error. • Cuando el usuario o el administrador del sistema proporcione credenciales correctas entonces podrá ingresar de manera exitosa a las diferentes aplicaciones.	

Tabla 33. Historia de usuario HE03-02 para el Sprint 3

HISTORIA DE USUARIO	**HE03-03**
Título: Monitoreo del estado de los nodos de una vivienda.	
Descripción: Como administrador del sistema, necesito visualizar un mapa en tiempo real que presente el estado de los nodos de cada vivienda, para que el personal de seguridad pueda tomar las acciones necesarias dependiendo de estos estados.	
Prioridad: Media	**Esfuerzo:** 8
Criterios de Aceptación:	
• Al detectarse un acto de intrusión, el icono circular que representa al nodo de la vivienda dentro del mapa cambiará su color de verde a amarillo. • Cuando un nodo pierda la conexión con la red, el icono circular que lo representa dentro del mapa cambiará su color a gris.	

Tabla 34. Historia de usuario HE03-03 para el Sprint 3

3.4.4.4. Sprint 4

3.4.4.4.1. Objetivo del Sprint

Desarrollar un módulo en la aplicación móvil que permita gestionar la activación y desactivación de los nodos, y un módulo en la aplicación web que presente los registros de las notificaciones de detección de intrusos.

3.4.4.4.2. Historias de usuario

El cuarto sprint se desarrolló en base a las historias de usuario presentadas en las Tablas 35 y 36, que se detallan a continuación junto con sus respectivos criterios de aceptación:

HISTORIA DE USUARIO	**HE03-04**
Título: Gestión del estado de los nodos.	
Descripción: Como usuario final, necesito activar o desactivar los nodos a través del aplicativo móvil**,** para que el sensor vinculado a la vivienda emita mensajes únicamente si existe alguna intrusión en el inmueble.	
Prioridad: Media	**Esfuerzo:** 8
Criterios de Aceptación:	
• Cuando el usuario cambia el estado del nodo de desactivado a activado, el botón representado por un switch cambiará su color de gris a verde. • Al detectarse un acto de intrusión y el nodo se encuentre en estado desactivado, ninguna aplicación recibirá estas notificaciones. • Al detectarse un acto de intrusión y el nodo se encuentre en estado activado, ambas aplicaciones recibirán notificaciones de alerta.	

Tabla 35. Historia de usuario HE03-04 para el Sprint 4

HISTORIA DE USUARIO	**HE03-05**
Título: Registro de alertas de detección de intrusos.	
Descripción: Como administrador del sistema, necesito tener un registro dentro del aplicativo web de las notificaciones emitidas al detectarse intrusos en las viviendas registradas en el sistema, para constatar que las alertas fueron recibidas por el personal de seguridad.	
Prioridad: Baja	**Esfuerzo:** 3
Criterios de Aceptación:	
• Al dar clic sobre el botón “REGISTROS” que se encuentra en el menú principal de la aplicación web, se mostrará los registros de todas las notificaciones de alertas emitidas por la detección de intrusos en una tabla con los siguientes campos: - Número de registro	

- Nombre de usuario - Fecha - Hora - Acciones • Cuando el administrador presione el botón “VER DETALLES” podrá visualizar en una interfaz la información de la notificación que fue emitida.

Tabla 36. Historia de usuario HE03-05 para el Sprint 4

El Anexo 2, muestra el Sprint Backlog considerado para cada uno de los Sprints. Además, de presentar el Sprint Review que permitió brindar transparencia sobre el incremento del producto.

CAPÍTULO IV

4. IMPLEMENTACIÓN DEL PROTOTIPO DE RED

4.1. RED LoRaWAN

Esta sección presenta la configuración e integración de los componentes LoRa con los servidores ChirpStack, los cuales permitirán recibir y publicar los mensajes de actos de intrusión detectados por los nodos, satisfaciendo de esta manera las historias de usuario HE01-01 y HE01-02.

4.1.1. Nodos detectores de intrusión

Antes de detallar la configuración del nodo es importante mencionar que en el Anexo 3, se encuentran el código fuente utilizado para el funcionamiento de los nodos detectores de intrusión.

4.1.1.1. Instalación y configuración del tablero Heltec

La programación de la tarjeta de desarrollo WiFi LoRa 32 (V2) se la realizó en el IDE de Arduino, mismo que necesita ser previamente configurado para poder trabajar (compilar código) con esta placa. Por ende, como primer punto es necesario describir el proceso realizado en dicha configuración, el cual se basa en la instalación del complemento Heltec ESP32 y la definición de los parámetros regionales de LoRaWAN.

Teniendo instalada y abierta la última versión del IDE Arduino disponible en su sitio oficial [94], se procede a colocar en el campo “Gestor de URLs Adicionales de Tarjetas”, ubicado en la ventana *Archivo → Preferencias*, la URL del repositorio que contiene los datos de la tarjeta WiFi LoRa 32 (V2): https://resource.heltec.cn/download/package_heltec_esp32_index.json, ver Figura 22.

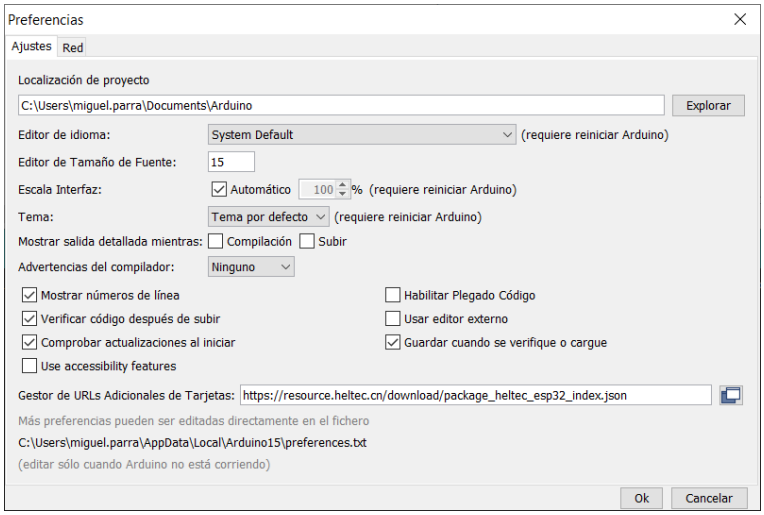


Figura 22. Agregación del repositorio Heltec en Arduino IDE

Lo realizado en el anterior paso indica al IDE en qué lugar debe buscar la tarjeta de desarrollo. Sin embargo, para poder utilizarla se la tiene que instalar mediante el "Gestor de tarjetas" localizado en el menú *Herramientas → Placa*. Dentro de esta ventana se coloca el nombre de la tarjeta a instalar, que en este caso es la placa WiFi LoRa 32 (V2), ver Figura 23. Se debe considerar que, al instalarla, todas las tarjetas que conforman el paquete de la serie Heltec también serán agregadas.

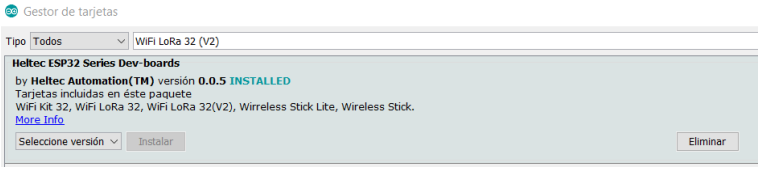


Figura 23. Instalación de la tarjeta WiFi LoRa 32 (V2)

Para finalizar la configuración se tiene que seleccionar, dentro del mismo menú, la región LoRaWAN que utilizará la placa, que en este caso es *REGION_US915,* ver Figura 24. Esta configuración permitirá que el nodo se comunique, mediante LoRa, dentro de la banda ISM 902-928 MHz.

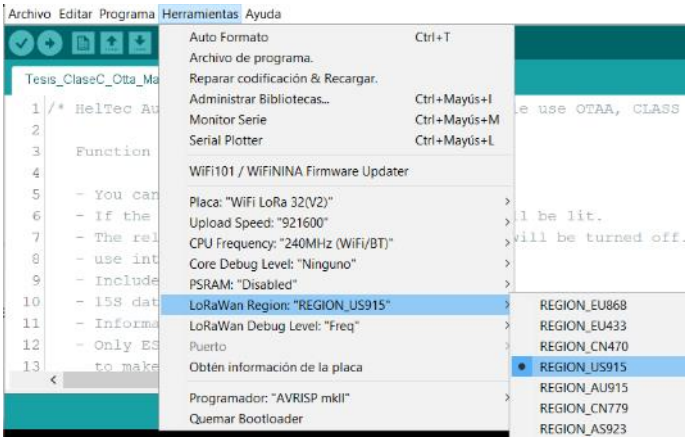


Figura 24. Región LoRaWAN utilizada por la placa

4.1.1.2. Utilización de la librería ESP32_LoRaWAN

Como se mencionó en la sección 3.1.1.2, la tarjeta de desarrollo tiene una propia librería de la pila LoRaWAN, denominada ESP32_LoRaWAN, que puede ser descargada en formato zip desde GitHub [95] o instalada directamente desde el administrador de bibliotecas de Arduino. La librería se basa en la versión 1.0.2 del protocolo LoRaWAN, y únicamente puede utilizarse en los tableros de la serie Heltec ESP32 y ESP8266, ya que necesita de una licencia que se encuentra relacionada con el ID del chip de la placa de desarrollo [96].

Con la funcionalidad ofrecida por esta librería se logró establecer dentro del sketch creado la siguiente configuración al nodo:

- Nodo de clase C para recibir en cualquier momento los mensajes de enlace descendente.
- Utilización de canales 0-7 del plan de frecuencias de la banda ISM US902-928 para la transmisión de enlaces ascendentes.
- Uso de OTAA para la activación del dispositivo en la red.
- Activación de mensajes confirmados para constatar que el mensaje emitido por el nodo llego al servidor de red.
- Activación del método ADR (Adaptive Data Rate) para que el servidor pueda optimizar la velocidad de datos de los nodos.

4.1.1.3. Estados notificados por los nodos

Los nodos se encuentran configurados para colocar en el payload del paquete LoRaWAN uno de cuatro posibles valores diferentes, ver Tabla 37, los cuales le permiten al backend conocer el estado en el que los nodos se encuentran y realizar ciertas configuraciones o emitir notificaciones a los usuarios.

Estado	Significado	Descripción
1	Nodo conectado en la red	Mensaje enviado cada 3 minutos. En el caso de no ser recibido en el backend, este notifica al administrador y al dueño del inmueble que el nodo se ha desconectado de la red.
2	Detección de intrusión	Mensaje enviado únicamente cuando los sensores detectan algún acto de intrusión y el nodo se encuentra en estado activo. De lo contrario, el mensaje no es enviado.
3	Nodo activado	Mensaje enviado inmediatamente después de recibir un mensaje de enlace descendente que solicita la activación del envío de notificaciones LoRa.
4	Nodo desactivado	Mensaje enviado inmediatamente después de recibir un mensaje de enlace descendente que solicita la desactivación del envío de notificaciones LoRa.

Tabla 37. Estados reportados por los nodos detectores de intrusión.

A continuación, se describe brevemente la programación y métodos utilizados en cada uno de los mensajes de estado presentados en esta tabla.

- **Estado 1:** para establecer el envío programado de los mensajes de conexión a la red se utilizó la función *millis()* de Arduino. Esta función permite obtener los milisegundos transcurridos desde que la tarjeta fue encendida. La función empieza a ejecutarse cuando el nodo se une a la red LoRaWAN, verificando en cada *loop()* la diferencia existente entre el tiempo actual (variable actualMillis) y el último tiempo guardado (variable previoMillis), si este llega a ser igual o superior a 3 minutos, se coloca 1 en el payload del mensaje y se lo envía mediante LoRa. Inmediatamente

después de esta acción se le reasigna el tiempo actual a la variable “previoMillis”. Este proceso se ejecuta indefinidamente mientras el nodo se encuentre encendido.

- **Estado 2:** mediante el uso de la función *digitalRead()* de Arduino se puede leer el valor del pin digital con el que se comunica el sensor. Si este valor es igual a HIGH se envía un mensaje mediante LoRa notificando la detección de intrusión. Se debe recordar que el mensaje es enviado únicamente si el nodo se encuentra habilitado para notificar actos de intrusión.
- **Estado 3:** al momento que un mensaje de enlace descendente es recibido por el nodo se ejecuta la función *downLinkDataHandle()* de la librería ESP32_LoRaWAN, la que permite verificar el payload emitido por la aplicación móvil. Si este valor es igual a 1 (ASCII) se habilita al nodo para que pueda enviar mensajes de detección de intrusos. Junto con esta acción se ejecuta una función que permite guardar, en la memoria EEPROM de la tarjeta, el nuevo estado del nodo. El propósito de este proceso es que el nodo pueda conservar dicho estado en el caso de un reinicio intempestivo de la placa. Para lograrlo se utilizaron los métodos *write()* and *read()* de la librería EEPROM.h. Adicional a esto el nodo envía una respuesta al backend para informarle que cambio su estado ha activado.
- **Estado 4:** se utiliza la misma función que en el anterior estado, solo que en este caso se verifica que el valor presente en el payload sea igual a 2 (ASCII). Al verificarlo el nodo se coloca en estado desactivado. De igual manera, este se encarga de enviarle una respuesta al backend para informarle que su estado ha cambiado.

4.1.1.4. Construcción de los circuitos

Dado que el prototipo utiliza dos tipos de sensores para la detección de intrusos, es conveniente presentar el circuito electrónico que se ocupó en cada uno.

4.1.1.4.1. Nodo detector de movimiento

Cuando el nodo se encuentre activado permitirá detectar la presencia de un intruso en base al cambio de radiación infrarroja que provocará cuando se mueva o ingrese en el entorno donde se encuentre instalado el sensor. Para la implementación de este componente se utilizó el sensor de movimiento PIR HC-SR501 junto con la placa WiFi LoRa 32 (V2).

Como se mencionó en 3.1.1.2, el sensor consta de 3 pines: uno para VCC, uno para GND y uno para la comunicación con la placa de desarrollo. Este último varia cuando el sensor detecta movimiento, es decir, genera un pulso digital alto de 3.3 V, de lo contrario permanece bajo con 0 V. La conexión de los pines entre el sensor y el tablero de desarrollo puede ser visualizado en la Tabla 38.

HC-SR501	WiFi LoRa 32 (V2)
VCC	VCC (5V)
GND	GND
Dout	GPIO22

Tabla 38. Pines de conexión entre el sensor HC-SR501 y la placa WiFi LoRa 32 (V2)

Al recibir una señal alta en el GPIO22, se comprobará que el nodo este activado para notificar dicha detección, si este es el caso se cocola un 1 en el payload del mensaje y se lo envía al gateway usando la tecnología de comunicación LoRa.

El sensor se configuro, mediante sus potenciómetros, para que funcione con un periodo de activación de 3 segundos (tiempo activo del pin de salida) y una sensibilidad de detección de 7 metros. La Figura 25, presenta el circuito electrónico del nodo detector de movimiento.

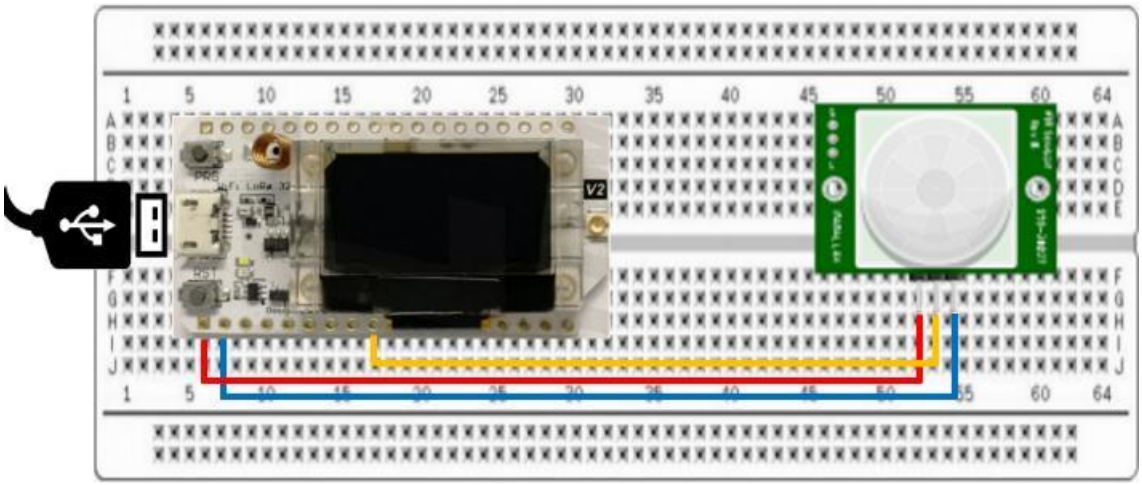


Figura 25. Circuito electrónico del nodo detector de movimiento

4.1.1.4.2. Nodo detector de apertura de puertas o ventanas

Cuando el nodo se encuentre activado permitirá detectar un intruso en base a la apertura de la puerta o ventana en la que se encuentre instalado, ya que esta acción provoca que el campo magnético existente entre el imán y el interruptor desaparezca haciendo que el estado del sensor cambie. Para la implementación de este componente se utilizó el sensor magnético MC-38 junto con la placa WiFi LoRa 32 (V2).

Al ser un sensor de tipo normalmente cerrado arrojará un valor de 0 cuando las dos partes se encuentren juntas y un 1 cuando estas se separen, ver Figura 26. La conexión de los pines entre el sensor y el tablero de desarrollo puede ser visualizado en la Tabla 39.

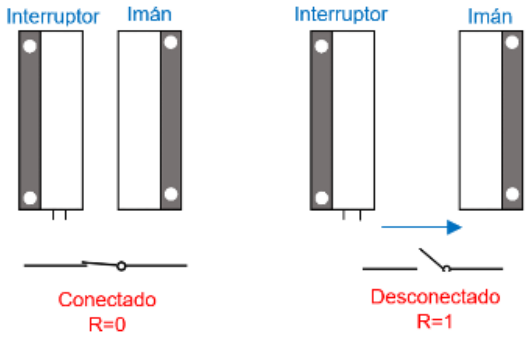


Figura 26. Funcionamiento del interruptor normalmente cerrado (NC)

HC-38	WiFi LoRa 32 (V2)
1	GND
2	GPIO22

Tabla 39. Pines de conexión entre el sensor MC-38 y la placa WiFi LoRa 32 (V2)

Al detectarse un valor 1 en el GPIO22, se verifica que el nodo se encuentre en estado activo, si este es el caso se cocola un 1 en el payload del mensaje y se lo envía a la puerta de enlace utilizando LoRa. Es importante mencionar que a este pin se lo configuro en modo INPUT_PULLUP para permitir la activación de una gran resistencia limitadora de corriente, y así evitar que el circuito de la placa sufra algún daño. La Figura 27 presenta el circuito electrónico del nodo en cuestión.

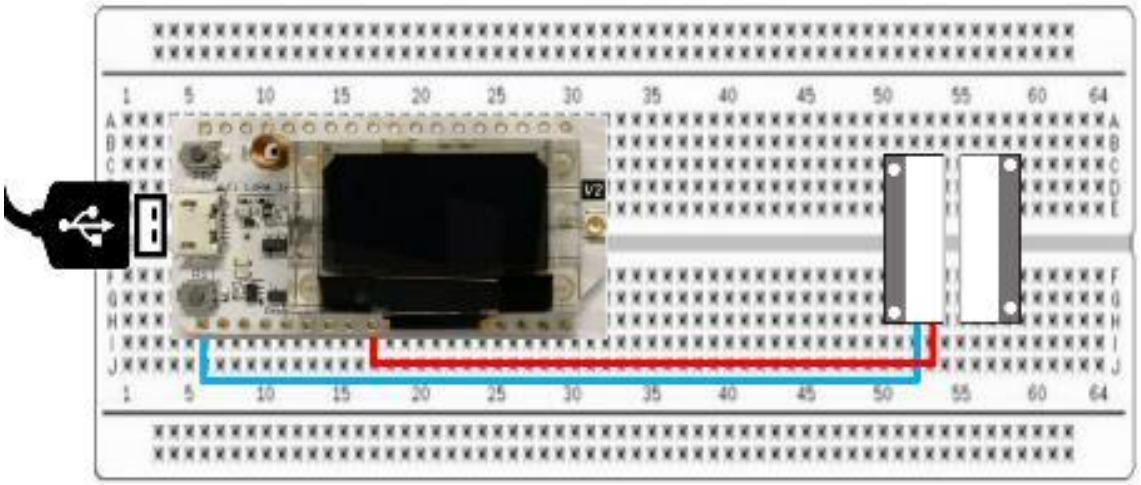


Figura 27. Circuito electrónico del nodo detector de apertura de puertas o ventanas

4.1.1.5. Mensaje enviado por el nodo

Con el fin de conocer la carga útil enviada por los dispositivos finales, así como otros datos relacionados a la transmisión del mensaje LoRa, se imprimió dicha información dentro del monitor serial de Arduino, ver Figura 28.

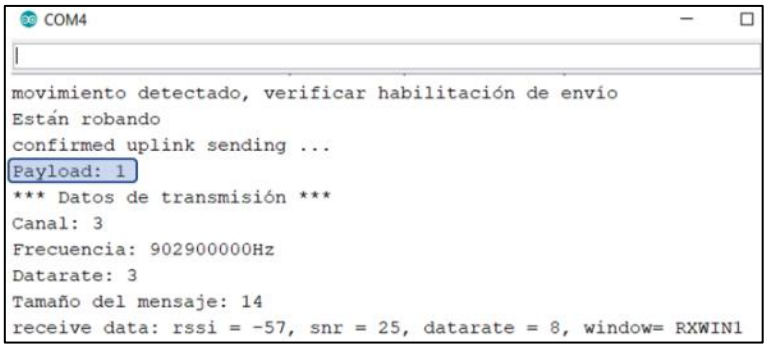


Figura 28. Datos de transmisión LoRa y contenido de la carga útil enviada por el nodo

En este caso se puede observar que al detectarse algún acto de intrusión se imprime un mensaje de alerta, pero para poder enviarlo mediante LoRa primero se verifica si el nodo se encuentra activo, si este es el caso imprime el mensaje “Están robando” y envía el paquete colocando 1 en el payload (cuadro azul). De igual manera se imprimen los datos de transmisión utilizados para enviar dicho mensaje.

4.1.2. Puerta de enlace RAK7258

Antes de encender al equipo es importante verificar que la antena LoRa se encuentre correctamente conectada para evitar posibles daños en el gateway. Para conectarse a este elemento existe la posibilidad de hacerlo mediante Wi-Fi o Ethernet. Además, su configuración puede realizarse a través de línea de comandos o desde la interfaz web de usuario, ver Figura 29, en ambos casos se necesita proporcionar las credenciales del administrador para ingresar al sistema.

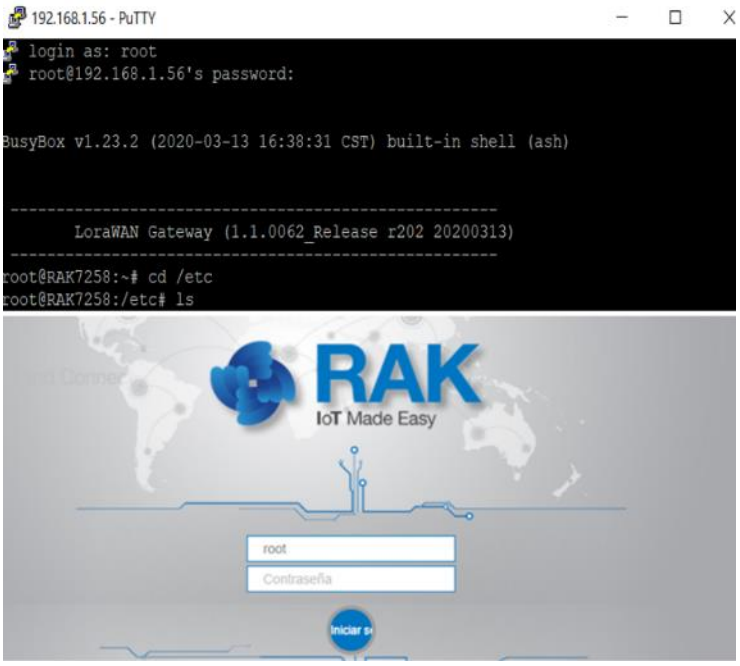


Figura 29. Interfaces para la configuración de la puerta de enlace RAK7258: línea de comandos y web

4.1.2.1. Configuración del reenviador de paquetes

Para poder recibir y retransmitir los mensajes emitidos por los nodos, se configuró el reenviador de paquetes (packet forwarder) desde la interfaz web del gateway. Dentro de este apartado, se debe colocar la dirección IP del servidor LoRaWAN al que se reenviará los paquetes, en este caso la dirección del servidor Chirpstack, y el puerto por defecto 1700. Ahora, con el afán de que el gateway "escuche" los mensajes de los nodos LoRa, se estableció el mismo plan de frecuencias que estos utilizan, es decir, el plan correspondiente a la Región US902-928 en los canales 0-7. Esta configuración se muestra en la Figura 30.

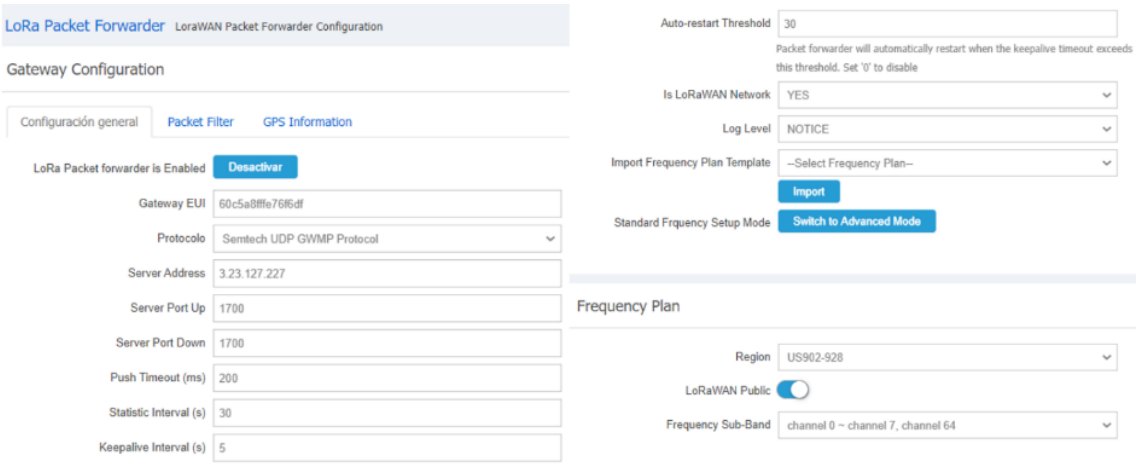


Figura 30. Configuración del reenviador de paquetes LoRa a través de la interfaz web

En el caso de querer configurar directamente al packet forwarder desde la línea de comandos, se debe ingresar al directorio /etc/config y modificar el contenido del archivo lora_pkt_fwd, tal como se muestra en la Figura 31.

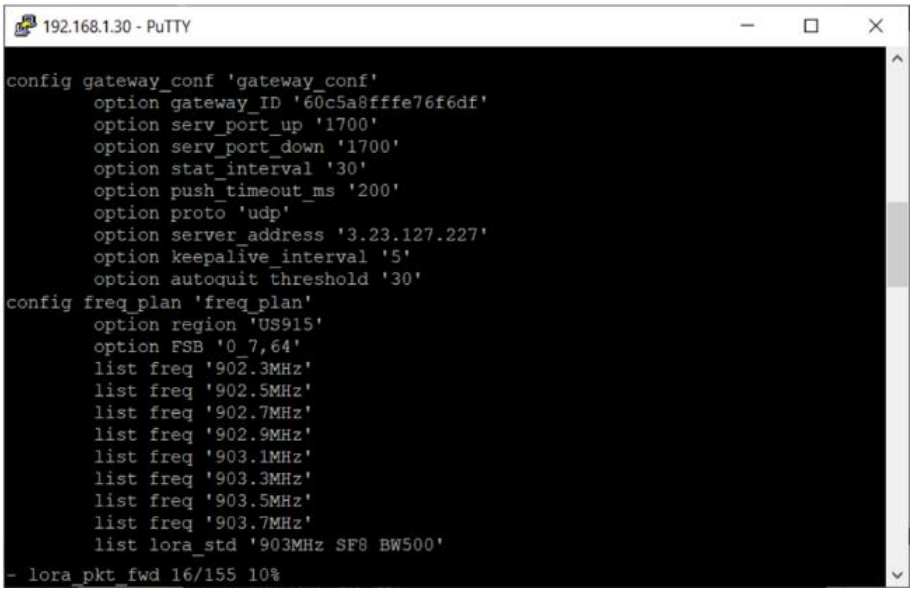


Figura 31. Configuración del reenviador de paquetes desde la línea de comandos

4.1.2.2. Visualización de paquetes

Para visualizar los paquetes que llegan a la puerta de enlace se hace uso de la sección “Registrador de paquetes LoRa”. Aquí se registran en tiempo real 2 tipos de mensajes: ascendentes y descendentes. La Figura 32, presenta el registro en cuestión, en este caso se logró captar el paquete que el nodo envía para solicitar su unión a la red LoRaWAN (color naranja).

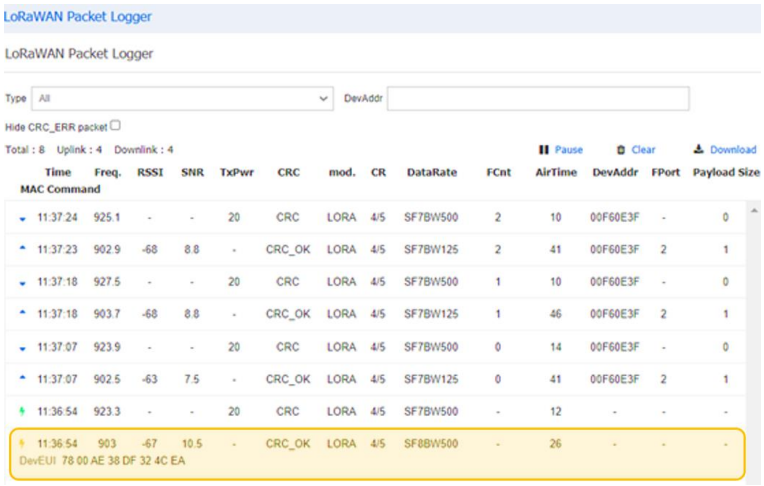


Figura 32. Mensajes LoRaWAN registrados por el RAK7258

Como se puede apreciar a cada mensaje de enlace ascendente le sigue uno de enlace descendente, esto sucede porque los nodos tienen activado la opción "mensaje confirmado", por ende, el servidor de red LoRaWAN debe responder cada paquete que recibe del nodo. La Figura 33, presenta dicha opción dentro la estructura del menaje de enlace ascendente LoRaWAN (azul) junto con los datos de transmisión LoRa (verde).

Figura 33. Estructura del mensaje de enlace ascendente LoRaWAN

4.1.3. Servidores ChirpStack

4.1.3.1. Acceso a la Instancia Amazon EC2

Dentro del prototipo planteado, los servidores ChirpStack se encuentran corriendo en una instancia t2.micro EC2 de Amazon Web Services con Ubuntu Server 18.04 LTS, ver Figura 34. Este tipo de instancias tienen un rendimiento base de CPU y son gratuitas durante un año, ofreciendo 750 horas de uso al mes por uno o más servicios.

Figura 34. Resumen de la instancia Amazon EC2

El acceso a la instancia se la realiza mediante PuTTY a través de una conexión SSH, para ello se necesita las credenciales de acceso en formato ppk, mismas que se pueden obtener desde el panel de control de Amazon en la opción “Pares de claves”. Al iniciar la comunicación también se debe colocar el nombre de usuario y el DNS público de la instancia, ver Figura 35.

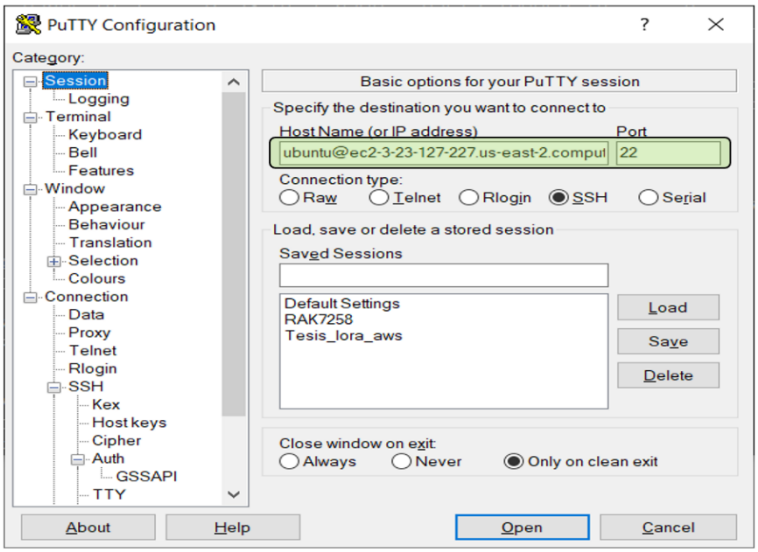


Figura 35. Acceso a la instancia EC2 mediante PuTTY

4.1.3.2. Instalación y configuración de ChirpStack

Dentro de la instancia creada se instalaron los servidores ChirpStack y todas sus dependencias mediante la ejecución del script “install.sh”, que fue obtenido al clonar el repositorio RAKwireless Ubuntu ChirpStack [97] mediante el comando git clone, previo a esto se debe instalar la tecnología GIT ejecutando el comando sudo apt install git.

Ahora bien, para que el servidor de red de ChirpStack funcione correctamente con el resto de componentes, se establece la banda de frecuencia ISM 902_928 junto con los canales 0-7 en el archivo de configuración “chirpstack-network-server.toml”, ver Figura 36, por defecto este se encuentra configurado para trabajar con las frecuencias ISM 863_870 de Europa.

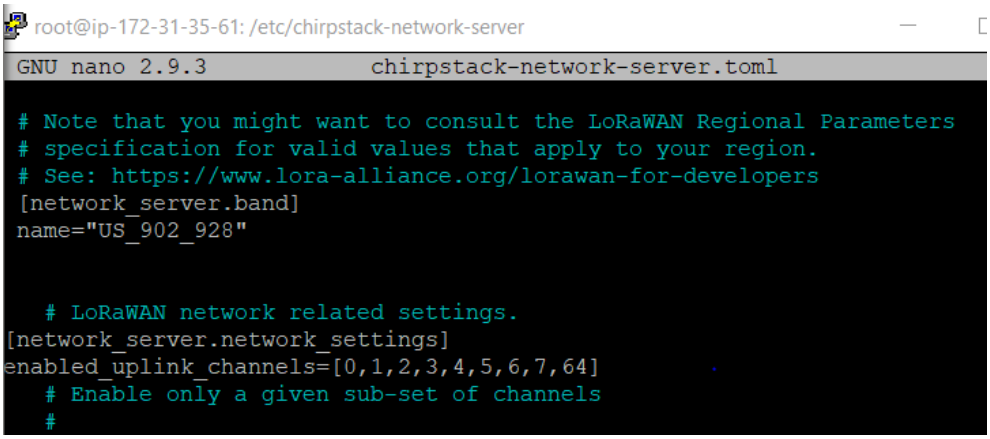


Figura 36. Configuración de la banda de frecuencia del servidor de red ChirpStack

Por otra parte, en el archivo de configuración del servidor de aplicaciones de ChirpStack se debe añadir una clave secreta, la cual permite firmar los JWTs (Json Web Tokens) utilizados para la autenticación del cliente y dar acceso a los métodos de la API de este servidor. La clave secreta puede ser generada ejecutando el comando *openssl rand -base64 32*, que posteriormente es colocado en el campo “jwt_secret” del archivo “chirpstack_application_server.toml”, tal como se muestra en la Figura 37.

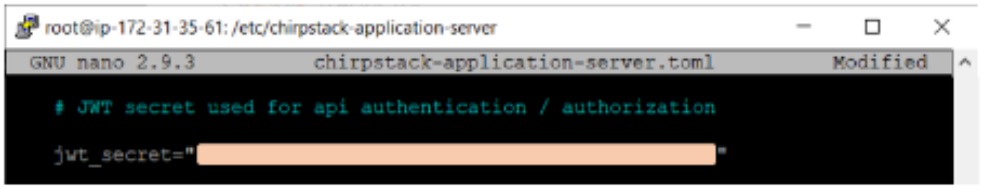


Figura 37. Asignación del secreto JWT para la autenticación de API

Para aplicar los cambios en las configuraciones realizadas se debe reiniciar los servicios de ambos servidores con los comandos *systemctl restart chirpstack-network-server* y *systemctl restart chirpstack-application-server.*

4.1.3.3. Configuración de la seguridad de AWS

Casi todo el tráfico de entrada a las instancias AWS se encuentra bloqueado por defecto, únicamente el puerto 22 (utilizado para conexiones SSH) permanece abierto. Debido a esto, es necesario agregar un conjunto de reglas de entrada que permiten habilitar los puertos que tanto el gateway como el servidor de red necesitan para comunicarse. Esto se lo realiza en la opción "Grupo de seguridad" del panel de AWS. Los puertos por habilitar son:

- Puerto UDP 1700, utilizado por el reenviador de paquetes;
- Puerto TCP 1883, utilizado por MQTT;
- Puerto TCP 8080, para acceder a la interfaz web de ChirpStack.

La Figura 38, presenta la regla de entrada establecida con los puertos mencionados.

Figura 38. Reglas de entrada del grupo de seguridad

4.1.3.4. Registro y configuración de la puerta de enlace

Con el fin de que el servidor de red de ChirpStack gestione los paquetes transmitidos por la puerta de enlace, esta debe ser añadida en el servidor a través de su interfaz web. Para esto es necesario ingresar en una ventana de cualquier navegador la dirección IP pública de la instancia AWS (3.23.127.227) junto con el puerto 8080. Dentro de la interfaz web de usuario, en la pestaña “Gateways”, se registra a la puerta de enlace colocando la siguiente información:

- Nombre: campo que solo puede contener palabras, números y guiones.
- Descripción: permite conocer algún dato adicional acerca de la puerta de enlace u ofrece una breve explicación de la misma.
- ID: identificador único de la puerta de enlace en la red. Esta información puede ser obtenida dentro del reenviador de paquetes en el campo Gateway EUI.
- Servidor de red: servidor al que se conecta la puerta de enlace. Previo a esto se debe crear un perfil de servicio asignándole un servidor de red que aprovisionará este perfil, los metadatos (SNR, RSSI, etc) que serán enviados al servidor de

aplicación, la frecuencia de solicitud de estado del dispositivo y el data rate mínimo y máximo permitido.

- Altitud: altura aproximada a la que se encuentra ubicada la puerta de enlace. Este campo debe ser colocado en metros.
- Ubicación: lugar donde se encuentre instalada la puerta de enlace. Para esto se debe arrastrar un marcador dentro del mapa.

Una vez establecido el registro, el servidor de red empezará a recibir los paquetes de la puerta de enlace. Si el proceso se realizó con éxito, se creará un campo que permitirá visualizar la última vez que el servidor ChirpStack recibió mensajes del gateway, ver Figura 39.

Figura 39. Detalles de la puerta de enlace registrada en ChirpStack

4.1.3.5. Registro y configuración de los nodos

Para poder registrar los nodos es necesario añadir previamente un perfil de dispositivo. Esto se lo realiza dentro de la pestaña “device-profiles”, en la cual se colocó lo siguiente:

- Un nombre que identifique al nuevo perfil.
- La versión del protocolo LoRaWAN soportada por el dispositivo, en este caso 1.0.2.
- La revisión de la especificación de los parámetros regionales que admitirá el nodo.
- El tipo de activación OTAA.
- Dispositivo de clase C
- Un Códec que permite al servidor de aplicaciones ChirpStack decodificar la carga útil binaria del dispositivo, ver Anexo 4, convirtiéndola en formato JSON.

De igual manera, se debe agregar una aplicación que contendrá a los nodos detectores de intrusión, esto se lo realiza dentro de la pestaña “Aplicaciones”, asignándole un nombre, una descripción y el mismo perfil de servicio utilizado por la puerta de enlace. Una vez hecho esto, se podrá registrar al nodo colocando la siguiente información:

- Nombre del nodo, que solo puede contener palabras, números y guiones.
- Descripción del dispositivo.
- DevEUI, que identificará de manera única al nodo en la red. Este campo puede ser generado aleatoriamente desde el formulario y tiene que ser colocado dentro del sketch creado en Arduino.
- El perfil de dispositivo creado anteriormente.
- Desmarcar la opción “disable frame-counter validation”, esto se lo realiza para evitar ataques de repetición.

Para que el nodo pueda comunicarse y usar la activación OTAA es obligatorio modificar el campo “Application key”, que se encuentra ubicado en la pestaña “KEYS (OTAA)” dentro del dispositivo registrado. La clave de aplicación es de 128 bits y puede ser generada aleatoriamente, lo importante en este punto es que esta clave también debe ser colocada en el sketch creado en Arduino.

Finalmente, para verificar que el nodo fue configurado correctamente, en la pestaña "Device Data", deben empezar a registrarse en vivo los paquetes que el dispositivo envía, tal como se muestra en la Figura 40.

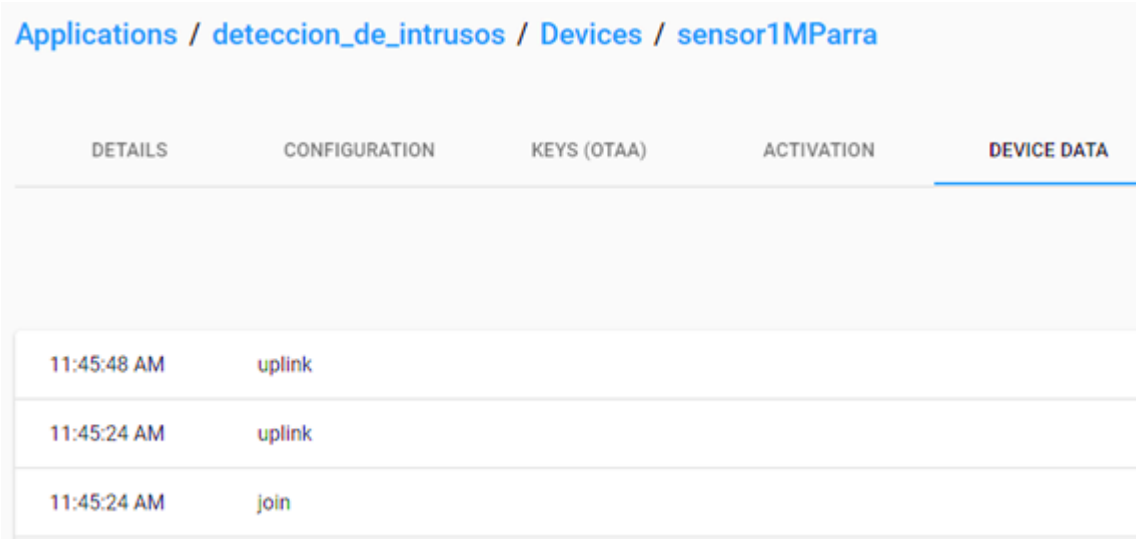


Figura 40. Registro de los paquetes enviados por el nodo a los servidores ChirpStack

Si se desea ver el contenido del mensaje receptado por el servidor de aplicación, debe seleccionar algún paquete para desplegar toda la información vinculada al mismo. Su formato es de tipo JSON, tal como se muestra en la Figura 41.

Figura 41. Información JSON enviada por el nodo detector de intrusión

4.1.3.6. Integraciones con la aplicación cliente

4.1.3.6.1. MQTT

El servidor de aplicación ChirpStack publica, en formato JSON, los datos enviados por los nodos en el tópico MQTT "application/1/device/+/rx", ver Figura 42. Esto se lo realiza con el afán de que la aplicación cliente pueda utilizar los datos que se publican mediante la suscripción al tópico mencionado.

Figura 42. Mensaje publicado en el tópico MQTT

Para evitar que cualquiera persona se suscriba o publique en algún tema en específico, se estableció la autenticación y autorización basada en el uso de contraseñas estáticas y listas de control de acceso (ACL). Para esto se hizo uso del comando mosquitto_passwd /etc/mosquitto/passwd nombreUsuario que permite asignar una contraseña a un usuario específico. Por su parte, los usuarios junto con los permisos de publicación o suscripción a los distintos tópicos deben ser añadidos en el archivo /etc/mosquitto/acls. Finalmente, para que el bróker Mosquitto empiece a trabajar con este tipo de autenticación, es

necesario agregar un nuevo archivo de configuración en /etc/mosquitto/conf.d/aut.conf, el cual contendrá la información presentada en la Figura 43.

```
allow_anonymous false
password_file /etc/mosquitto/passwd
acl_file /etc/mosquitto/acls
```

Figura 43. Configuración del archivo aut.conf de Mosquitto

4.1.3.6.2. Utilización del API Rest

Para la integración con la aplicación cliente el servidor ChirpStack ofrece entre muchas opciones un API RESTful. Para poder utilizar los métodos provistos por la API se necesita de un token. Este token tiene una duración de 1 día y se obtiene utilizando el método POST "/api/internal/login" presente en la consola del API localizada en la interfaz web de ChirpStack, la Figura 44, presenta dicho método.

Figura 44. Obtención del JWT desde la consola del API RESTful

4.2. APLICACIONES WEB Y MÓVIL

4.2.1. Backend del sistema

Para satisfacer la historia de usuario HE02-01 que tiene como descripción capturar, interpretar y procesar los datos obtenidos a través de las aplicaciones web, móvil y del servidor de aplicaciones de ChirpStack, se ha implementado un backend mediante el uso del framework sails.js, que fue instalado por medio del comando *npm install sails –g*. El cual permitirá realizar peticiones, efectuar respuestas HTTP e integrarse con la base de datos.

En la Figura 45, se presenta las carpetas que conforman la estructura del backend del sistema.

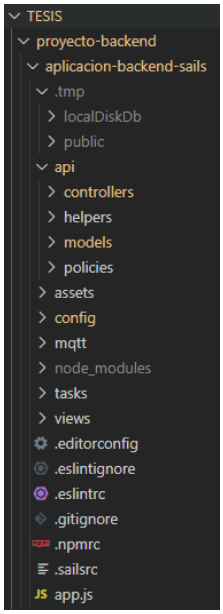


Figura 45. Estructura del backend del sistema

A continuación, se detalla brevemente el contenido de estas carpetas:

- **Controllers**: dentro de esta carpeta se encuentran los controladores que son archivos de tipo JavaScript o TypeScript, y que sirven para acceder a los métodos HTTP, mismos que permiten gestionar la información que será almacenada en la base de datos, ver Figura 46.

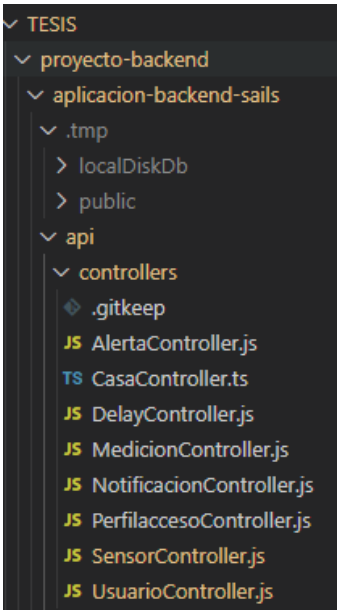


Figura 46. Estructura del backend del sistema, carpeta “Controllers”

- **Models:** dentro de esta carpeta se encuentran los modelos que son archivos de tipo TypeScript y que representan las tablas de la base de datos, como: Perfil de Acceso, Usuario, Casa, Sensor, Notificación y Alerta, ver Figura 47.

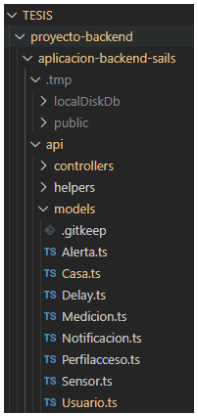


Figura 47. Estructura del backend del sistema, carpeta “Models”

- **Config:** dentro de esta carpeta se encuentran los archivos de configuración del backend, como: “datastore.js” (almacén de datos), que sirve para configurar la conexión a la base de datos; “security.js”, que sirve para la seguridad del sistema, es decir, controlar el acceso HTTP para gestionar el permiso a dominios o hosts que deseen cargar datos o acceder a la información desde servidores ajenos, ver Figura 48; y “routes.js”, que sirve para gestionar las solicitudes entrantes al sistema, es decir, cuando se realiza una solicitud a la aplicación, el framework sails.js toma todos los parámetros que vienen con ella y los pone a disposición como parámetros o params en el objeto de solicitud con el fin de utilizarlos en los respectivos controladores, ver Figura 49.

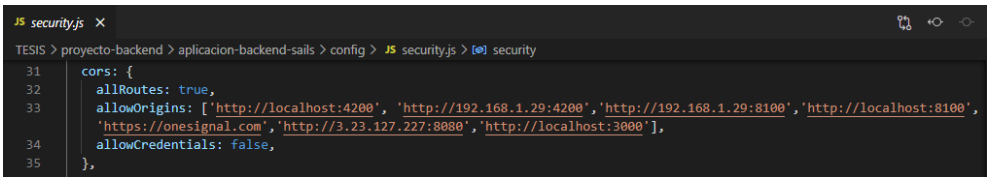


Figura 48. Permisos establecidos dentro del Archivo “security.js”

Figura 49. Configuración del Archivo “routes.js”

Después de realizar la instalación del framework se procede a levantar y configurar la base de datos en la herramienta Docker, la cual permite crear contenedores ligeros para ejecutar varios procesos y aplicaciones por separado, ver Figura 50. En este caso se ha seleccionado el gestor de base de datos MySQL.

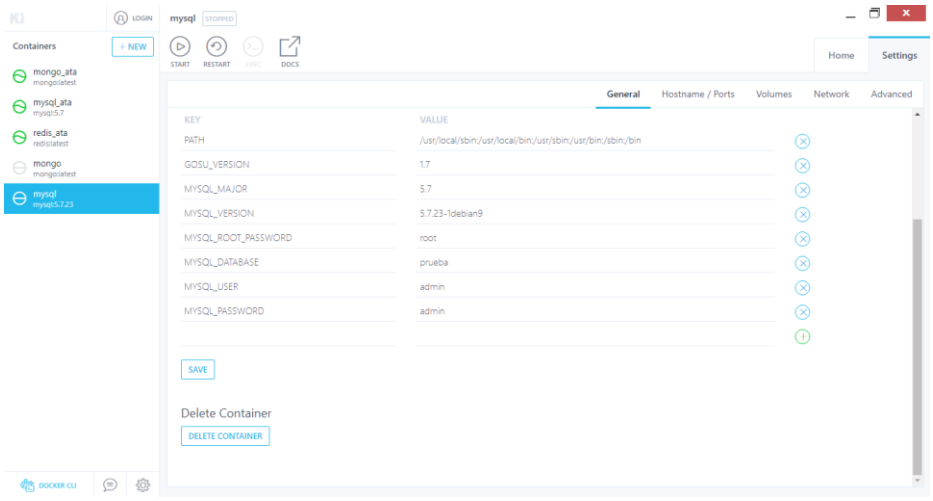


Figura 50. Base de datos MySQL configurado en la herramienta Docker

Una vez que se ha efectuado el proceso de levantar y configurar la base de datos, se procede a integrarlo con el framework, para esto se hace se hace uso del archivo “datastores.js”, que establece la conexión con la base de datos por medio del adaptador de MySQL *sails-mysql*, que se instala mediante el comando *npm install sails-mysql --save* y la URL de conexión *protocol://user:password@host:port/database*.

A continuación, se describe la composición de la URL de conexión, Figura 51:

- **protocol:** es la sección de la URL que se basa en el adaptador que se va a utilizar (mysql://, mongodb://, etc.). El resto de los parámetros que compone la URL son credenciales que sirven para localizar y acceder a la base de datos.
- **user:** es el nombre de usuario mediante el cual se permite ingresar a la base de datos.
- **password:** es la contraseña que se ha establecido para ingresar a la base de datos.
- **host:** es la dirección IP o dominio donde se encuentra funcionando la base de datos.
- **port:** es el puerto por el cual se pueden enviar y recibir la información de la base de datos.
- **database:** es el nombre de la base de datos.

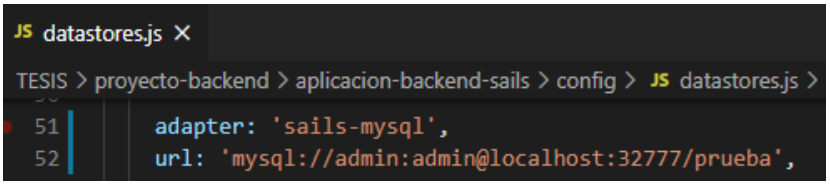


Figura 51. Estructura de la URL de conexión

Ahora, se procede a implementar el método que permitirá capturar los datos publicados por el broker MQTT provisto por ChirpStack. El método que se ha desarrollado es *suscribirMqtt*, el cual utiliza la librería MOSCA para suscribirse al tópico publicado por el broker Mosquitto de ChirpStack. La librería fue instalada por medio del comando *npm i mosca mqtt*.

En la Tabla 40, se detallan los métodos usados de la librería mosca en el sistema.

Método	Descripción
client = mqtt.connect ([url], opciones)	Sirve para conectarse a la URL especificada con las opciones proporcionadas. Este devuelve un cliente que se conecta al broker para enviar o recibir información.
client.on([connect], function)	Este método recibe como parámetros el evento “connect” que sirve para establecer y verificar la conexión y la función que se activa cuando se completa el evento. Se produce un error si el cliente se desconecta.
client.subscribe([topico])	Este método permite suscribirse a uno o varios tópicos, los cuales deben ser proporcionados como argumentos.
client.on([message], function)	Este método recibe como parámetros el evento “message” que sirve para recibir los mensajes publicados por el tópico.

Tabla 40. Métodos de la librería mosca

Para poder suscribirse a las publicaciones que emite el broker de ChirpStack, se procede a configurar la conexión mediante el método *connect* que recibe como parámetros un string de conexión que está constituida de la siguiente estructura:

- mqtt, protocolo de comunicación.
- url, dirección IP o dominio donde se encuentra alojado el servidor de aplicaciones de ChirpStack.
- 1883, puerto de conexión y las credenciales para efectuar la autenticación en el servidor (usuario y contraseña).

La Figura 52, presenta la utilización del método en cuestión.

```
let topicoString = `application/1/device/+/rx`;
client = mqtt.connect('mqtt://hostChirpStack:1883',{username:"usuario",password:"contraseña"});
client.on('connect', ()=>{
    client.subscribe(topicoString);
})
```

Figura 52. Parámetros utilizados para establecer la conexión con el broker MQTT de ChirpStack

Una vez se ha establecido la conexión con el broker provisto por Chirpstack, se procede a suscribirse al tópico haciendo uso del método *client.suscribe*, pasando como parámetro la siguiente estructura "*application/1/device/+/rx*", esto con el fin de escuchar los mensajes emitidos por los nodos al servidor LoRaWAN, ver Figura 53. El significado de los niveles que conforman el tópico mencionado es descrito a continuación:

- **application:** es el nivel que contiene a todas las aplicaciones registradas en el servidor de aplicación de ChirpStack.
- **1:** especifica el ID de aplicación a subscribirse.
- **device:** es el nivel que engloba a los dispositivos registrados dentro del ID de aplicación dado.
- **+:** especifica que se suscribirá a todos los dispositivos dentro de ese nivel.
- **rx:** permite determinar el tipo de evento a subscribirse, que en este caso corresponde al evento de recepción de mensajes de los dispositivos finales LoRa.

```
let topicoString = `application/1/device/+/rx`;
client = mqtt.connect('mqtt://hostChirpStack:1883',{username:"usuario",password:"contraseña"});
client.on('connect', ()=>{
    client.subscribe(topicoString);
})
```

Figura 53. Estructura del tópico

Después de haberse suscrito al tópico, se hace uso del método *client.on(message,function)* que permitirá recibir y manejar los datos publicados por el broker, ver Figura 54.

```
client.on('message', (topicoString, message)=>{
    const objRespuesta = JSON.parse(message);
    if(objRespuesta.deviceName){
        var arrIdTelf = [];
        const string = objRespuesta.deviceName;
        const codigoSensorCS = objRespuesta.devEUI;
        const usuCortado = string.slice(7, string.length);
        const payload = objRespuesta.object.lectura;
        switch (payload) {···
        }
    }
});
```

Figura 54. Método client.on(message,function)

4.2.2. Configuración de notificaciones

La interfaces que se muestran en la Figura 55 y Figura 56, son el resultado final de la pantalla que permite al administrador del sistema observar la notificación de alerta o aviso de desconexión emitido por los nodos finales a través de la aplicación web, de la misma manera, las interfaces que se muestran en la Figura 57 y Figura 58, son el resultado final de la pantalla que permitirá al usuario observar la notificación de alerta o aviso de desconexión que emiten los nodos finales al dispositivo móvil; mismas que a su vez satisfacen la historia de usuario HE02-02.

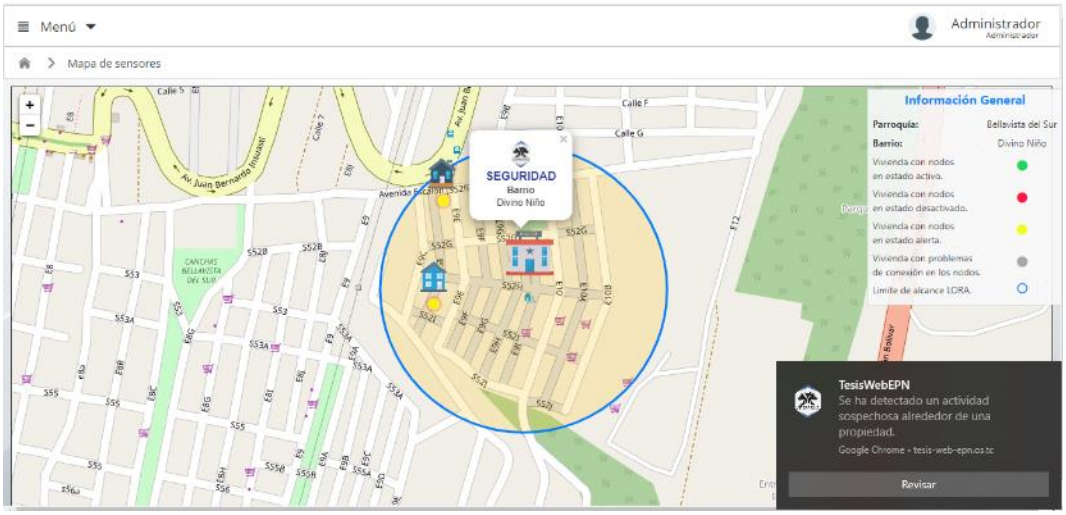


Figura 55. Interfaz de la aplicación web que muestra la notificación de alerta de detección de intrusos emitidos por los nodos de la vivienda

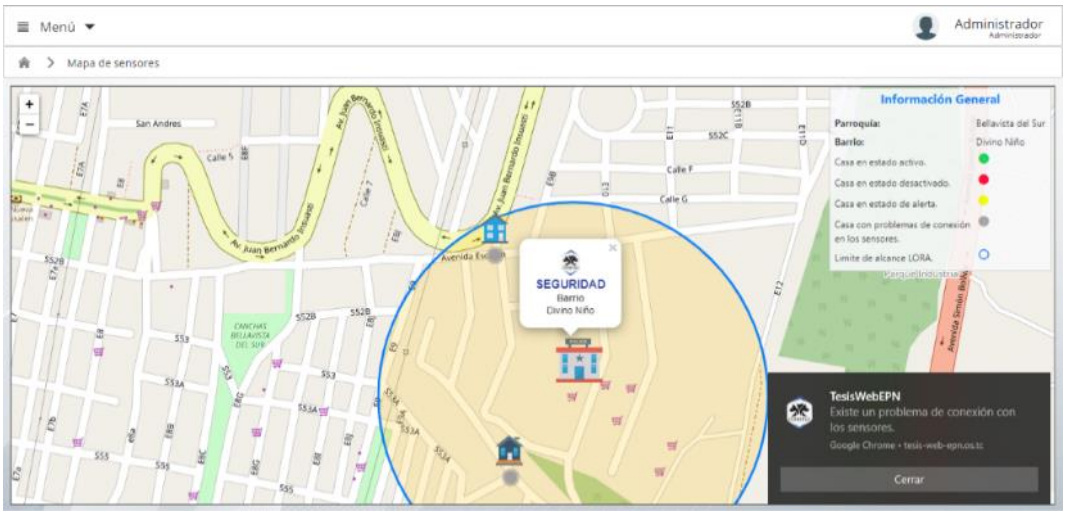


Figura 56. Interfaz de la aplicación web que muestra la notificación de alerta de desconexión de los nodos

Figura 57. Interfaz de la aplicación móvil que muestra los mensajes de alerta de detección de intrusos

Figura 58. Interfaz de la aplicación móvil que muestra los mensajes de desconexión de los nodos

Para la implementación de notificaciones se hizo uso del servicio de envío de notificaciones push OneSignal. Esta herramienta permite la integración con el backend de manera segura y sencilla mediante el uso de peticiones HTTP al API REST provista por la misma, ver Figura 59.

Aquí se ha implementado un método que incluye la librería axios y que servirá para realizar el envío de notificaciones a los respectivos usuarios que usan la aplicación web o móvil, ver Tabla 42. Se debe de tomar en cuenta que antes de hacer uso del API REST, es

necesario tener una cuenta registrada, ya que esta permitirá hacer uso de todas las funcionalidades que brinda este servicio de notificaciones.

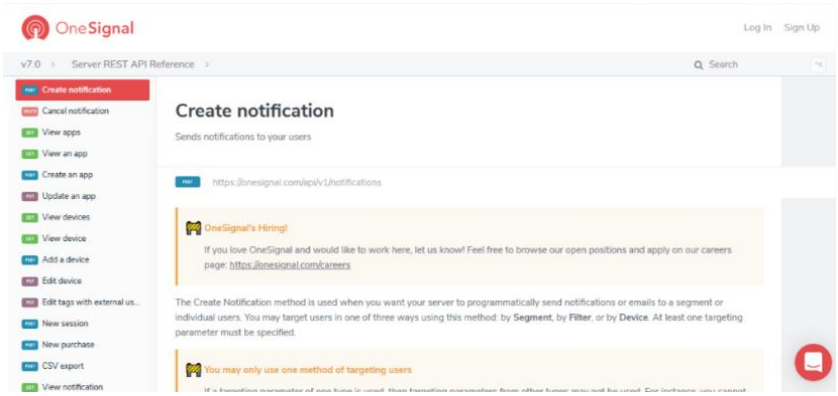


Figura 59. Documentación de la API REST del servicio de envío de notificaciones push OneSignal

Para la incorporación de OneSignal tanto en la aplicación web como en la móvil, se debe de realizar una serie de configuraciones documentadas en el sitio web de esta herramienta [98].

La integración con la aplicación web es muy sencilla, puesto que se debe de crear una nueva Aplicación/SitioWeb, ver Figura 60, asignarle un nombre y seleccionar la opción Web push, ver Figura 61. El Panel de OneSignal mostrará el tipo de integración que se desea utilizar, en este caso se usó la de configuración de código personalizado ya que permite a los desarrolladores tener control total sobre cómo funciona OneSignal, escribiendo código JavaScript personalizado, ver Figura 62.

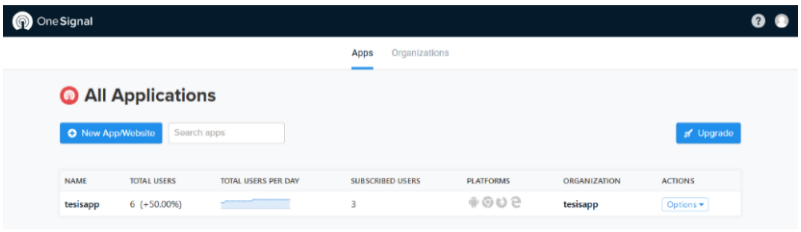


Figura 60. Panel de configuración de la herramienta OneSignal

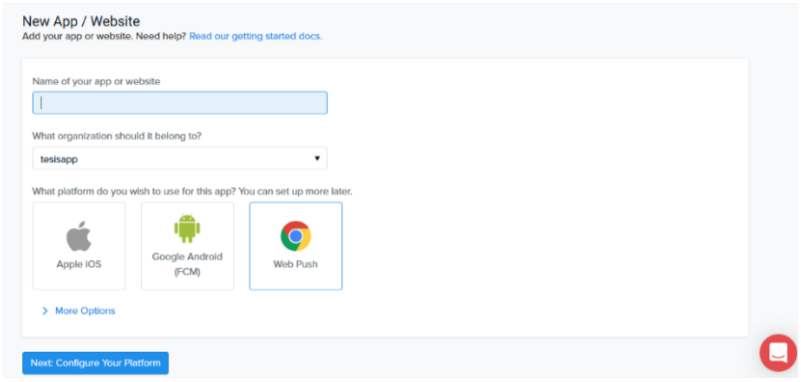


Figura 61. Interfaz de configuración de la herramienta OneSignal para asignar nombre de la aplicación

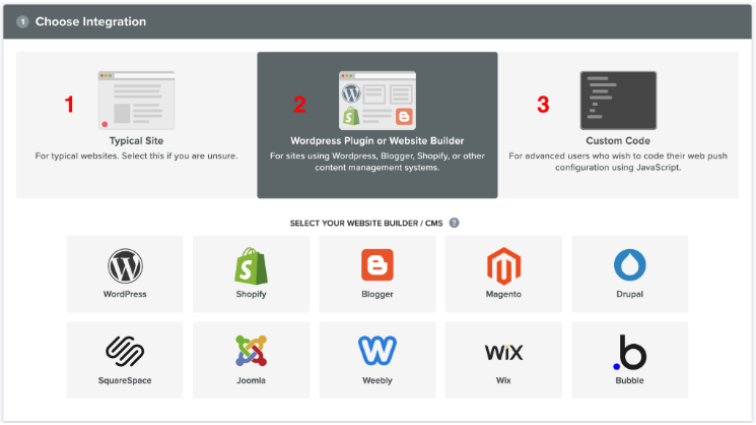


Figura 62. Interfaz de configuración de la herramienta OneSignal que permite obtener el código embebido para la integración con la aplicación web

Para la integración de OneSignal con la aplicación móvil es necesario hacer uso de una clave de API de Firebase Server que debe generarse en la plataforma de Firebase. Esto se lo hace con el fin de enviar notificaciones push a dispositivos que utilizan aplicaciones móviles Android. Para generar la clave es necesario disponer de una cuenta de Google, puesto que esta será solicitada al acceder a la página de Firebase console, ver Figura 63.

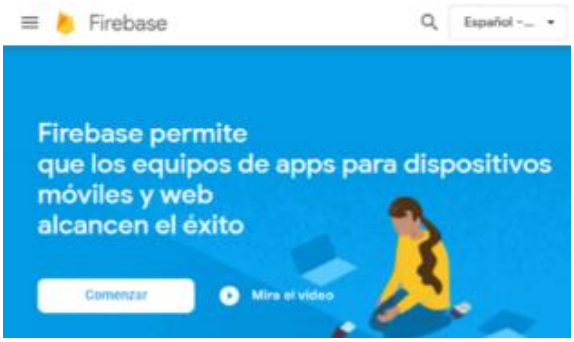


Figura 63. Consola de la herramienta Firebase

Al validarse el acceso a la consola se debe crear un nuevo proyecto, ver Figura 64.

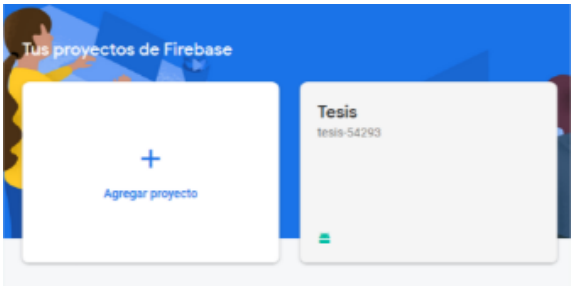


Figura 64. Consola de la herramienta Firebase para la creación de un nuevo proyecto

Una vez creado el proyecto, se tiene que dirigir al icono de ajustes, ubicado en la parte superior izquierda de la pantalla, y seleccionar “Configuración del proyecto”. Posterior a esto, se debe escoger la opción “CLOUD MESSAGING” que permitirá visualizar las llaves

“Server Key” y “Sender ID” que serán utilizadas para la configuración de la herramienta en el aplicativo, ver Figura 65.

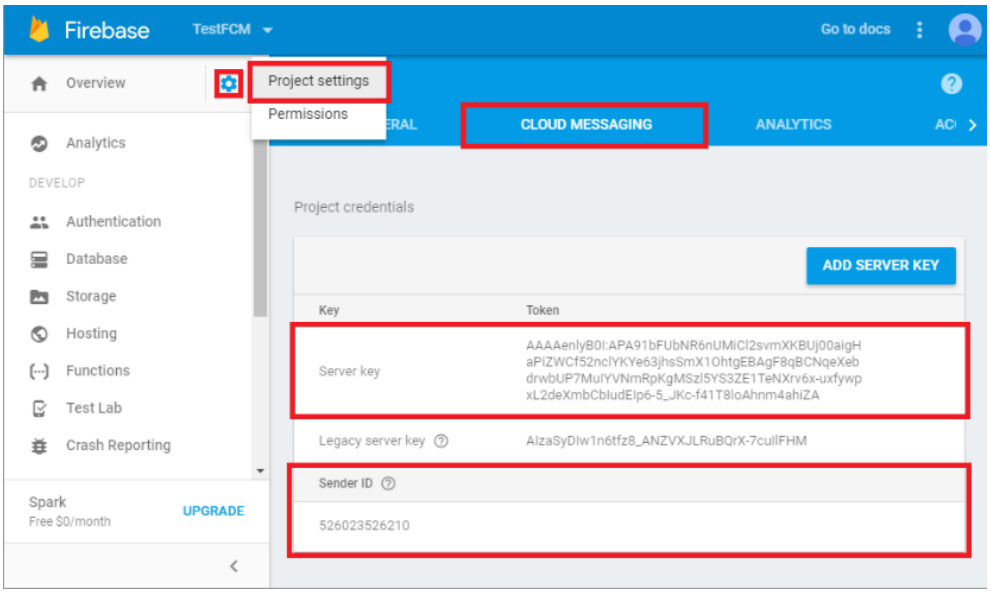


Figura 65. Documentación de la API REST del servicio de envío de notificaciones push OneSignal

Fuente [99]

Una vez concluido con el proceso de generación de las claves del servidor de Firebase, se procede a configurar los ajustes de la plataforma Android de la aplicación en la consola de OneSignal. Dentro de esta, se tiene que dirigir a la opción “Settings” y dentro de plataformas de aplicaciones nativas hacer clic en Google Android, ver Figura 66.

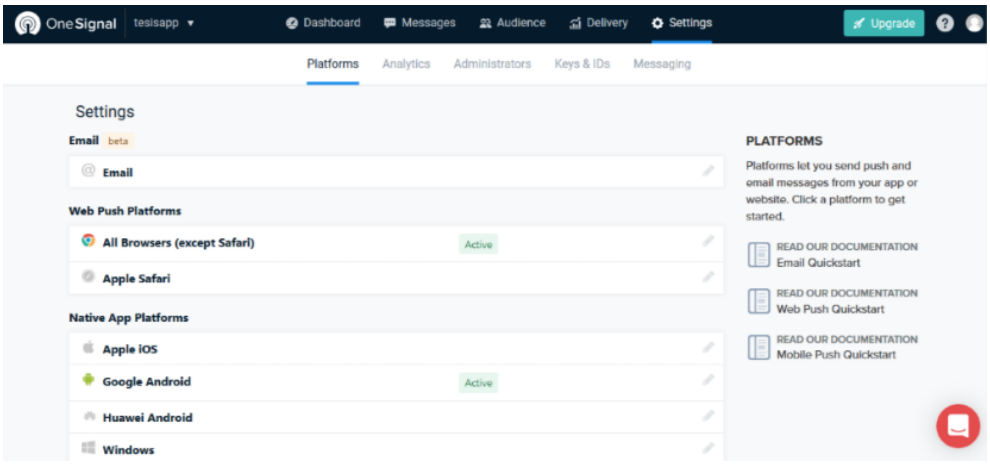


Figura 66. Panel de administración, botón Settings

Colocar la clave de servidor y el ID de remitente de Firebase en los campos solicitados y hacer clic en guardar, ver Figura 67. Una vez concluido con este proceso, se permitirá realizar notificaciones sin ningún inconveniente a los usuarios que hagan uso de la aplicación móvil.

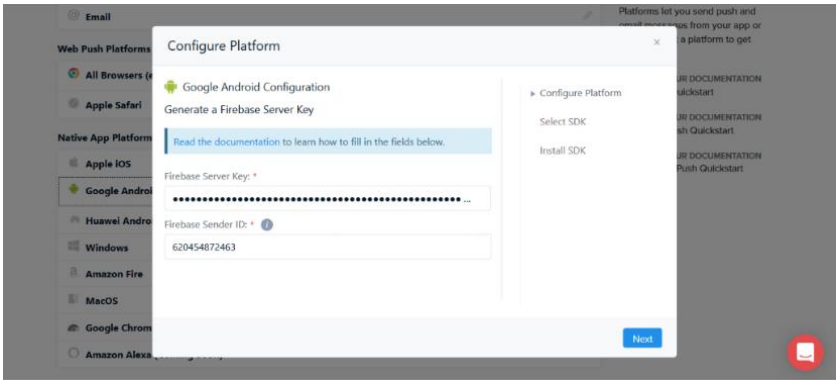


Figura 67. Clave de servidor y el ID de remitente de Firebase

Ahora bien, una vez que se ha finalizado la integración con la herramienta y se ha puesto en marcha el proceso de enviar los mensajes desde los nodos hasta el backend del sistema, este recibirá los datos, los interpretará y empezará a emitir los mensajes a cada una de las aplicaciones mediante el uso del método implementado mediante la librería axios, ver Figura 68. Es así como el administrador del sistema y el usuario que hace uso de la aplicación móvil podrán ser notificados y visualizar las notificaciones

```
function enviarNotificacion(parametrosEntrada){
    axios({
        method:'POST',
        url: 'https://onesignal.com/api/v1/notifications',
        data: {
        "app_id": "idSecretoOneSignal",
        "include_player_ids": "dispositivosANotificar",
        "contents": {"en": "Se ha detectado un actividad sospechosa alrededor de una propiedad."},
        "url":`urlDestino`,
        "web_buttons": [{"id": "web-button", "text": "Revisar", "icon": "../../assets/images/reminder.svg", "url": `urlDestino`}]},
        headers: headers,
    })
    .then(function (response) {
        console.log(`Se envio notificacion a ${nombreUsuario}`);
    }).catch(function (error) { console.log(error);});
}
```

Figura 68. Método utilizado para enviar notificaciones a la aplicación web y móvil

La interfaz que se muestra en la Figura 69, es el resultado final de la pantalla que permitirá al administrador del sistema visualizar de manera detallada el mensaje de alerta emitido por el nodo al detectar un intruso en la vivienda.

Figura 69. Información presentada en la aplicación web al recibir una alerta de detección de intrusos

La interfaz que se muestra en la Figura 70, es el resultado final de la pantalla que permitirá al administrador del sistema visualizar de manera detallada el mensaje de aviso de desconexión del nodo.

Figura 70. Información presentada en la aplicación web al recibir una alerta de desconexión de un nodo

La interfaz que se muestra en la Figura 71, es el resultado final de la pantalla que permitirá al usuario que hace uso de la aplicación móvil visualizar de manera detallada el mensaje de alerta emitido por el nodo al detectar un intruso en la vivienda.

Figura 71. Información presentada en la aplicación móvil al recibir una alerta de detección de intrusos

La interfaz que se muestran en la Figura 72, es el resultado final de la pantalla que permitirá al usuario que hace uso de la aplicación móvil visualizar de manera detallada el mensaje de aviso de desconexión del nodo.

Figura 72. Información presentada en la aplicación móvil al recibir una alerta de desconexión de un nodo

4.2.3. Operaciones CRUD

La Figura 73, muestra el botón *"USUARIOS"* que se encuentra en el menú principal de la aplicación web, el cual proporciona el acceso a la interfaz que permite gestionar a los usuarios dueños de los inmuebles.

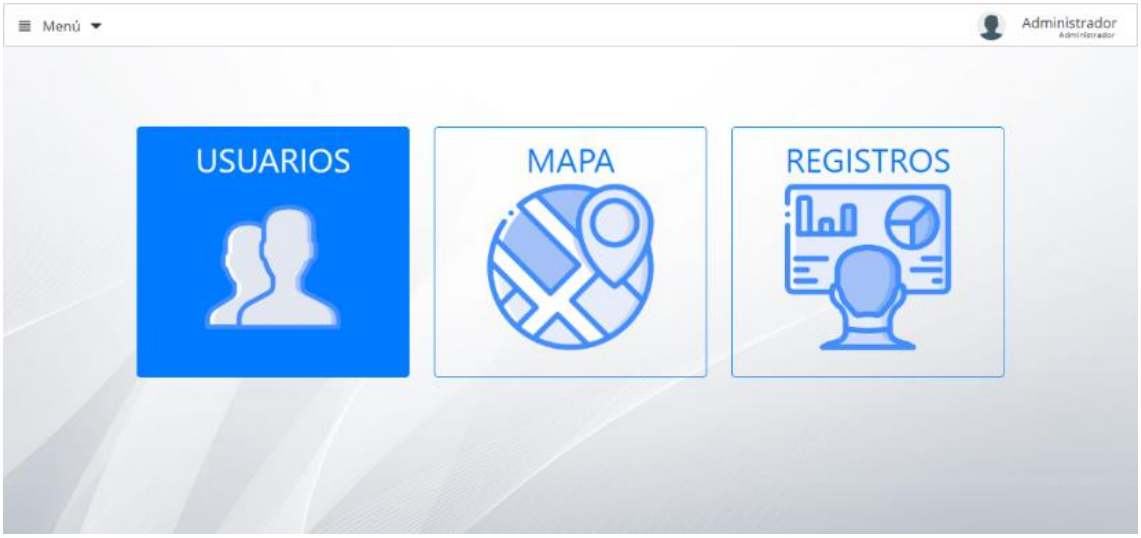


Figura 73. Interfaz que presenta la opción "USUARIOS" en el menú principal de la aplicación web

La interfaz que se presenta en la Figura 74, permitirá al administrador del sistema gestionar la información de los usuarios mediante operaciones CRUD (crear, buscar, actualizar y borrar), satisfaciendo de esta forma la historia de usuario HE03-01.

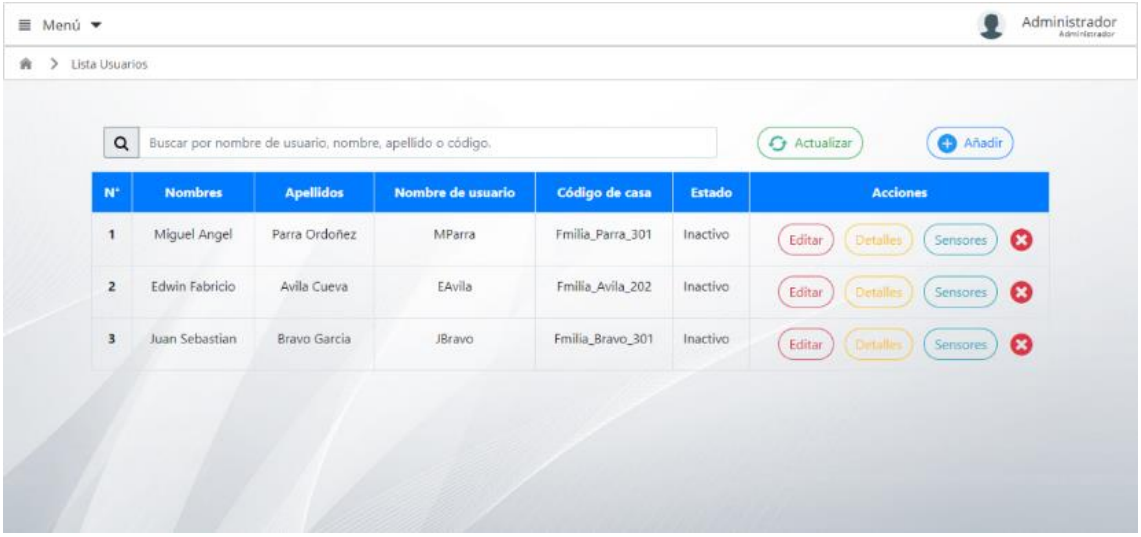


Figura 74. Interfaz que muestra los usuarios registrados en el sistema

Para realizar el proceso de creación de un usuario, se desarrollaron tres formularios (usuario, vivienda y nodo) y se implementó un servicio HTTP que permitirá la comunicación entre el frontend y el backend de la aplicación.

En la Tabla 41, se detallan las operaciones CRUD a usar en el sistema.

Método	Descripción
crear()	Este método se encarga de crear un nuevo registro, introduciendo como parámetros todos los campos necesarios para la creación de los registros.
actualizar()	Este método permite realizar la modificación de un registro introduciendo como parámetros el id del registro a actualizar y el campo que se desea modificar.
buscarPorId()	Este método permite obtener la información de un registro introduciendo como parámetro el id del registro a buscar.
eliminarPorId()	Este método se encarga de borrar el registro en la base de datos, introduciendo como parámetro el id del registro a eliminar.

Tabla 41. Operaciones CRUD

El formulario usuario, ver Figura 75, está constituido por los campos personales del propietario de la vivienda (nombres, apellidos, teléfono, correo electrónico, usuario y contraseña) y un botón que permite validar la información de los datos ingresados.

Figura 75. Interfaz que muestra el formulario del usuario

Antes de registrar al nuevo usuario en la base de datos, se procede a encriptar la contraseña mediante el uso de la librería bycript de JavaScript. Esta librería se instala por medio del comando *npm install bycript* y sirve para codificar las contraseñas mediante el uso de una función hash. En la Tabla 42, se presenta los métodos utilizados de la librería bycript en la aplicación web y móvil.

Nombre	Descripción
bcrypt.genSaltSync(10)	Es un método que recibe como parámetro un número que permite generar una cadena aleatoria de dígitos. Al aplicar este método junto con hashSync() la salida ya no es predecible. Es decir, la misma contraseña ya no producirá el mismo hash, puesto que se le agrega la cadena aleatoria al principio o al final, estableciendo así una contraseña de mayor longitud y más compleja.
bcrypt.hashSync()	Es un método que recibe como parámetro la contraseña proporcionada por el usuario y el genSaltSync para generar una contraseña hash.
bycript.compareSync()	Es un método que permite comparar la contraseña proporcionada por el usuario y la contraseña encriptada y almacenada en la base de datos. El método retornará un valor de true si las contraseñas coinciden, pero si no coinciden, retornará false y no permitirá la autenticación del usuario en las aplicaciones.

Tabla 42. Métodos de la librería bycript

Por su parte, al formulario vivienda, ver Figura 76, lo conforman los campos correspondientes a la ubicación y descripción de las viviendas que contienen los nodos detectores de intrusión. Estos campos son: calle principal, calle secundaria, nombre del propietario, número de manzana, número de bloque, tipo, color, número de vivienda y

número de piso. Además, posee un botón que permite validar la información ingresada en cada uno de estos.

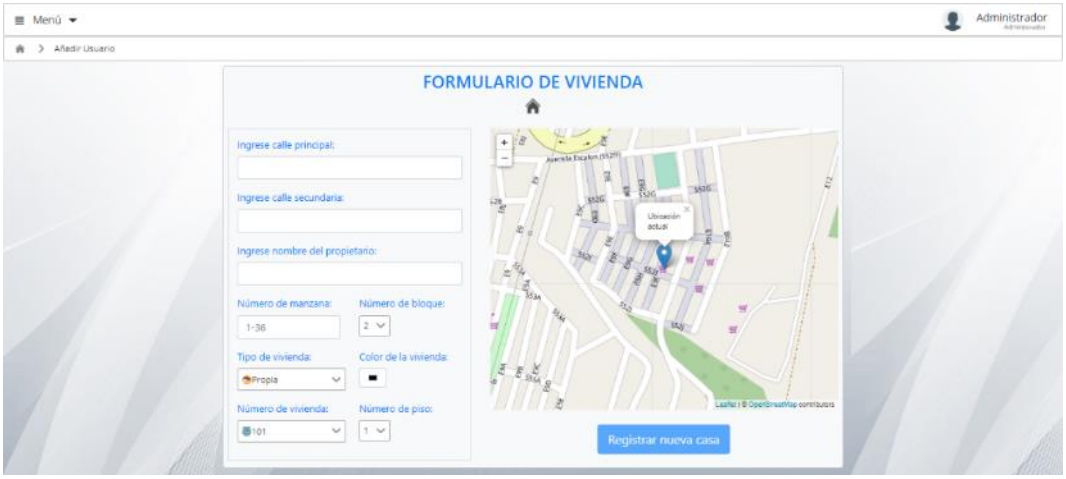


Figura 76. Interfaz que muestra el formulario de la vivienda

Por otro lado, el formulario nodo, ver Figura 77, está constituido por los campos que describen al sensor que está integrado en el nodo final (nombre del sensor, ubicación del sensor, tipo de sensor y estado) y un botón que permite validar la información ingresada por el administrador dentro de ellos.

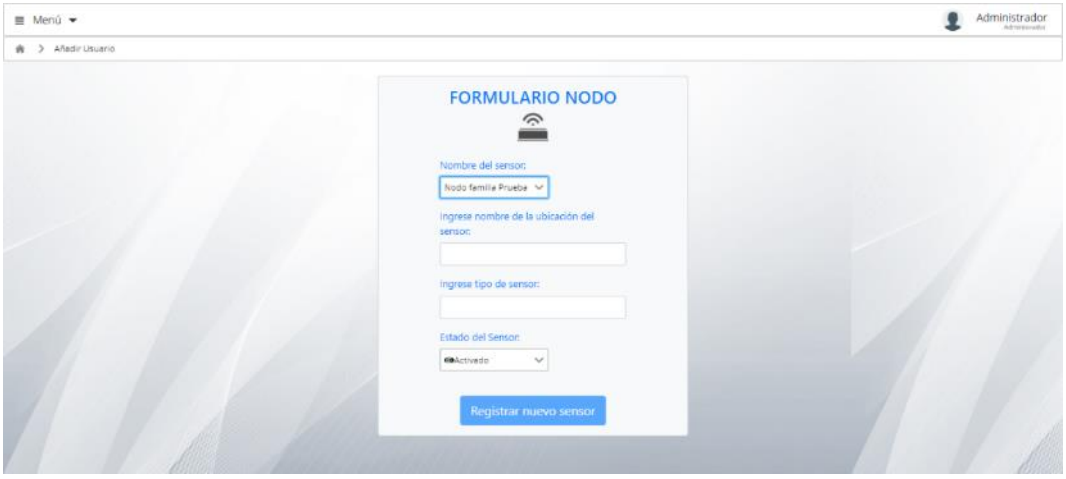


Figura 77. Interfaz que muestra el formulario del nodo final

Para obtener la lista de los sensores registrados desde el API de ChirpStack, se ha implementado el método c*onectarChirpStack*, el cual mediante una petición HTTP de tipo POST obtiene el token de acceso que permite utilizar todos los métodos provistos por esta API, ver Figura 78.

```
conectarChirpstack(){
    const headers = {
        "Content-Type" : "text/plain",
        "Accept" : "application/json"
    }

    const body = {
        "password": "contraseña",
        "username": "usuario"
    }

    const options = {
        headers : headers,
    }
    return this._httpClient.post('http://hostChirpStack:8080/api/internal/login',body,
    options);
}
```

Figura 78. Método conectar ChirpStack

Ahora bien, la interfaz que se muestra en la Figura 79, es el resultado final de la pantalla que permitirá actualizar los campos relacionados a la residencia. El único campo que no es posible modificar es el código de vivienda, debido a que es único y se conforma mediante el primer apellido del propietario y el número de vivienda.

Figura 79. Interfaz que permite modificar la información de un registro por medio de un formulario

La interfaz que se muestra en la Figura 80, presenta todos los usuarios registrados en el sistema. Al presionar el botón de color amarillo, representado con la palabra *"Detalles",* el método denominado *buscarporid* trabajará de manera inmediata para poder obtener la información del registro y mostrarlo en pantalla, ver Figura 81.

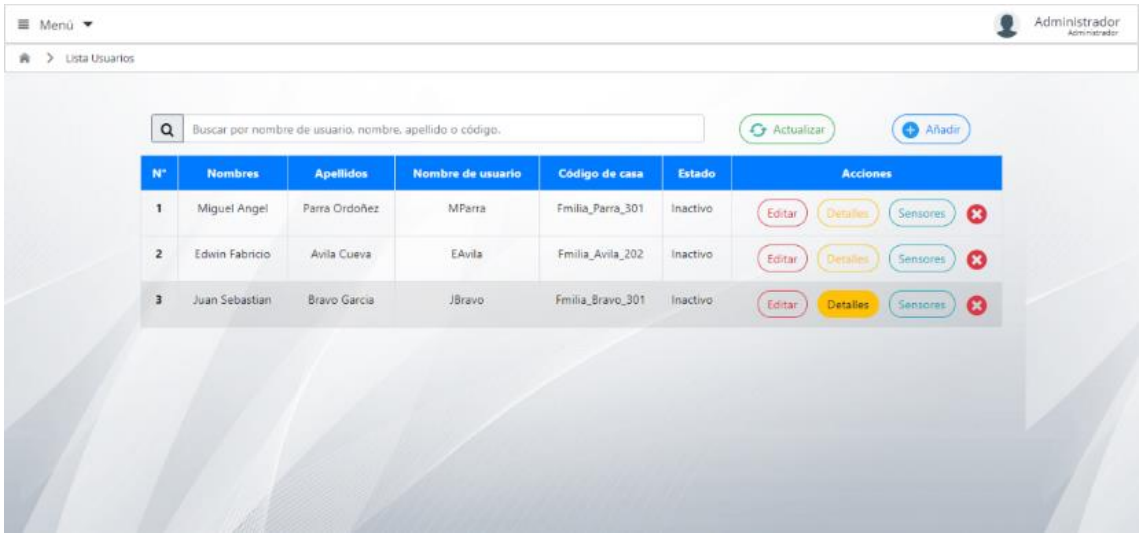


Figura 80. Interfaz que muestra la tabla con todos los registros de usuarios en el sistema

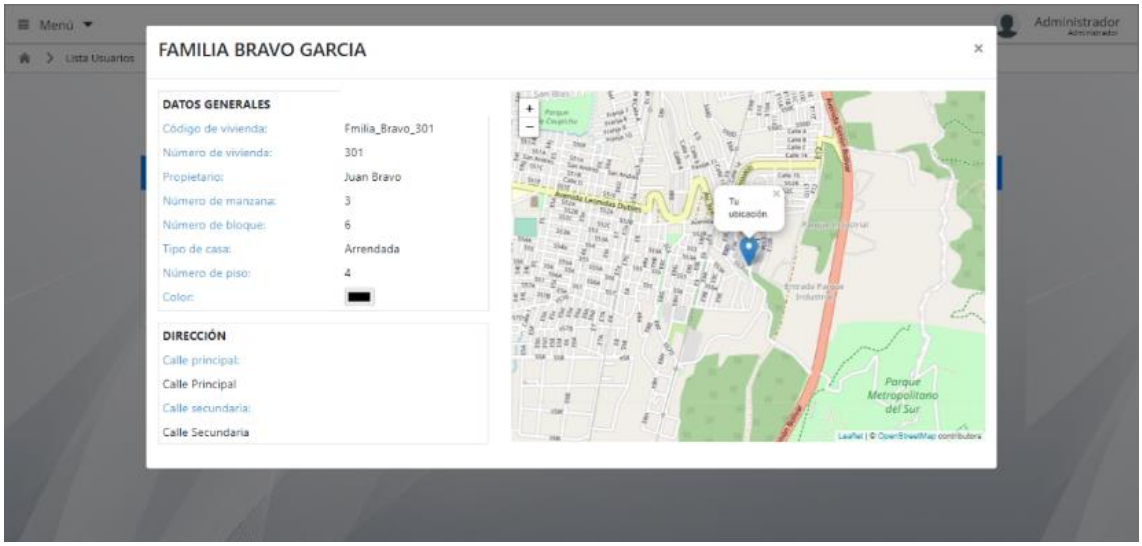


Figura 81. Interfaz que permite visualizar la información obtenida de un registro

Finalmente, al presionar el botón de color rojo representado por el icono de *"X"*, dentro de la interfaz que presenta todos los usuarios registrados en el sistema, el método denominado *eliminarporid* será ejecutado borrando el registro seleccionado, ver Figura 81.

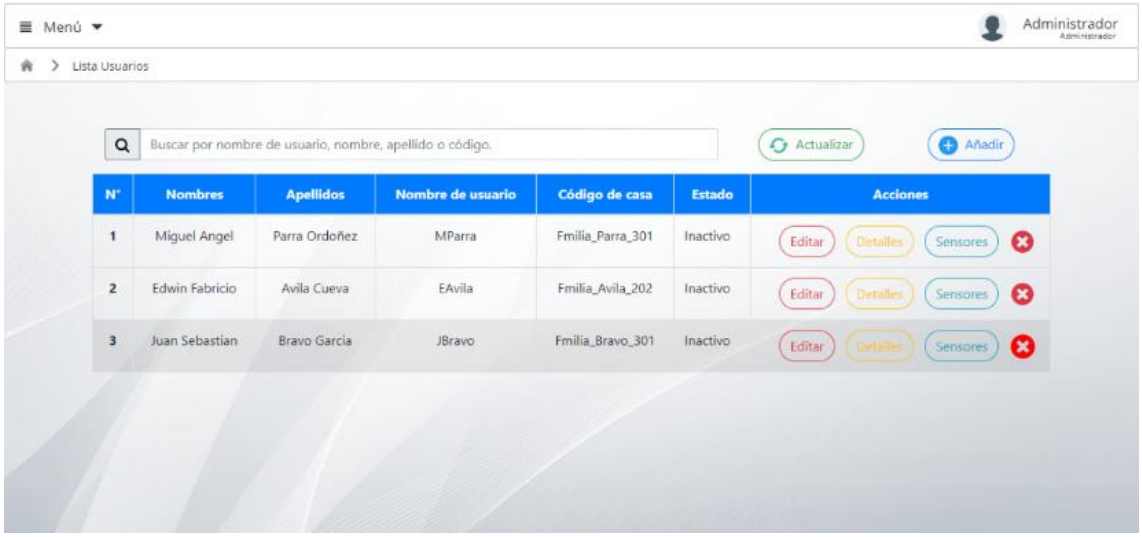


Figura 82. Botón que permite eliminar un usuario registrado en el sistema

4.2.4. Módulos de control de acceso

La interfaz que se presenta en la Figura 83, permitirá al administrador del sistema iniciar sesión y autenticarse a la aplicación web para hacer uso de sus funciones, de igual forma, la interfaz que se presenta en la Figura 84, permitirá al usuario propietario de una vivienda, iniciar sesión y autenticarse a la aplicación móvil; mismas que a su vez satisfacen la historia de usuario HE03-02.

Figura 83. Interfaz que muestra el formulario para el inicio de sesión en la aplicación web

Figura 84. Interfaz que muestra el formulario para el inicio de sesión en la aplicación móvil

Para efectuar el proceso de inicio de sesión y autenticación a las diferentes aplicaciones se hace uso de un servicio HTTP, que implementa el método “autenticarUsuario”, ver Figura 85, mismo que recibe como parámetros el nombre del usuario y la contraseña.

```
autenticarUsuario(nombreUsuario,contrasenaUsuario){
  this.nombreUsuario = nombreUsuario;
    const url = `${this.url}/autenticarUsuario?nombre=${nombreUsuario}&contrasena=${contrasenaUsuario}`
    this._httpClient.get(url)
```

Figura 85. Método “autenticarUsuario”

En la Tabla 43, se presentan los componentes utilizados para la autenticación y seguridad tanto de la aplicación web como móvil.

Nombre de los componentes	Descripción
BCRYPT	Es una librería que permite hacer uso de una función hash de codificación de contraseñas para evitar ataques de búsqueda de fuerza bruta.
JWT	Es un estándar abierto que permite manejar la autenticación en aplicaciones móviles o web. Define un método compacto que encapsula medios de seguridad para peticiones HTTP garantizando el envío de datos válidos y seguros entre aplicaciones o servicios. Está basado en el formato JSON para crear un token de seguridad [100].
GUARDS	Son interfaces true-false que indican si se tiene permitido o no el acceso a navegar dentro de una ruta especifica (vistas o páginas). Permite proteger a las rutas con el fin de ser accedidas siempre y cuando el usuario ésta logueado.

Tabla 43. Componentes utilizados para la autenticación y seguridad de las aplicaciones web y móvil

Una vez el usuario o administrador del sistema ha proporcionado las credenciales para el acceso, se efectúa una consulta a la base de datos del sistema para comprobar si el usuario existe, si el nombre de usuario proporcionado no está registrado en la base de datos emitirá un mensaje de error y desplegará en las aplicaciones un mensaje de acceso denegado, como se muestran en la Figura 86 y Figura 87. Por otro lado, si el nombre de usuario proporcionado se encuentra registrado en la base de datos, se procede a comparar la contraseña mediante la función “compareSync”, ver Figura 88, que ofrece la librería bycript, misma que devuelve “true” si la contraseña ingresada por el usuario es correcta y “false” si no lo es, impidiendo así el acceso a las funcionalidades de los aplicativos.

Figura 86. Interfaz que muestra el mensaje de acceso denegado a la aplicación web

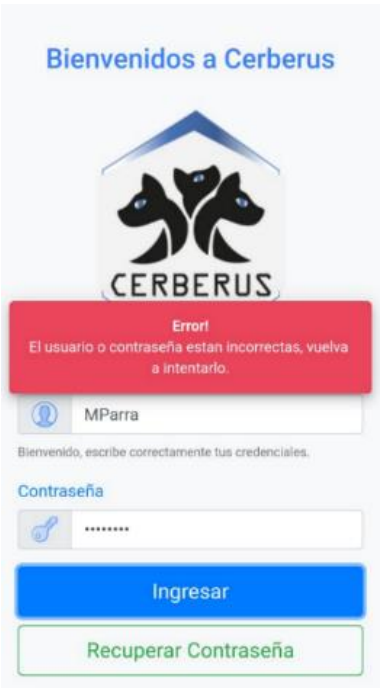


Figura 87. Interfaz que muestra el mensaje de acceso denegado a la aplicación móvil

```
checkContrasena(contrasenaFront,contrasenaUsuario):boolean{
  return bycript.compareSync(contrasenaFront,contrasenaUsuario,);
}
```

Figura 88. Método “compareSync”

Por otro lado, si no existe ningún inconveniente con los datos proporcionados por el usuario, la aplicación web y móvil, permitirán el acceso a las funcionalidades, como se muestran en la Figura 89 y Figura 90.

Figura 89. Interfaz que muestra el menú principal de la aplicación web al autenticarse correctamente

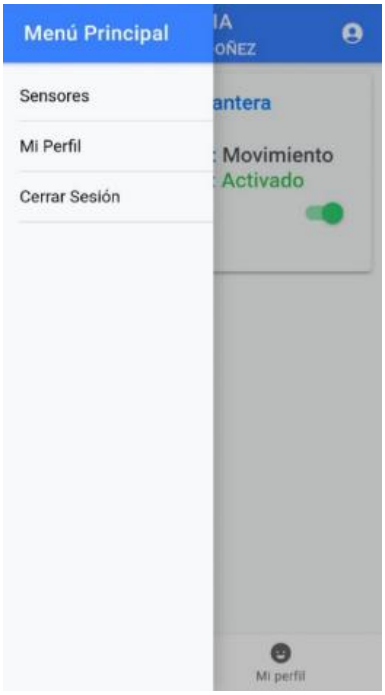


Figura 90. Interfaz que muestra el menú principal de la aplicación móvil al autenticarse correctamente

Después de realizar la comprobación del usuario en el sistema, se hace uso de la librería JWT (Json Web Token, por sus siglas en ingles) que se instala por medio de la librería *npm install jsonwebtoken*. Esta librería permite generar un token de acceso de tres partes

codificadas en Base64, de esta manera siempre que el usuario requiera realizar alguna operación con privilegios o acceder a las funcionalidades de las aplicaciones, debe enviar el token y el servidor verificará la firma y obtendrá los datos, ver Figura 91.

```
const token = localStorage.getItem('tokenSession');
try {
  const decode = jwt.verify(token, 'APEMAFCO@TLWSC');
  return true;
} catch(err) {
  alert('No Tienes Acceso')
    this._route.navigate(['/login']);
    return false;
}
```

Figura 91. Método “jwt.verify”

En la Tabla 44, se presentan los métodos utilizados por la librería jsonwebtoken en la aplicación web y móvil.

Nombre	Descripción
jwt.sign({nombreUsuario}, 'APEMAFCO@TLWSC', { expiresIn: '30m' });	Es un método de la librería jsonwebtoken que permite generar un token de acceso de tres partes codificadas en base64, introduciendo como parámetros el nombre de usuario, una clave secreta y el tiempo de expiración o validación del token.
jwt.verify(token, 'APEMAFCO@TLWSC');	Es un método que permite comprobar la validación y expiración del token generado, introduciendo como parámetros el token y la clave secreta.

Tabla 44. Métodos de la librería jsonwebtoken

Por otra parte, los guards trabajan de manera efectiva junto a la librería JWT para la seguridad en el lado del cliente, protegiendo el acceso no deseado a usuarios que no estén registrados en el sistema. En la Tabla 45, se presentan los guards utilizados por la librería jsonwebtoken en la aplicación web y móvil

Nombre	Descripción
CanActivate	Sirve para controlar si el usuario puede acceder a una página determinada.
CanActivateChild	Sirve para controlar si el usuario puede acceder a las páginas hijas de una determinada ruta.

Tabla 45. Tipos de interfaces guard del framework angular

4.2.5. Mapa en tiempo real

La Figura 92, muestra el botón *"MAPA"* que se encuentra en el menú principal de la aplicación web, el cual proporciona el acceso a la interfaz que permite monitorear el estado de los nodos de una vivienda.

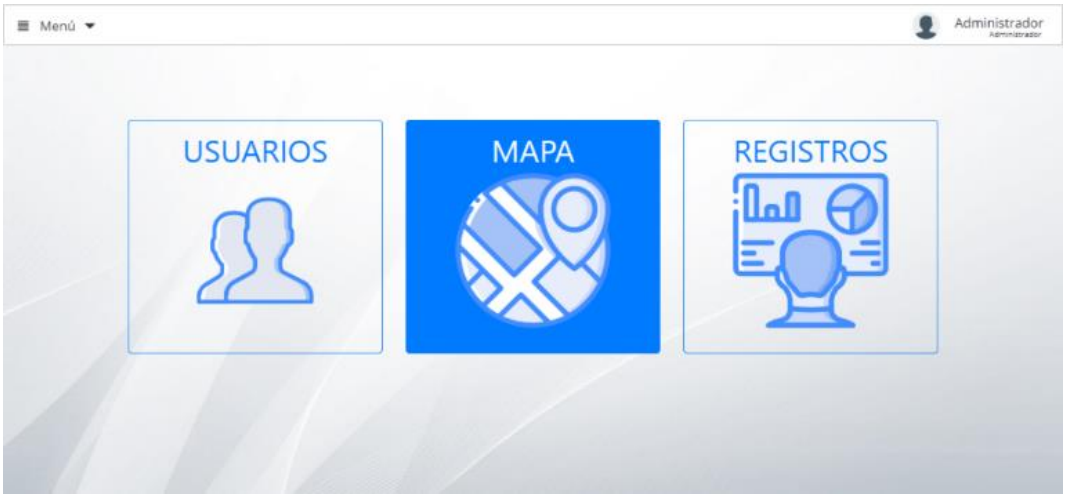


Figura 92. Interfaz que presenta el menú principal señalando la opción "MAPA"

La interfaz que se presenta en la Figura 93, permitirá al administrador del sistema visualizar el mapa en tiempo real, mostrando el estado de los nodos que están instalados en cada vivienda; misma que a su vez satisface la historia de usuario HE03-03.

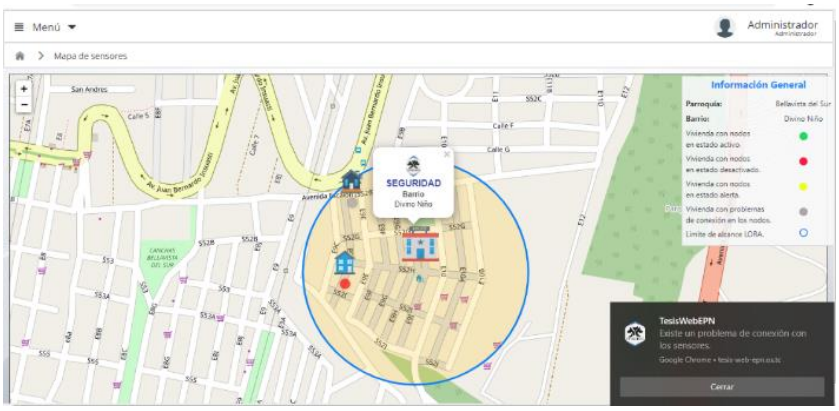


Figura 93. Interfaz que presenta el mapa y el estado de los nodos en tiempo real

Para la implementación del mapa se hizo uso de la librería de código abierto Leaflet, que permite desarrollar mapas interactivos compatibles con todas las plataformas. Esta librería se instaló usando el comando *npm install leaflet* por medio del administrador de paquetes de Node.js.

En la Tabla 46, se presenta una breve descripción de los métodos utilizados para la implementación del mapa.

Método	Descripción
setView()	Permite configurar las coordenadas geográficas y el nivel de zoom de la zona a visualizar.
marker()	Permite establecer el punto geográfico donde se ubican los distintos elementos del mapa.
addTo()	Permite añadir elementos dentro del mapa, como: iconos, marcadores, círculos, polígonos, etc.
bindPopup()	Se encarga de desplegar una ventana emergente con la información específica del marcador al hacer clic sobre el objeto.
openPopup()	Permite desplegar una ventana emergente fija al renderizar el mapa.
circle()	Permite dibujar un círculo dentro del mapa.
icon()	Permite establecer una imagen especifica al marcador.

Tabla 46. Métodos de la librería de código abierto Leaflet.

El circulo de color azul dentro del mapa representa el límite del alcance LoRa, el cual se ha dibujado por medio del método “circle”, aquí se especifica la latitud y longitud en la que será creado y el radio del mismo, que en este caso es de 218m, ver Figura 94.

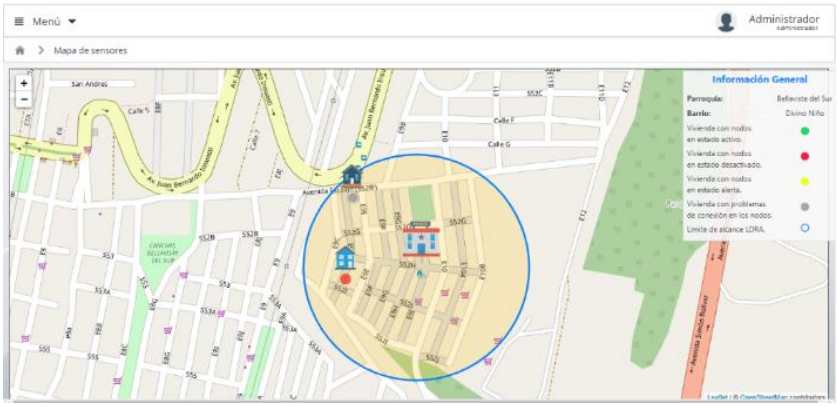


Figura 94. Límite de alcance LoRa representado por el círculo de color azul

Ahora bien, para poder presentar el mapa con la característica de tiempo real se utilizó la librería de JavaScript socket.io, la cual consiste en la comunicación bidireccional entre un servidor y un cliente. La librería se instaló mediante el comando *npm install socket.io*. El servidor (funcionado en el backend) emitirá un evento al cliente (frontend de la aplicación), el cual recibirá un mensaje y se encargará de efectuar los cambios necesarios en el mapa.

Se hizo uso de esta librería debido a la facilidad de enviar datos en cualquier momento dando como resultado un mapa en tiempo real. En la Tabla 47, se presenta la descripción de los métodos utilizados para la comunicación entre el servidor y el cliente de la aplicación.

Método	Descripción
server = require('http').createServer() **io = require('socket.io')(server)**	Permite configurar el servidor y establecer la conexión bidireccional para la comunicación.
io.on([connection].socket=>{})	Es un evento que permite escuchar y comprobar cuando un cliente se conecta al servidor.
io.on([cambioMapa].socket=>{})	Es un evento que permite escuchar y efectuar cambios en el mapa.
server.listen(port)	Este método permite configurar el puerto por el cual se podrá realizar la conexión entre cliente y servidor. Además de mantener la comunicación y escuchar los cambios que reciba del cliente o servidor.
io.emit([cambioMapa])	Es un método que permite emitir o comunicar un mensaje mediante sockets a los diferentes tipos de eventos.

Tabla 47. Métodos de la librería socket.io

Una vez que se ha configurado las herramientas y métodos que permiten la visualización del mapa en tiempo real, se procede a detallar la presentación de los nodos que están integrados en cada vivienda, ver Figura 95. Los nodos en el mapa están representados por un icono circular que dispone de diferentes colores dependiendo de su estado, siendo estos los siguientes:

- verde, al menos un nodo de la vivienda se encuentra en estado activo.
- rojo, todos los nodos de la vivienda están en estado desactivado.
- amarillo, al menos un nodo de la vivienda está en estado alerta.
- gris, al menos un nodo tiene problemas de conexión en la vivienda.

Para la implementación del icono dentro del mapa se hace uso del método *icon,* y para establecer el estado del nodo se hace una consulta a la base de datos obteniendo el estado actual del nodo.

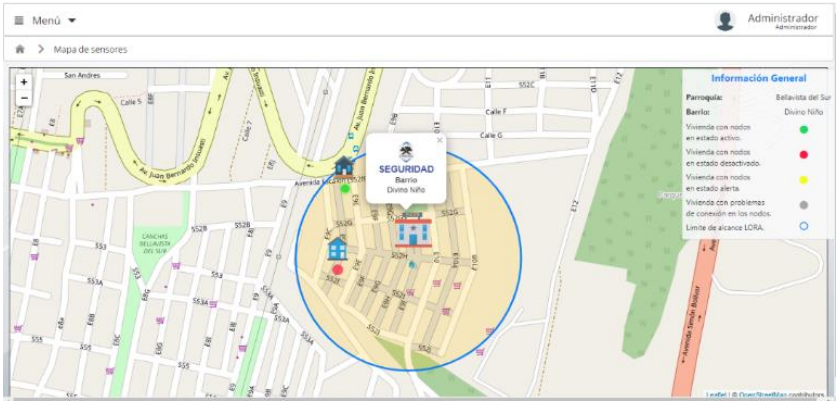


Figura 95. Interfaz que muestra el estado de los nodos de las viviendas

Para mostrar el icono circular de color verde en el mapa, ver Figura 96, el nodo debe estar en estado activo. Esto se lo realiza a través de la aplicación móvil, cambiando el estado del nodo de desactivado a activado. Al realizar dicha acción el nodo emitirá un mensaje al backend, el cual lo recibe por medio del método “suscribirMqtt”.

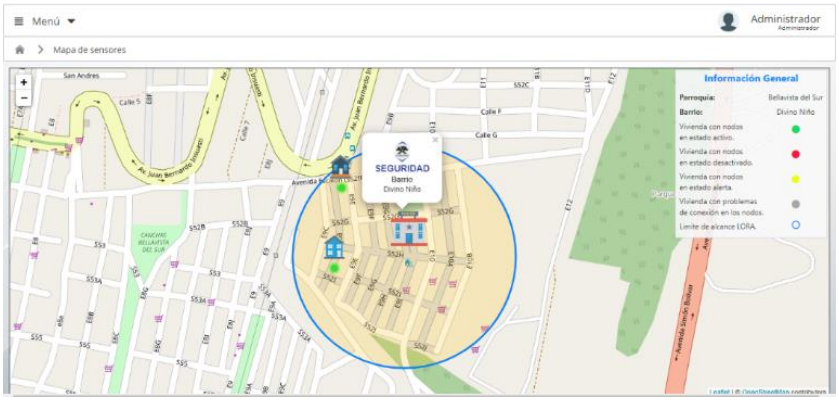


Figura 96. Interfaz que muestra al nodo de la vivienda en estado activo

Para mostrar el icono circular de color rojo en el mapa, ver Figura 97, el nodo debe estar en estado inactivo. Esto se lo realiza a través de la aplicación móvil, cambiando el estado del nodo de activado a desactivado. Al realizar dicha acción el nodo emitirá un mensaje al backend, el cual lo recibe por medio del método “suscribirMqtt”.

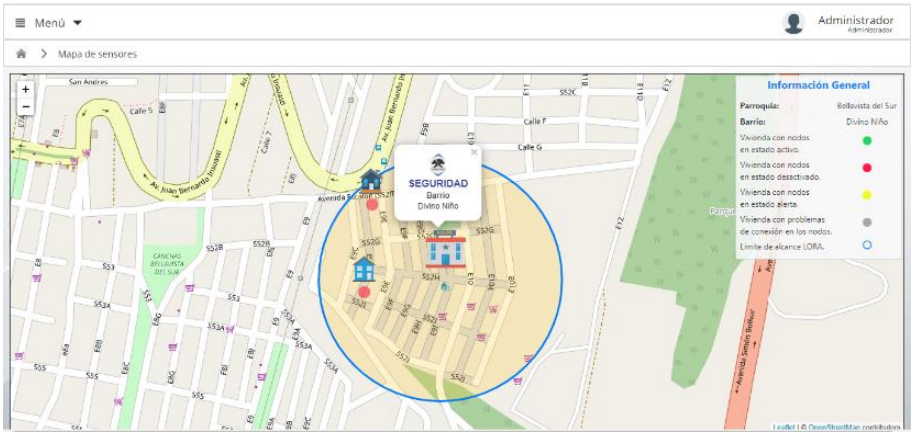


Figura 97. Interfaz que muestra al nodo de la vivienda en estado inactivo

Para mostrar el icono circular de color rojo en el mapa, ver Figura 98, el nodo debe estar en estado alerta. Esto lo realiza el nodo al notificar un acto de intrusión en el inmueble al backend, el cual lo recibe por medio del método “suscribirMqtt”.

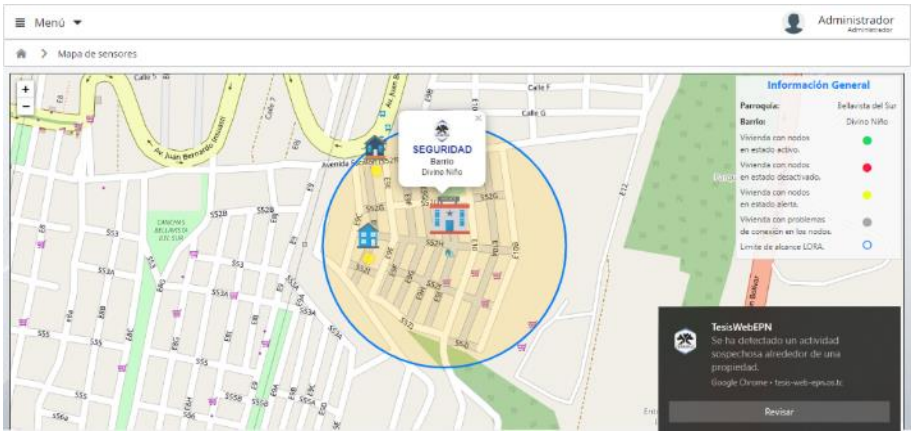


Figura 98. Interfaz que muestra al nodo de la vivienda en estado alerta

Debido a que el nodo emite mensajes de estado de conexión cada 3 minutos, el backend del sistema los recibe e interpreta, es decir, utiliza el método actualizar para modificar el campo “estadoConectado” a “true” de la tabla sensor. De esta manera el backend conoce que el nodo mantiene conexión con la red y no tiene problemas para emitir o recibir mensajes. Sin embargo, cuando el nodo deja de emitir los mensajes de estado de conexión, el backend modificará el campo a “false” y emitirá un mensaje de problema de conexión tanto al administrador del sistema como al usuario de la vivienda, además de cambiar a gris el icono circular en el mapa, ver Figura 99.

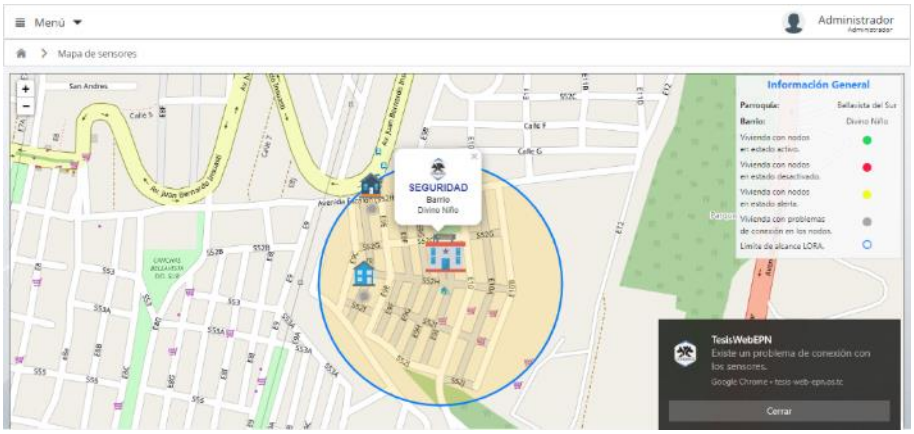


Figura 99. Interfaz que muestra al nodo de la vivienda en estado desconectado de la red

4.1.1. Gestión del estado de los nodos

La interfaz que se presenta en la Figura 100, es el resultado final de la pantalla que permitirá al usuario visualizar en forma de lista todos los nodos vinculados a su vivienda. Dentro de esta interfaz también el usuario podrá activar o desactivar los nodos, de modo que si el usuario se encuentra en su vivienda deberá colocar al nodo en estado desactivado, como se muestra en la Figura 101; de esta manera se satisface la historia de usuario HE03-04.

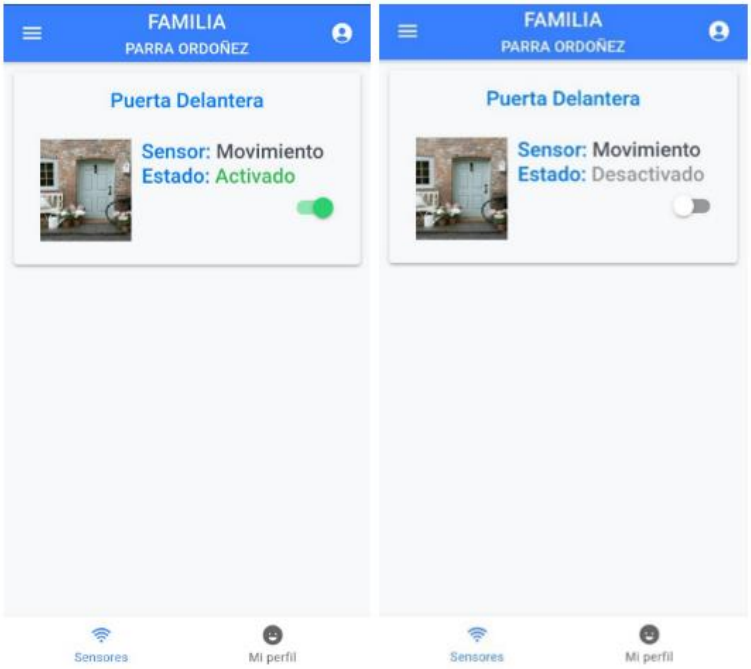


Figura 100. Interfaces de la aplicación móvil qué muestra el listado de los nodos vinculados a la vivienda en estado activado o desactivado

Figura 101. Interfaz de la aplicación móvil qué muestra el cambio de estado del nodo de activado a desactivado

Por otro lado, si el usuario se encuentra ausente, este deberá colocar al nodo en estado activado para que el nodo pueda emitir mensajes de alerta cuando exista una intrusión en la vivienda, como se muestra en la Figura 102.

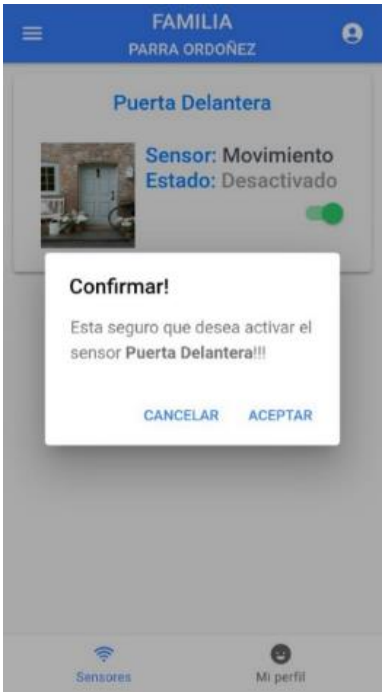


Figura 102. Interfaz de la aplicación móvil qué muestra el cambio de estado de desactivado ha activado del nodo vinculado en la vivienda

Para proceder a efectuar la activación o desactivación de los nodos de la vivienda, se hace uso de un servicio HTTP que implementa dos métodos: activar sensor y desactivar sensor, los cuales incluyen instrucciones que permiten hacer peticiones HTTP del tipo POST al servidor de aplicaciones de Chirpstack, enviando como parámetros el id del nodo y un mensaje con los números 1 (el nodo tiene permitido enviar mensajes a través de LoRa cuando el sensor detecte intrusos) o 2 (el nodo tiene prohibido enviar mensajes al detectar una intrusión).

El proceso para realizar el cambio de estado de un nodo se lo hace mediante el botón representado por un switch, como se muestra en la Figura 103. Al efectuarse el proceso se debe esperar a que el nodo emita un mensaje de enlace ascendente hacia el backend del sistema, donde recibirá como datos el número 2 o 3. Al recibir un 2, el método actualizar modificará el estado del nodo dentro de la base de datos pasando de desactivado a activado. Por otro lado, al recibir un 3, el método actualizar cambiará el estado del nodo de activado a desactivado. Mientras este proceso se está realizando, la aplicación móvil mostrará un mensaje al usuario informándole el cambio de estado del nodo, ver Figura 104.

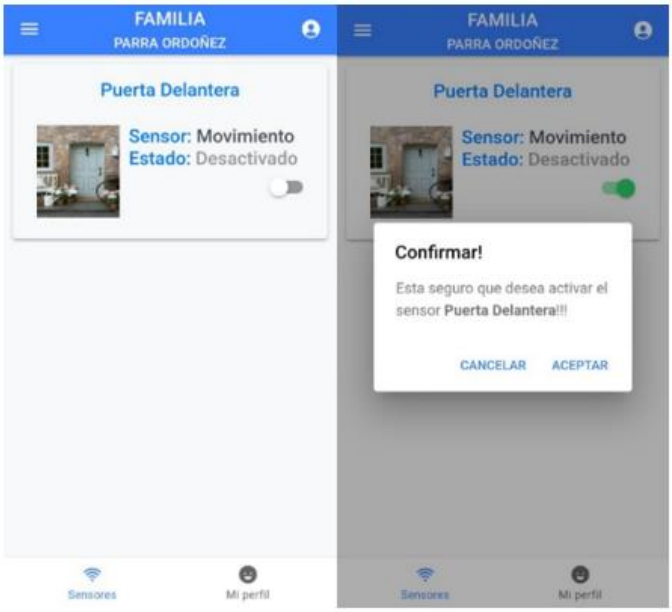


Figura 103. Interfaces que muestran el proceso de cambio de estado de un nodo vinculado a la vivienda

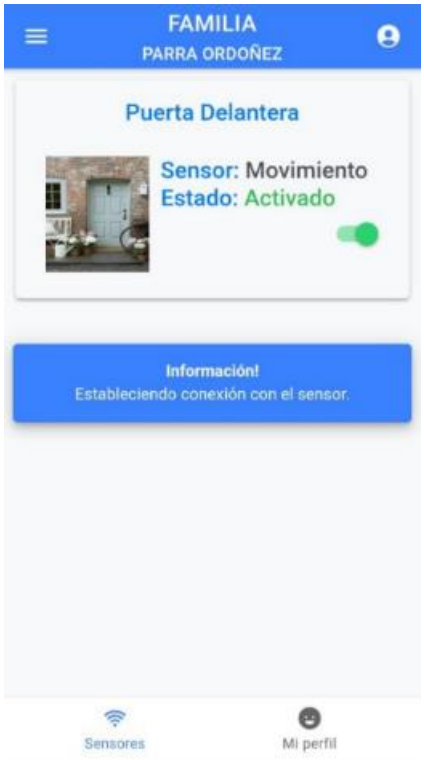


Figura 104. Interfaz que muestra el mensaje de confirmación de cambio de estado del nodo.

Finalmente, el cambio de estado también se refleja dentro del mapa de la aplicación web, Figura 105.

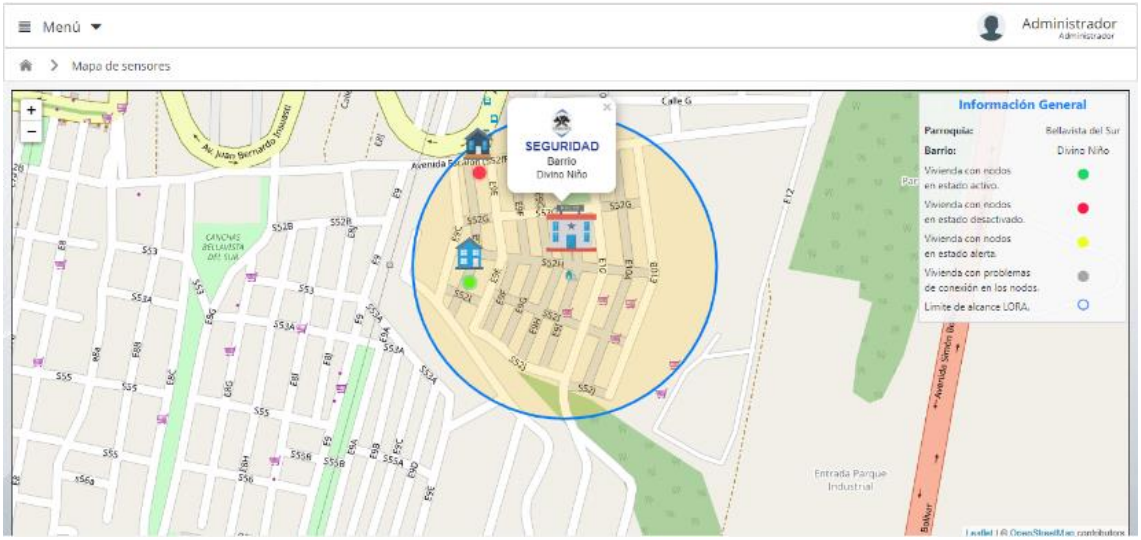


Figura 105. Interfaz de la aplicación web que muestra el cambio de estado del nodo

4.1.2. Registro de alertas

La Figura 106, muestra el botón *"REGISTROS"* que se encuentra en el menú principal de la aplicación web, el cual proporciona el acceso a la interfaz que permite visualizar el registro de alertas de detección de intrusos.

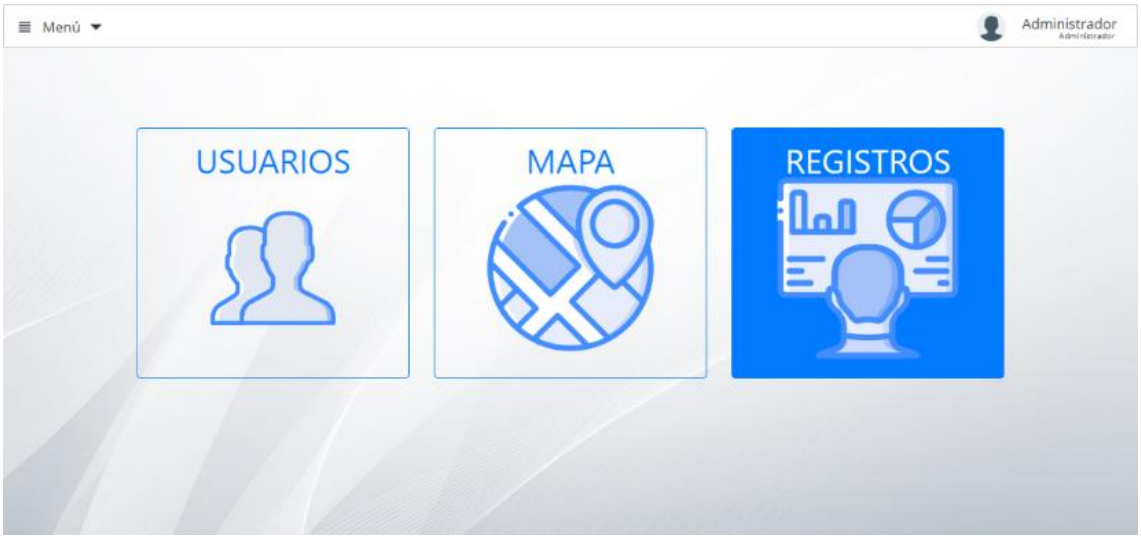


Figura 106. Interfaz que presenta el menú principal señalando la opción "REGISTROS"

La interfaz que se presenta en la Figura 107, es el resultado final de la pantalla que permitirá al administrador del sistema llevar un registro de todas las notificaciones que fueron emitidas durante la detección de intrusos. Esto se lo realizó con el propósito de tener constancia de las alertas que fueron recibidas; mismas que a su vez satisfacen la historia de usuario HE03-05.

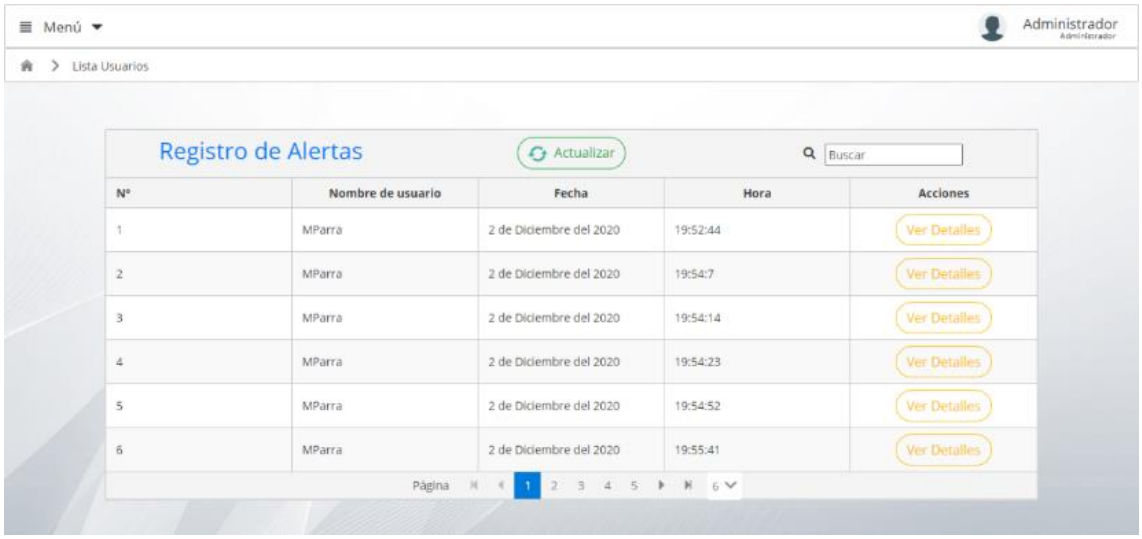


Figura 107. Interfaz que muestra todos los registros de alerta de detección de intrusos

Para mostrar el registro de alertas, se utilizó una tabla que permitió organizarlos. Además, se implementó un botón para mostrar detalladamente la información de cada notificación, ver Figura 108.

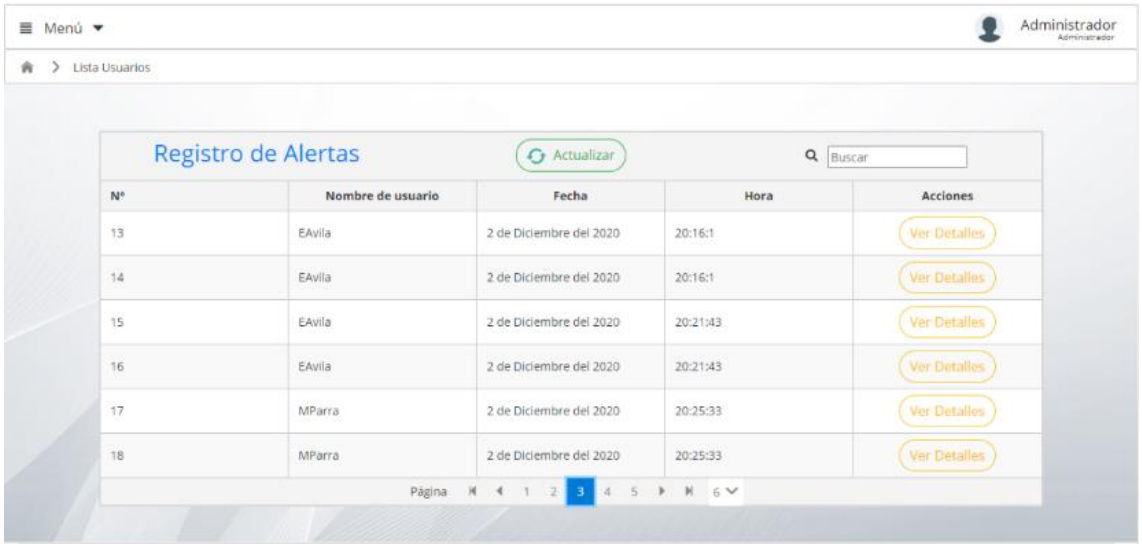


Figura 108. Interfaz que muestra la tabla donde se organizan los registros de alertas

La interfaz que se presenta en la Figura 109, es el resultado final de la pantalla que permitirá mostrar al administrador del sistema la información de cada alerta generada.

Figura 109. Interfaz que muestra la información relevante de la vivienda

CAPÍTULO V

5. PRUEBAS Y EVALUACIÓN DE RESULTADOS

5.1. ALCANCE

La prueba consistió en colocar de manera fija al gateway RAK7258 en la terraza de un bloque del conjunto Divino Niño, y transmitir desde los dos nodos implementados, configurados con DR0/SF10, un total de 20 mensajes de enlace ascendente; cada uno enviado desde distintos puntos geográficos con relación a la ubicación del gateway. La Figura 110, presenta el recorrido realizado, así como los puntos desde los que se transmitió cada mensaje LoRa.

Figura 110. Recorrido realizado para la transmisión de mensajes LoRa

Cabe mencionar que, para obtener el valor preciso de la distancia existente entre el punto de transmisión y la puerta de enlace, se emplea la opción “medir la distancia” de la herramienta Google Maps. Con estos datos, más los registros obtenidos del gateway y de los servidores de ChirpStack, se procedió a crear la Tabla 48, misma que contiene la información de los 16 mensajes que fueron receptados por la puerta de enlace LoRa.

Coordenadas RAK7258	-0.331214, -78.524810		**Nodo con sensor de movimiento**		**Nodo con sensor magnético**	
N° medida	**Coordenadas (latitud - longitud)**	**Distancia (m)**	**RSSI (dbm)**	**SNR (dB)**	**RSSI (dbm)**	**SNR (dB)**
1	-0.330833, -78.524996	**41**	-95	4,3	-97	4,5
2	-0.330887, -78.525693	**87**	-111	1,8	-112	-0,5
3	-0.330123, -78.525710	**148**	-122	-9,5	-123	-11,5
4	-0.330192, -78.526570	**218**	-131	-14.8	-132	-14,9

5	-0.330871, -78.526908	**228**	-103	5	-106	4
6	-0.330918, -78.527743	**323**	-109	2,5	-111	3,8
7	-0.330960, -78.528424	**392**	-112	0,3	-113	1,8
8	-0.330971, -78.528867	**448**	-114	-1,3	-115	-0,8
9	-0.331361, -78.529536	**517**	-118	-4,5	-117	-4,8
10	-0.331207, -78.530467	**610**	-122	-5,5	-120	-5,8
11	-0.331138, -78.531133	**693**	-125	-9	-124	-9,3
12	-0.331070, -78.531521	**740**	-126	-10,3	-126	-10,8
13	-0.331012, -78.531895	**783**	-128	-11	-127	-12
14	-0.330934, -78.532309	**829**	-129	-12,8	-129	-12,7
15	-0.330869, -78.532737	**872**	-130	-13,2	-131	-13,8
16	-0.330756, -78.533147	**922**	-132	-13,8	-132	-14,3

Tabla 48. Información referente a los mensajes receptados por el gateway en la prueba de distancia

Antes de analizar los resultados obtenidos es importante conocer la información que se encuentra registrada en esta tabla, por ende, se considera necesario aclarar dos campos relevantes. El primero corresponde al RSSI, el cual indica la potencia de la señal recibida expresada en dBm, tomando en cuenta que 0 dBm es equivalente a 1 mW. Generalmente el RSSI se expresa en valores negativos y mientras más cercano de 0, mejor es la señal. LoRa con SF12 puede “escuchar” señales con un RSSI mínimo de -137 dBm. El segundo campo concierne a la relación señal ruido (SNR), la cual indica la diferencia existente entre la potencia de la señal recibida y la potencia del ruido de fondo (toda señal interferente que daña a la señal transmitida), este indicador se expresa en dB. Mientras menor es el SNR, “peor” es la comunicación, normalmente un valor menor a 10 dB resulta en una mala comunicación debido a que el ruido de fondo es tan alto que lo hace casi indistinguible de la señal. Sin embargo, LoRa puede demodular señales por debajo de este nivel, hasta -20 dBm con un SF12.

Ahora bien, como se aprecia en los registros, existe un gran cambio en los valores de RSSI y SNR de ambos nodos a partir de la medición número 5 (sección color azul). Esto se debe a que las 4 primeras medidas fueron obtenidas sin que exista una línea de visión (SLV) entre los dispositivos LoRa, donde factores como: el número de viviendas en la zona y el material con el que se las ha construido (concreto armado), afectaron la intensidad de la señal recibida. Para apreciar de mejor manera lo mencionado se hace uso de la Figura 111, la cual muestra la relación existente entre el RSSI con respecto a la distancia recorrida.

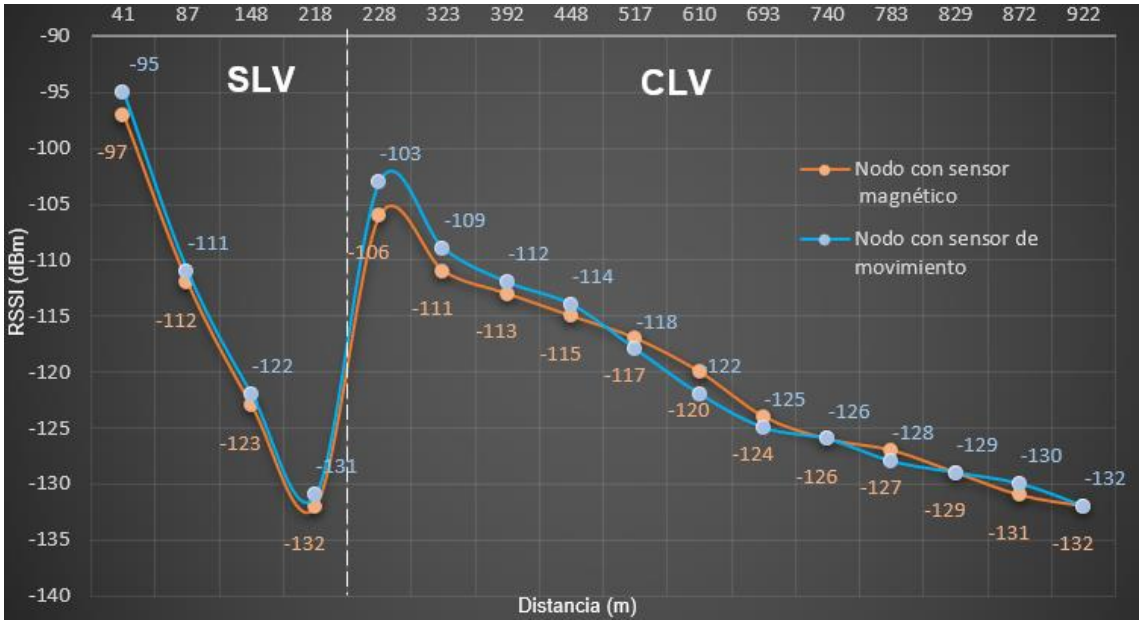


Figura 111. Relación entre distancia de transmisión y nivel de RSSI

Aquí se presentan 2 escenarios para cada nodo implementado. El primero corresponde a las pruebas realizadas sin línea de visión, en el que se visualiza como el RSSI disminuye rápidamente conforme la distancia aumenta, llegando a obtener un máximo de 218 m de comunicación, debido a que los obstáculos presentes en el camino bloquean, reflejan y absorben la señal emitida por los dispositivos LoRa. En el segundo escenario, con línea de visión (CLS), se observa que el RSSI de ambos nodos, a pesar de estar a una distancia mayor a la que se obtuvo en el último registro del anterior caso (registro 4), aumenta a un valor de -103 y -106 dBm, y a partir de estas medidas empieza a decrecer conforme se van emitiendo los mensajes desde los distintos puntos presentados. Sin embargo, en este escenario se logra obtener un máximo de 922 m de comunicación, ya que no existe obstáculos que impidan que la señal se atenúe rápidamente.

Lo mismo sucede con los valores obtenidos de la relación señal ruido, presentados en la Figura 112, los cuales reflejan que mientras mayor es la distancia menor es el valor de SNR. Es decir, este valor va decreciendo conforme los nodos se alejan del gateway ya que el ruido de fondo corrompe a las señales emitidas por cada nodo. Teóricamente un dispositivo que trabaje con un SF10 puede recuperar señales con un valor límite de SNR igual a -15, lo cual es corroborado en la prueba realizada, ya que para las distancias máximas obtenidas con y sin línea de visión este valor fue de -13.8 y -14.3 en el primer escenario, y de -14,8 y -14,9 en el segundo.

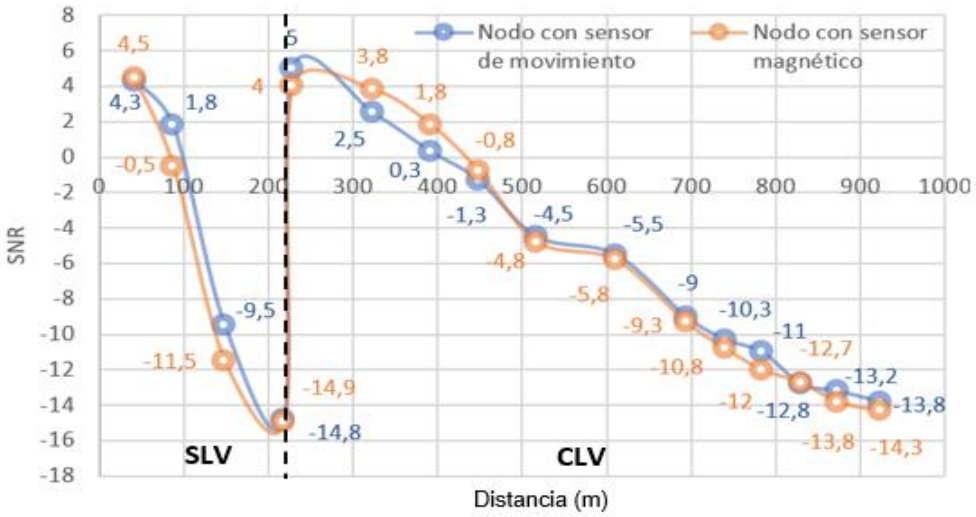


Figura 112. Relación entre distancia de transmisión y nivel de SNR

Como se pudo observar, el prototipo de red planteado tiene un radio de alcance máximo de transmisión LoRa de 218 m sin línea de visión y de 922 m con línea de visión, lo que la hace una solución viable, puesto que con una solo puerta de enlace se podría cubrir toda el área del sector residencial propuesto, esto ya que el mismo presenta un radio aproximado de 205 m. En la Figura 113, se presenta los radios de cobertura tanto de la zona residencial como de la comunicación inalámbrica LoRa sin línea de visión.

Figura 113. Cobertura de la red LoRaWAN

5.2. TIEMPO DE RESPUESTA

Antes de empezar a detallar la prueba realizada se debe aclarar que el mensaje de enlace ascendente transmitido por el nodo contiene 14 bytes, como se aprecia en la Figura 114,

de los cuales 1 byte corresponde al payload utilizado para notificar los distintos estados que transmite el nodo final, y los 13 bytes restantes están relacionados a los parámetros que añade el protocolo LoRaWAN para dar formato al mensaje.

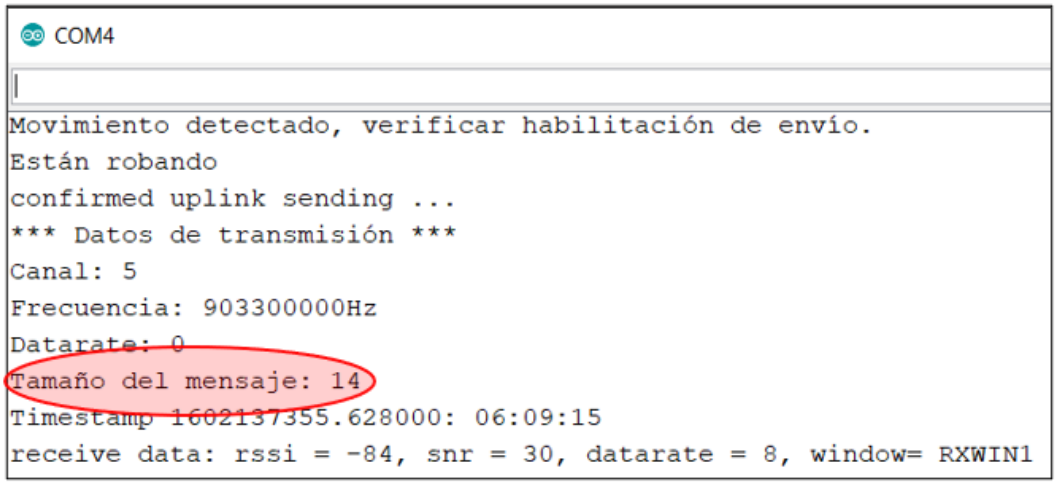


Figura 114. Tamaño del payload para los mensajes de enlace ascendente transmitidos

Dicho esto, es necesario especificar que la prueba consistió en medir 2 tiempos:

1. Tiempo que tarda en llegar el mensaje de confirmación, por parte del nodo, al activar o desactivar los mensajes de intrusión en el dispositivo final desde el aplicativo móvil.
2. Tiempo que le toma al prototipo de red enviar un mensaje de enlace ascendente, desde el dispositivo final hacia el backend del sistema.

Para ambos casos se utilizó la configuración de DR0/SF10 y se consideró 3 escenarios basados en los valores de RSSI de la señal, donde los registros obtenidos en cada uno permitieron determinar el tiempo promedio de las mediciones propuestas.

5.2.1. Activación o desactivación de mensajes de intrusión

Se ha escogido este proceso debido a que utiliza los dos tipos de enlace que pueden establecerse dentro de la red (uplink y downlink). El mensaje de enlace descendente es utilizado para conocer si debe activar o desactivar el envío de mensajes de detección de intrusos, y el mensaje de enlace ascendente permite confirmar que dicha actividad se realizó exitosamente, esto fue descrito con mayor detalle en la sección 4.1.1.

En esta prueba se utilizó el objeto “Date” de JavaScript junto con el método valueOf() para obtener las marcas de tiempo Unix en milisegundos. Se registraron dos marcas de tiempo dentro de la base de datos en cada muestra tomada, la primera se obtiene inmediatamente después de ejecutar la acción de activar o desactivar el dispositivo final desde el aplicativo

móvil, y la segunda es registrada al momento en que el mensaje de respuesta llega al backend. Como se mencionó anteriormente se propuso 3 escenarios, en los que se tomó en cuenta un valor alto, medio y bajo de RSSI, esto se lo hizo con el afán de conocer el tiempo que tardaría en completarse el proceso descrito cuando los componentes LoRa trabajan bajo esas condiciones. Es conveniente señalar también que la prueba fue aplicada únicamente al nodo con sensor magnético y que para lograr reducir el RSSI se tuvo que alejar el nodo del gateway a ciertas distancias.

Una vez obtenidos 60 registros (30 por tipo de enlace) para cada escenario propuesto se procedió a restar los tiempos de respuesta y de salida de los mensajes registrados en la base de datos, la Figura 115 muestra los distintos valores obtenidos al aplicar esta operación aritmética. Además, con los paquetes registrados en la puerta de enlace se logró identificar los valores de RSSI que emitió el nodo, mismos que hicieron posible definir los siguientes intervalos a cada escenario:

- **Escenario 1:** -73 a -85 dBm (paquetes enviados a una distancia aproximada de 20 a 30 m).
- **Escenario 2:** -104 a -113 dBm (paquetes enviados a una distancia aproximada de 60 a 95 m).
- **Escenario 3:** -123 a -129 dBm (paquetes enviados a una distancia aproximada de 145 a 185 m).

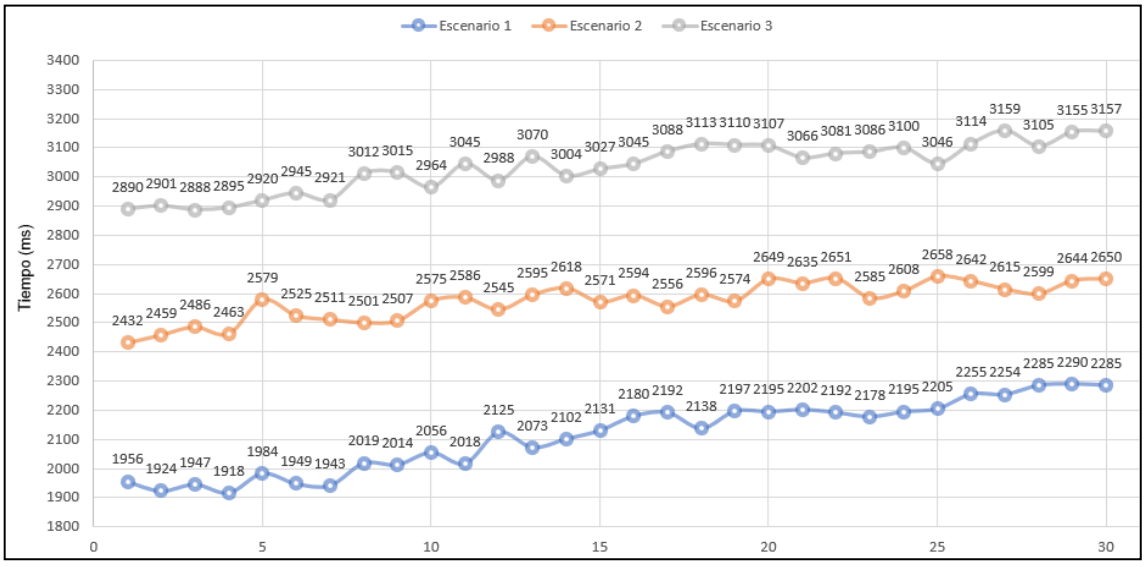


Figura 115. Tiempos de respuesta para el proceso de activación o desactivación de mensajes de detección de intrusos

En esta figura cada punto representa el tiempo aproximado que tardo la red en completar el proceso en las distintas muestras realizadas. Como se observa, existen variaciones en

los tiempos de cada escenario, sin embargo, las líneas que los representan tienden a ir creciendo ligeramente mientras el RSSI de la señal disminuye. Es decir, mientras menor es el RSSI más tiempo le toma al prototipo de red completar el proceso propuesto. De ahí que este se demora más en el escenario 3 debido a que presenta el menor intervalo de RSSI de todos. Se puede observar también que la diferencia existente entre los tiempos presentados de cada escenario se encuentra en el orden de los milisegundos. Así que para tener una idea clara de lo mencionado se hace uso de la Tabla 49, la cual presenta un resumen de todos los tiempos obtenidos para cada situación.

Escenario	1	2	3
Intervalo de RSSI (dBm)	-73 a -85	-104 a -113	-123 a -129
Tiempo mínimo (ms)	1918	2432	2888
Tiempo máximo (ms)	2290	2658	3159
Tiempo promedio (ms)	2113	2574	3034
Desviación estándar	120	63	85

Tabla 49. Resumen de los tiempos de respuesta para el proceso de activación o desactivación de mensajes de detección de intrusos

Con la ayuda de esta tabla se puede determinar que para el escenario 1, cuando los nodos se encuentran a una distancia próxima a la puerta de enlace, la solución propuesta tarda un tiempo estimado de 2113 ±120 ms en completar el proceso. Este tiempo tiende a incrementarse en 461 ms en el escenario 2 y 921 ms en el escenario 3, donde el proceso finaliza en 3034 ±85 ms; considerando que este último valor será el tiempo máximo aproximado en completarse el proceso, ya que aquí los nodos se encuentran casi al límite de la comunicación LoRa.

5.2.2. Envío de mensajes ascendentes

Dado que el prototipo de red involucra dos tipos de comunicaciones (LoRa e IP) para notificar los distintos estados del nodo, entre ellos la detección de un acto de intrusión en el inmueble, se considera conveniente conocer el tiempo aproximado que tarda en transmitir el mensaje de enlace ascendente en cada una de ellas. Por tal motivo, la prueba consiste en medir el tiempo de los 2 tramos que conforman este enlace. El primer tramo corresponde a la comunicación LoRa, la cual se lleva a cabo desde el nodo hacia la puerta de enlace, y el segundo, a la comunicación TCP/IP que se establece desde la puerta de enlace hacia el backend del sistema.

Para realizar esta prueba se hizo uso del protocolo NTP (Network Time Protocol), el cual permitió sincronizar los relojes de los elementos ubicados en los extremos de cada tramo mencionado. En cada elemento se utilizaron librerías que permitieron establecer un cliente

capaz de intercambiar paquetes UDP en el puerto 123 con un servidor NTP (en este caso el host 0.openwrt.pool.ntp.org) para obtener el tiempo en milisegundos y así mantenerse sincronizados. Una vez hecho esto se registraron 3 marcas de tiempo, las cuales se detallan a continuación:

- **Marca de tiempo 1:** se registra en el nodo al instante antes de transmitir el mensaje de enlace ascendente mediante la comunicación LoRa. Para esto se utilizaron 3 librerías, la librería WiFi.h que permite establecer la configuración necesaria para conectarse a una red mediante Wi-Fi, la librería WiFiUDP.h para la creación de una instancia que permitirá enviar y recibir mensajes UDP y la librería NTPClient.h que permite establecer un cliente capaz de manejar las solicitudes NTP en el puerto 123. Cabe mencionar que la marca de tiempo fue impresa en el Monitor Serial de Arduino.
- **Marca de tiempo 2**: se registra en la puerta de enlace, y se utiliza para establecer el punto final del tramo de la comunicación LoRa y el inicio de la comunicación por TCP/IP. En el Gateway no fue necesario incluir alguna librería puesto que predeterminadamente ya se encontraba corriendo un cliente NTP.
- **Marca de tiempo 3:** es registrada en el backend del sistema al momento en que llega el mensaje, y determina el punto final del tramo TCP/IP. Para esto se utilizó el módulo ntp-client que permitió establecer al cliente NTP.

La Figura 116, presenta los elementos que registran las marcas de tiempo descritas, así como los tramos que conforman el enlace ascendente de la prueba realizada.

Figura 116. Elementos que conforman los puntos extremos del tramo LoRa y TCP/IP

Esta prueba se basó en 3 escenarios determinados por los niveles de RSSI de la señal, tomando en cada uno 90 registros (30 por elemento). La resta en esta ocasión se efectuó entre el registro final e inicial de cada tramo propuesto. Cabe aclarar que la prueba se la realizó al nodo con sensor de movimiento y que los intervalos de RSSI, en base a los registros de los mensajes LoRa obtenidos del gateway, se establecieron de la siguiente manera:

- **Escenario 1:** -76 a -80 dBm (paquetes enviados a una distancia aproximada de 22 a 25 m).
- **Escenario 2:** -98 a -103 dBm (paquetes enviados a una distancia aproximada de 45 a 60 m).
- **Escenario 3:** -124 a -128 dBm (paquetes enviados a una distancia aproximada de 150 a 175 m).

La Figura 117, muestra la gráfica resultante de cada escenario una vez efectuada la resta de los registros tanto del tramo LoRa como del tramo TCP/IP.

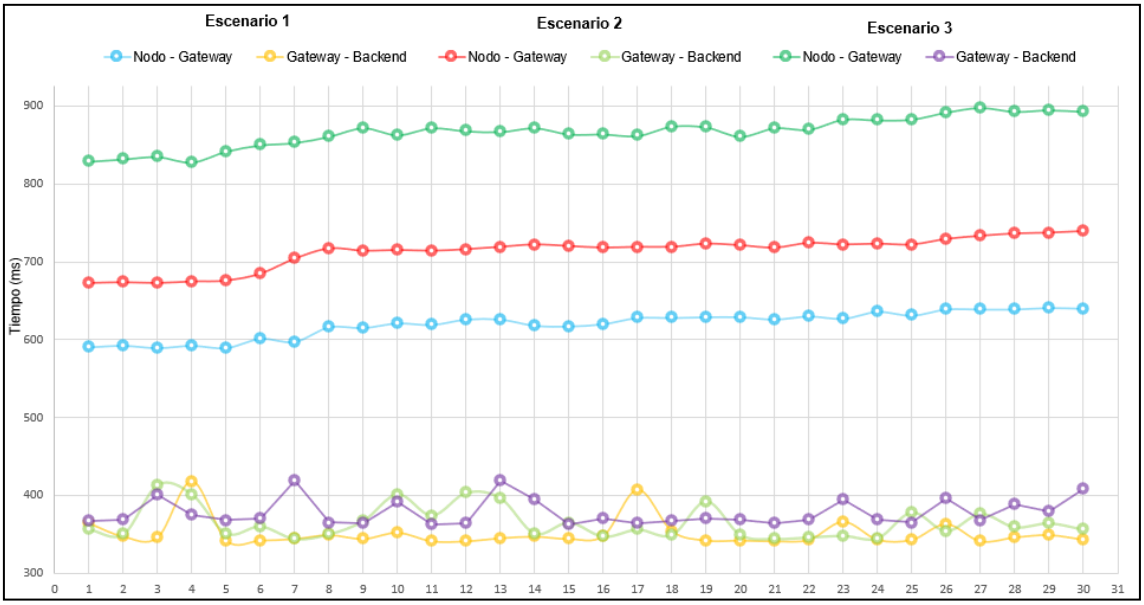


Figura 117. Tiempos que toma enviar un mensaje de enlace ascendente en los tramos LoRa y TCP/IP

Como se aprecia, existe una diferencia notable de tiempos entre los puntos que conforman las líneas del tramo LoRa (nodo-gateway) con respecto a los del tramo TCP/IP (gateway-backend), y esto se debe a que en el primer caso la disminución del RSSI afecta directamente a la comunicación inalámbrica LoRa, ya que al disminuir este valor el tiempo en llegar el mensaje tiende a incrementarse. Por su parte, los puntos que conforman las líneas correspondientes al tramo TCP/IP no sufren variaciones considerables, y en los tres escenarios las medidas se mantienen similares, esto se debe a que el segundo tramo utiliza una conexión vía Ethernet, por ende, el rendimiento es más estable porque no se ve tan afectado por factores externos como sucede en una comunicación inalámbrica.

Para entender mejor lo mencionado se utiliza la Tabla 50, la cual presenta los tiempos promedios para cada trama en los distintos escenarios establecidos.

Escenario	1		2		3	
Conexión	**LoRa**	**TCP/IP**	**LoRa**	**TCP/IP**	**LoRa**	**TCP/IP**
Valor mínimo (ms)	589	341	673	344	828	362
Valor máximo (ms)	641	418	739	413	898	419
Promedio (ms)	620	351	713	365	867	378
Desviación estándar	17	18	20	21	20	17

Tabla 50. Resumen de los tiempos que toma enviar un mensaje de enlace ascendente en los tramos LoRa y TCP/IP

En esta tabla se puede observar una diferencia en el tiempo de los distintos escenarios para el tramo LoRa. El promedio de tiempo en el primer escenario es de 620 ± 17 ms, que se incrementa en 93 y 247 milisegundos para el segundo y tercero; reiterando que esto ocurre solo cuando el RSSI disminuye. Algo importante de resaltar es la diferencia de tiempo que existe entre el escenario 2 y 3, la cual es de 154 ms, considerando que entre ellos existe una disminución de RSSI entre 21 y 25 dBm. Por lo que se puede determinar que uno de los tiempos máximos que le tomaría al prototipo implementado en enviar un mensaje de enlace ascendente, dentro del tramo LoRa, se encontraría cercano a 898 ms, que es el valor máximo obtenido en el escenario 3, puesto que a este le faltarían 4 dBm para llegar al límite de la comunicación LoRa.

Considerando los valores máximos obtenidos en ambos tramos, se puede evidenciar que la solución propuesta utiliza alrededor de 1,3 s en enviar una notificación. Es decir, la notificación es inmediata, por lo que el personal encargado de la seguridad podrá llegar a tiempo a verificar lo que sucede en el inmueble, puesto que un robo dentro del domicilio tarda aproximadamente entre 3 a 12 minutos según [101].

5.3. CONSUMO DE CORRIENTE

Esta prueba tuvo como fin conocer el consumo de corriente que tienen los nodos LoRa de clase C dentro del prototipo de red desarrollado, y mostrar la diferencia de consumo que existe con una configuración de clase A. En base a los resultados también se logró estimar, como ejemplo, el tiempo de vida útil que tendría una batería de 4200 mAh bajo estas condiciones de funcionamiento.

Para la medición de corriente se empleó un multímetro conectado en serie entre la fuente de alimentación para protoboard y el pin de tensión de 3.3v de la placa, tal como se muestra en la Figura 118. La fuente utilizada fue el módulo YwRobot 545043 el cual suministra 3.3V o 5V (seleccionable mediante un jumper) como tensión de salida.

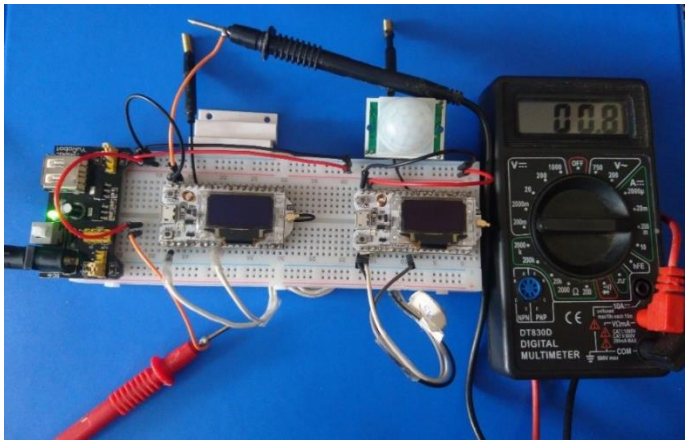


Figura 118. Colocación en serie del multímetro para la medición de corriente

Es importante aclarar que ambos nodos funcionan en base a dos estados, el estado “transmitiendo” que ocurre al momento que el nodo envía un mensaje mediante la comunicación LoRa, y el estado “en espera” que ocurre cuando el nodo se encuentra “atento” en recibir algún mensaje en la ventana RX2, o en detectar alguna situación de intrusión en la vivienda. Para comprobar el consumo energético de los nodos se tomó 20 mediciones a cada uno (10 por estado). La Figura 119, muestra los datos obtenidos al haber llevado a cabo este proceso.

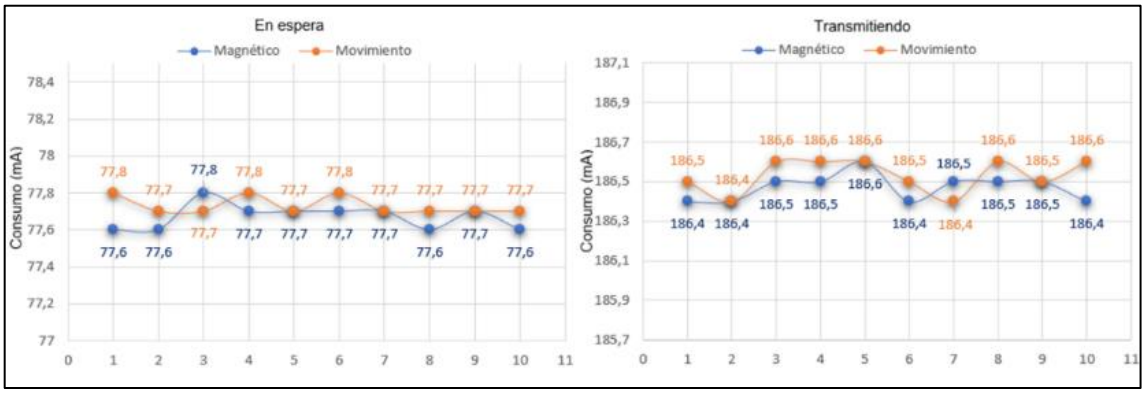


Figura 119. Consumos de corriente de los nodos de clase C para los estados: en espera y transmitiendo

Aunque se observa como las medidas de consumo para ambos nodos se mantienen similares, existe cierta tendencia por parte del nodo de movimiento a ser superior al magnético. Esto se debe a que el sensor de movimiento necesita consumir energía extra para funcionar (~50µA), mientras que el magnético no, ya que funciona únicamente como un interruptor, es decir, cuando el imán está cerca de la base, el circuito se encuentra cerrado y cuando se aleja, el circuito se encuentra abierto. Ahora bien, en lo que sí existe una gran diferencia es entre los dos estados que presentan los nodos, se observa como los valores se incrementan notablemente cuando estos se encuentran transmitiendo el mensaje. Esto es normal ya que el dispositivo final necesitará consumir energía adicional

para poder enviar dicho mensaje con cierto nivel de potencia. La Tabla 51, presenta un resumen de los consumos promedios para ambos casos.

Nodo	Magnético	Movimiento
Medición	**Amperaje (mA)**	**Amperaje (mA)**
En espera	77,67	77,73
Transmitiendo	186,47	186,53

Tabla 51. Resumen del consumo de corriente de los nodos de clase C para los estados: en espera y transmitiendo

En base a la tabla presentada, se puede evidenciar que cuando los nodos se encuentran en estado de transmisión, consumen ~109 mA adicionales a los que normalmente necesitan para poder funcionar como dispositivos de clase C. Sin embargo, se debe tomar en cuenta que este estado permanece activo únicamente milésimas de segundo, por lo que al terminar de enviar el mensaje los nodos retornarán al estado “en espera” consumiendo la mayor parte de tiempo ~77,7 mA. Se debe considerar que mientras menor sea este valor mayor será el periodo de duración de la batería.

Para entender el impacto que tiene en consumo energético el usar un dispositivo final de clase C, se cambió la configuración al nodo de movimiento como clase A; tomando en esta configuración, al igual que en el anterior caso, 10 mediciones por estado. La Figura 120 presenta la gráfica resultante de los datos obtenidos en ambas clases.

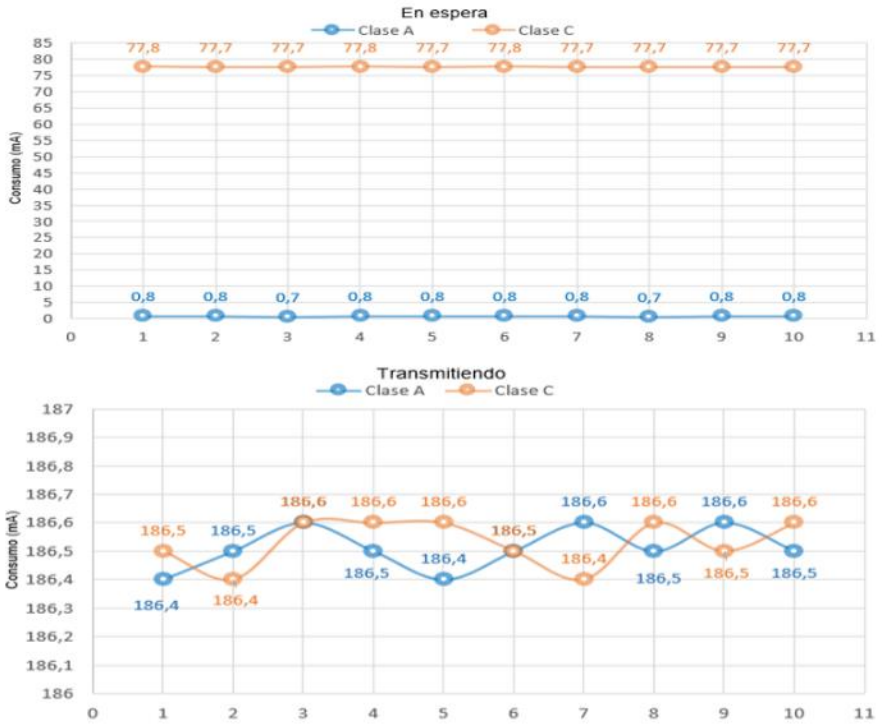


Figura 120. Consumos de corriente de los nodos de clase A y C para los estados: en espera y transmitiendo.

Como se aprecia en esta figura, existe una notable diferencia de consumo dentro del estado “en reposo” de ambas clases, mismo que llega a ser inferior a 1mA cuando el nodo utiliza la configuración de clase A. Este nivel de consumo se debe a que dentro de la clase A el nodo puede usar el modo de “sueño profundo”, el cual pone a “dormir” al procesador principal y a la mayor parte de periféricos con el fin de reducir el consumo al mínimo; sin embargo, deja al coprocesador ULP (Ultra Low Power) y a la memoria RTC (Real Time Clock) encendidos de forma que utilicen un temporizador interno para despertar nuevamente al procesador. Este modo únicamente puede ser implementado en un nodo de clase A ya que no necesita ejercer el rol de actuador dentro de la red; requisito que es indispensable dentro del prototipo implementado. Por otra parte, en la gráfica del estado “transmitiendo” no existe variación significativa, y esto se debe a que en ambos casos se necesitará de la misma potencia de transmisión para poder enviar algún mensaje mediante LoRa.

Las mediciones anteriores permitieron crear la Tabla 52, la cual presenta el consumo promedio para ambas clases y que permite llevar a cabo una estimación aproximada del tiempo de vida útil que tendría una batería de 4200 mAh en estas condiciones.

Clase	A	C
Medición	**Amperaje (mA)**	**Amperaje (mA)**
En espera	0,8	77,7
Transmitiendo	186,5	186,5

Tabla 52. Resumen del consumo de corriente de los nodos de clase A y C para los estados: en espera y transmitiendo

En base a los valores de esta tabla, se puede determinar que una batería con capacidad de 4200 mAh, considerando únicamente el estado “en espera”, duraría aproximadamente 5252 horas (218 días) trabajando con una configuración de clase A y 54 horas (3 días) con la configuración de clase C. Con estos resultados se evidencia como un nodo de clase C no es óptimo si se desea trabajar con baterías. Sin embargo, para la solución propuesta esto no genera inconveniente, ya que se considera tenerlo conectado a la corriente eléctrica debido a la accesibilidad de tomacorrientes que tienen las viviendas; evitando, de esta manera, que el usuario este pendiente en cambiar o recargar baterías cada determinado tiempo.

CAPÍTULO VI

6. CONCLUSIONES, RECOMENDACIONES Y POSIBLE IMPLEMENTACIÓN

6.1. CONCLUSIONES

- El trabajo presentado, basándose en los principios de LPWAN y de ciudades inteligentes ha logrado demostrar que la tecnología LoRa junto con LoRaWAN son útiles para implementar soluciones relacionadas a la seguridad de un inmueble, pues permitieron notificar inmediatamente diferentes actos de intrusión tanto al personal encargado de la seguridad del sitio como a su propietario.
- Se ha evidenciado que el proyecto de código abierto ChirpStack facilita la creación de redes LoRaWAN privadas, proporcionando componentes adicionales que permiten establecer una sencilla integración con la aplicación cliente.
- Las pruebas realizadas permitieron conocer que el prototipo desarrollado tiene un radio de alcance máximo de transmisión LoRa de 218 m sin línea de visión y que utiliza alrededor de 1,3 s en enviar una notificación desde el nodo hacia el backend del sistema. También se evidenció que el consumo de corriente de los nodos detectores de intrusión cuando se encuentran en reposo es de 77,7 mA.
- Al finalizar la implementación del prototipo propuesto y en base a los resultados obtenidos al efectuar las pruebas sobre el mismo, se ha comprobado que la solución propuesta puede ser implementada fácilmente en zonas residenciales.
- Este proyecto ha demostrado ser de gran ayuda para el personal responsable de la seguridad de la zona, puesto que le permite monitorear en tiempo real los inmuebles del sector, notificándoles inmediatamente si se detecta algún acto de intrusión en las mismas. Incluso, brinda tranquilidad al propietario del inmueble ya que le proporciona seguridad a su hogar cuando este no se encuentra presente. Además, en el caso de concretarse la intrusión estaría seguro de que alguien acudiría en su ayuda.
- La instancia EC2 de capa gratuita de AWS ha sido de mucha ayuda para la realización de este proyecto, ya que permitió alojar a los servidores ChipStack ofreciendo alta disponibilidad y seguridad al mismo tiempo.
- El proceso y herramientas establecidas por el marco de trabajo Scrum demostraron ser de gran utilidad para garantizar y agilizar el desarrollo del prototipo, permitiendo atender mejor las necesidades y deseos del usuario final, mediante la entrega temprana de funcionalidades.

6.2. RECOMENDACIONES

- El prototipo fue desarrollado y puesto a prueba en una zona urbana, por esta razón, se recomienda a futuros estudiantes que tengan interés en este proyecto, implementarlo en una zona rural, para hacer comparaciones con los resultados obtenidos.
- En el caso de querer implementar soluciones similares al aire libre, es recomendable hacerlo con puertas de enlace multicanal y de tipo outdoor ya que tienen resistencia al polvo y al agua, además cumplen completamente con el estándar LoRaWAN.
- Para proporcionar un mayor alcance de la red, se recomienda hacer uso de antenas de fibra de vidrio, las cuales son ideales para trabajos pesados y brindan una ganancia de 6dBi a diferencia de los 2 dBi ofrecidos por las antenas de los componentes utilizados.
- Es importante verificar que las antenas de los componentes LoRa se encuentren correctamente colocadas antes de encenderlos, pues el chip LoRA podría dañarse por el exceso de energía que se genera al no encontrarse estas conectadas.
- Para impulsar el desarrollo del prototipo propuesto se debe considerar la implementación de servidores con una arquitectura altamente escalable capaz de gestionar un gran volumen de datos, debido a que la solución requiere de acceso y manipulación de los datos en tiempo real. También es esencial contar con escalabilidad elástica para integrarse en una infraestructura Cloud.

6.3. POSIBLE IMPLEMENTACIÓN

Al concluir con el desarrollo de la solución propuesta, se ha tomado en cuenta que la implementación de este proyecto puede servir para brindar apoyo e incluso automatizar el proceso “Encargo de Domicilio” utilizado actualmente por las Unidades de Policía Comunitaria [102]. Este proceso pretende evitar el robo de viviendas, a través del patrullaje focalizado y visitas constantes a los inmuebles durante la ausencia temporal de sus propietarios. El objetivo del proceso es ofrecer seguridad a las viviendas de los ciudadanos que tengan la necesidad de ausentarse temporalmente de su domicilio, por viajes o casos fortuitos. Para solicitar el servicio, el dueño de la propiedad debe ingresar al sitio en línea https://siipne.policia.gob.ec/indexSiipne.php y llenar un formulario en el cual colocará la fecha de inicio y fin del encargo.

Mediante la implementación de la red LoRaWAN, de la aplicación web y de la aplicación móvil tanto el personal de la Policía Nacional como el propietario de la vivienda podrán recibir notificaciones o mensajes de alerta en tiempo real cuando los dispositivos finales

detecten cualquier intrusión en el domicilio. De esta manera, los miembros encargados de patrullar la zona no tendrán la necesidad de verificar constantemente si existe algún inconveniente o problema con la vivienda, solo acudirán cuando hayan recibido un mensaje de alerta de detección de intrusos o un aviso de problema de conexión del dispositivo final, lo cual les permitirá enfocarse en otro tipo de actividades que ayuden al bienestar de la comunidad.

REFERENCIAS BIBLIOGRÁFICAS

[1] A. Bera, “Burglary Statistics (Infographic),” SafeAtlast, 2019. [Online]. Available: http s://safeatlast.co/blog/burglary-statistics/. [Accessed: 30-Jun-2019].

[2] J. Espín, “Delitos contra la propiedad: el mayor problema de inseguridad ciudadana en el DMQ,” FLACSO Ecuador, 2014. [Online]. Available: http s://repositorio.flacsoandes.edu.ec/bitstream/10469/2294/1/BFLACSO-CS28-04-Espín.pdf. [Accessed: 30-Jun-2019].

[3] R. Muggah, “The Rise of Citizen Security in Latin America and the Caribbean,” OpenEdition, 2017. [Online]. Available: http s://journals.op enedition.org/p oldev/2377. [Accessed: 4-Jul-2019].

[4] SEMTECH, “What is LoRa?,” SEMTECH, 2019. [Online]. Available: http s://www.semtech.com/lora. [Accessed: 4-Jul-2019].

[5] “Venta Departamento en Quito, Pichincha (85715)- iCasas.ec", Icasas.ec, 2020. [Online]. Available: https://www.icasas.ec/propiedad/85715. [Accessed: 5-Jul- 2020]

[6] S. Sruthy and S. N. George, "WiFi enabled home security surveillance system using Raspberry Pi and IoT module," 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Kollam, 2017, pp. 1-6, doi: 10.1109/SPICES.2017.8091320.

[7] X. Mao, K. Li, Z. Zhang and J. Liang, "Design and implementation of a new smart home control system based on internet of things," 2017 International Smart Cities Conference (ISC2), Wuxi, 2017, pp. 1-5, doi: 10.1109/ISC2.2017.8090790.

[8] A. Javare, T. Ghayal, J. Dabhade, A. Shelar and A. Gupta, "Access control and intrusion detection in door lock system using Bluetooth technology," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 2246-2251, doi: 10.1109/ICECDS.2017.8389852.

[9] S. Kumar and S. R. Lee, "Android based smart home system with control via Bluetooth and internet 615 connectivity," The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014), JeJu Island, 616 2014, pp. 1-2, doi: 10.1109/ISCE.2014.6884302. 617.

[10] L. Tanutam and W. Atmadja, “Home Security System with IOT Based Sensors Running On House Infra Structure Platform,” 2019 IOP Conference Series: Earth and Environmental Science, 2020, doi:10.1088/1755-1315/426/1/012151.

[11]. J. Souifi, Y. Bouslimani, M. Ghribi, A. Kaddouri, T. Boutot and H. H. Abdallah, "Smart Home Architecture based on LoRa Wireless Connectivity and LoRaWAN® Networking Protocol," 020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP), EL OUED, Algeria, 2020, pp. 95-99, doi: 10.1109/CCSSP49278.2020.9151815

[12] "¿Qué es una Smart City? - iUrban", iUrban, 2020. [Online]. Available: https://iurban.es/que-es-una-smart-city. [Accessed: 02- Jul- 2020]

[13] "Qué es una Smart city: tecnología, ventajas y seguridad | VIU", Universidadviu.com, 2020. [Online]. Available: https://www.universidadviu.com/que-es-una-smart-city-tecnologia-ventajas-y-seguridad/. [Accessed: 04- Jul- 2020]

[14] C. Manville, J. Millard and J. Paderson, "Mapping Smart Cities In The EU", Europarl.europa.eu, 2014. [Online]. Available: https://www.europarl.europa.eu/RegData/etudes/etudes/join/2014/507480/IPOL-ITRE_ET(2014)507480_EN.pdf. [Accessed: 05- Jul- 2020]

[15] K. Rose, S, Eldridge and L. Chapin, "La Internet De Las Cosas - Una breve Reseña", Internetsociety.org, 2015. [Online]. Available: https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf. [Accessed: 29- Jul- 2020]

[16] "¿Qué es el Internet de las cosas (IoT)?", redhat.com, 2020. [Online]. Available: https://www.redhat.com/es/topics/internet-of-things/what-is-iot. [Accessed: 29- Jul- 2020]

[17] Y. Mohamed, " What is the role of IoT in Smart Cities? Finextra Research, 2019. [Online]. Available: https://www.finextra.com/blogposting/17931/what-is-the-role-of-iot-in-smart-cities#:~:text=The%20IoT%20includes%20only%20smart%20sensors%20and%20other%20devices.&text=IoT%20offers%20new%20opportunities%20for,for%20monitoring%20widely%20dispersed%20processes. [Accessed: 18- Jul- 2020]

[18] "Internet de las cosas (IoT) y Seguridad Inteligente -", Instalacionestk.com, 2019. [Online]. Available: https://www.instalacionestk.com/iot-seguridad-inteligente/. [Accessed: 19- Jul- 2020]

[19] "Tecnologías de Comunicación para IoT", Efor.es, 2020. [Online]. Available: https://www.efor.es/sites/default/files/tecnologias-de-comunicacioon-para-iot.pdf. [Accessed: 23- Jul- 2020]

[20] C. Hayes "Gestión de la potencia IoT con tecnología LPWAN", Electronicspecifier.com, 2019. [Online]. Available: https://www.electronicspecifier.com/news/analysis/managing-power-in-the-iot-with-cellular-lpwan-technology. [Accessed: 23- Jul- 2020]

[21] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “A comparative study of LPWAN technologies for large-scale IoT deployment” *ICT Express*, p. 7, 2017. [online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959517302953/pdfft?md5=66f3fc165d056677b9f8d369645f8b6a&pid=1-s2.0-S2405959517302953-main.pdf. [Accessed: 23- Jul- 2020]

[22] "LPWAN | MYTHINGS, LoRa, NB-IoT, Sigfox | BehrTech", BehrTech, 2020. [Online]. Available: https://behrtech.com/lpwan-technology/. [Accessed: 30- Jul- 2020]

[23] "5 Things to know about the LPWAN market in 2020", Iot-analytics.com, 2020. [Online]. Available: https://iot-analytics.com/5-things-to-know-about-the-lpwan-market-in-2020/. [Accessed: 30- Jul- 2020]

[24] "Our story | Sigfox", Sigfox.com, 2020. [Online]. Available: https://www.sigfox.com/en/sigfox-story. [Accessed: 31- Jul- 2020]

[25] "Sigfox Buy", Buy.sigfox.com, 2020. [Online]. Available: https://buy.sigfox.com/buy/offers/US. [Accessed: 31- Jul- 2020]

[26] "¿Qué es Sigfox? | Sigfox build", Build.sigfox.com, 2020. [Online]. Available: https://build.sigfox.com/sigfox. [Accessed: 31- Jul- 2020]

[27] "Internet of Things (NB-IoT) | Internet of Things", Internet of Things, 2020. [Online]. Available: https://www.gsma.com/iot/mobile-iot-technology-nb-iot/. [Accessed: 01- Aug- 2020]

[28] "Mobile IoT LPWA - LTE-M & NB-IoT Commercial Launches | GSMA", Internet of Things, 2020. [Online]. Available: https://www.gsma.com/iot/mobile-iot-commercial-launches/. [Accessed: 01- Aug- 2020]

[29] S. Shah, "T-Mobile reveals $6 per device price tag for Magenta NB-IoT plan | Internet of Business", Internet of Business, 2020. [Online]. Available: https://internetofbusiness.com/t-mobile-price-nb-iot-connectivity/. [Accessed: 01- Aug- 2020]

[30] B. Ray, "LTE-M Vs. NB-IoT: ¿What's The Difference?» Iota Communications, Inc.", Iota Communications, Inc., 2020. [Online]. Available: https://www.iotacommunications.com/blog/lte-m-vs-nb-iot/. [Accessed: 02- Aug- 2020]

[31] Y. Hwang, "Explicación de IoT celular: NB-IoT vs. LTE-M vs. 5G y más", Iotforall.com, 2020. [Online]. Available: https://www.iotforall.com/cellular-iot-explained-nb-iot-vs-lte-m/. [Accessed: 02- Aug- 2020]

[32] I. Things and L. (LPWA), "NB-IoT, LTE-M & LPWA Network Solutions at AT&T Business", Business.att.com, 2020. [Online]. Available: https://www.business.att.com/products/lpwa.html. [Accessed: 02- Aug- 2020]

[33] "Tecnologías de red IoT celular | Citykinect", Citykinect, 2020. [Online]. Available: https://citykinect.com/technologies/cellular-lpwan-networks/. [Accessed: 02- Aug- 2020]

[34] "About LoRa Alliance® | LoRa Alliance®", Lora-alliance.org, 2020. [Online]. Available: https://lora-alliance.org/about-lora-alliance. [Accessed: 03- Aug- 2020]

[35] "Home page | LoRa Alliance®", Lora-alliance.org, 2020. [Online]. Available: https://lora-alliance.org/. [Accessed: 03- Aug- 2020]

[36] "LoRaWAN. What is it? A technical overview of loRa and loRaWAN", Lora-alliance.org, 2015. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf. [Accessed: 04- Aug- 2020]

[37] L. Slats, "Una breve historia de LoRa: tres inventores comparten su historia personal en la conferencia The Things", Blog.semtech.com, 2020. [Online]. Available: https://blog.semtech.com/a-brief-history-of-lora-three-inventors-share-their-personal-story-at-the-things-conference. [Accessed: 5- Aug- 2020]

[38] E. Hewitson "¿Cuál es la diferencia entre LoRa y LoRaWAN?", Wyld Networks, 2020. [Online]. Available: https://wyldnetworks.com/what-is-the-difference-between-lora-and-lorawan/. [Accessed: 10- Aug- 2020]

[39] "LoRa and LoRaWAN: A Technical Overview | DEVELOPER PORTAL", Lora-developers.semtech.com, 2019. [Online]. Available: https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf. [Accessed: 10- Aug- 2020]

[40] LoRa Alliance Technical committee, "LoRaWAN® Regional Parameters v1.0.2rB | LoRa Alliance®", Lora-alliance.org, 2017. [Online]. Available: https://lora-

alliance.org/sites/default/files/2018-05/lorawan_regional_parameters_v1.0.2_final_1944_1.pdf. [Accessed: 6- Aug- 2020]

[41] "Frequency Bands allocated to Terrestrial Broadcasting Services", Itu.int, 2018. [Online]. Available: https://www.itu.int/en/ITU-R/terrestrial/broadcast/Pages/Bands.aspx. [Accessed: 8- Aug- 2020]

[42] R. Lie, "Mobilefish.com - LoRa/LoRaWAN tutorial.", Mobilefish.com, 2019. [Online]. Available: https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_tutorial.html. [Accessed: 11- Aug- 2020]

[43] E. Ruano Lin, "LoRa protocol. Evaluations, limitations and practical test", Upcommons.upc.edu, 2016. [Online]. Available: https://upcommons.upc.edu/handle/2117/98853. [Accessed: 11- Aug- 2020]

[44] T. Telkamp, "Curso intensivo de LoRa por Thomas Telkamp", The Things Network, 2016. [Online]. Available: https://drive.google.com/file/d/0B49WRBg1NvMgQmdlM3doaWdVMWc/view. [Accessed: 11- Aug- 2020]

[45] M. Casanova, "El protocolo LoRaWAN {Todo lo que debes saber]", Automatización del Internet de las Cosas, SL, 2020. [Online]. Available: https://alfaiot.com/blog/ultimas-noticias-2/post/el-protocolo-lorawan-6. [Accessed: 20- Aug- 2020]

[46] N. Somin, M. Luis, T. Eirich, T. Kramp and O. Hersent, "LoRaWAN® Specification v1.0.2 | LoRa Alliance®", Lora-alliance.org, 2016. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-05/lorawan1_0_2-20161012_1398_1.pdf. [Accessed: 22- Aug- 2020]

[47] T. Bouguera, J. Diouris, J. Chaillout, R. Jaouadi and G. Andrieux, "Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN", Hal.archives-ouvertes.fr, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01828769/document. [Accessed: 23- Aug- 2020]

[48] Gemalto, Actility and Semtech, "LoRaWAN Security. Full end-to-end encryption for IoT application providers", Lora-alliance.org, 2017. [Online]. Available: https://lora-alliance.org/sites/default/files/2019-05/lorawan_security_whitepaper.pdf. [Accessed: 24- Aug- 2020]

[49] "Security", The Things Networ0k, 2020. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/security.html. [Accessed: 25- Aug- 2020]

[50] "Addressing & Activation", The Things Network, 2020. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/addressing.html. [Accessed: 26- Aug- 2020]

[51] T. Michaslki, "Explicando LoRaWAN", Ubidots, 2017. [Online]. Available: https://ubidots.com/blog/explaining-lorawan/. [Accessed: 26- Aug- 2020]

[52] R. Barry and J. Meijers, "IoT connectivity comparison (GSM vs LoRa vs Sigfox vs NB-Iot)", Polymorph, 2019. [Online]. Available: https://www.polymorph.co.za/iot-connectivity-comparison-gsm-vs-lora-vs-sigfox-vs-nb-iot/. [Accessed: 28- Aug- 2020.

[53] Maida, EG, Pacienzia, J. Metodologías de desarrollo de software [Online]. Tesis de Licenciatura en Sistemas y Computación. Facultad de Química e Ingeniería “Fray Rogelio Bacon”. Universidad Católica Argentina, 2015. Available en: http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf [Accessed: 18- Oct- 2020].

[54] "The Benefits of Adhering to a Software Development Methodology", *Segue Technologies*, 2020. [Online]. Available: https://www.seguetech.com/benefits-adhering-software-development-methodology-concepts/. [Accessed: 18- Oct- 2020].

[55] "¿Qué es una metodología de desarrollo de software? | 4R Soluciones | Diseño, Desarrollo y Programación Web & Mobile", *4R Soluciones | Diseño, Desarrollo y Programación Web & Mobile*, 2013. [Online]. Available: https://www.4rsoluciones.com/blog/una-metodologia-desarrollo-software. [Accessed: 18- Oct- 2020].

[56] "Metodología tradicional o ágil ¿Cuál es la mejor opción para mi proyecto de desarrollo de software? - Scio México", *Scio México*, 2019. [Online]. Available: https://www.scio.com.mx/blog/metodologia-tradicional-o-agil-software/. [Accessed: 18- Oct- 2020].

[57] ALFARO-HERRERA, Julio César, SÁNCHEZ-DELGADO, Octavio, CORDOVA-OSORIO, Luis Alberto, VALENTÍN-JIMÉNEZ, Carlos Miguel. Scrum como metodología para proyectos de redes. Revista de Tecnologías Computacionales. 2017. 1-1:38-42.

[58] "Scrum Guide | Scrum Guides", *Scrumguides.org*, 2020. [Online]. Available: https://www.scrumguides.org/scrum-guide.html. [Accessed: 18- Oct- 2020].

[59] Heltec, "WIFI LoRa 32 (V2)", *Heltec Automation*, 2018. [Online]. Available: https://heltec.org/project/wifi-lora-32/. [Accessed: 06- Nov- 2020].

[60] Punto Flotante, "Manual del sensor infrarrojo de movimiento PIR HC-SR501", *Puntoflotante.net*, 2017. [Online]. Available: https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf. [Accessed: 06- Nov- 2020].

[61] "Arduino - Door Sensor | Arduino Tutorial", *Arduino Getting Started*, 2020. [Online]. Available: https://arduinogetstarted.com/tutorials/arduino-door-sensor. [Accessed: 09- Nov- 2020].

[62] H. Visser, "The future of single-channel gateways", *The Things Network*, 2018. [Online]. Available: https://www.thethingsnetwork.org/forum/t/the-future-of-single-channel-gateways/6590. [Accessed: 11- Nov- 2020].

[63] "WisGate Edge Lite | RAK7258", *RAKwireless Store*, 2020. [Online]. Available: https://store.rakwireless.com/products/rak7258-micro-gateway?variant=27155353174116. [Accessed: 11- Nov- 2020].

[64] RAKwireless, *Downloads.rakwireless.com*, 2020. [Online]. Available: https://downloads.rakwireless.com/LoRa/Indoor-Gateway-RAK7258/Hardware-Specification/Indoor_Gateway_RAK7258_Product_Brief_V1.2.pdf. [Accessed: 11- Nov- 2020].

[65] "Duty Cycle", The Things Network, 2020. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/duty-cycle.html#maximum-duty-cycle. [Accessed: 11- Nov- 2020].

[66] "ChirpStack architecture - ChirpStack open-source LoRaWAN® Network Server", Chirpstack.io, 2020. [Online]. Available: https://www.chirpstack.io/project/architecture/. [Accessed: 14- Nov- 2020].

[67] "HTML: HyperText Markup Language", *MDN Web Docs*, 2005. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML. [Accessed: 16- Nov- 2020].

[68] "CSS: Cascading Style Sheets", *MDN Web Docs*, 2005. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS. [Accessed: 22- Nov- 2020].

[69] "JavaScript", *MDN Web Docs*, 2005. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript. [Accessed: 22- Nov- 2020].

[70] "Typed JavaScript at Any Scale.", *Typescriptlang.org*, 2012. [Online]. Available: https://www.typescriptlang.org/. [Accessed: 22- Nov- 2020].

[71] E. Vicente Bonet, "Lenguaje C", Informatica.uv.es, 2020. [Online]. Available: https://informatica.uv.es/estguia/ATD/apuntes/laboratorio/Lenguaje-C.pdf. [Accessed: 22- Nov- 2020].

[72] "Angular", *Angular.io*, 2010. [Online]. Available: https://angular.io/docs. [Accessed: 22- Nov- 2020].

[73] "Las 5 principales ventajas de usar Angular para crear aplicaciones web - campusMVP.es", *campusMVP.es*, 2017. [Online]. Available: https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx. [Accessed: 22- Nov- 2020].

[74] "Ionic - Cross-Platform Mobile App Development", *Ionic Framework*, 2020. [Online]. Available: https://ionicframework.com/. [Accessed: 22- Nov- 2020].

[75] T. Company, "Sails.js | Realtime MVC Framework for Node.js", *Sailsjs.com*, 2012. [Online]. Available: https://sailsjs.com/. [Accessed: 22- Nov- 2020].

[76] "Acerca | Node.js", *Node.js*, 2020. [Online]. Available: https://nodejs.org/es/about/. [Accessed: 22- Nov- 2020].

[77] “V. Agafonkin, "Leaflet — an open-source JavaScript library for interactive maps", *Leafletjs.com*, 2010. [Online]. Available: https://leafletjs.com/. [Accessed: 22- Nov- 2020].

[78] “E. Decena, "Mosca un servidor en tiempo real para IoT", *Medium*, 2018. [Online]. Available: https://medium.com/@eddydecena/mosca-un-servidor-en-tiempo-real-para-iot-a963ed008320. [Accessed: 22- Nov- 2020].

[79] J. López Quijado, "LA LIBRERÍA EEPROM – Recursos para programadores", Eldesvandejose.com, 2016. [Online]. Available: https://eldesvandejose.com/2016/04/01/la-libreria-eeprom/. [Accessed: 10- Dec- 2020].

[80] “¿Qué es AWS?", Amazon Web Services, Inc., 2020. [Online]. Available: https://aws.amazon.com/es/what-is-aws/. [Accessed: 22- Nov- 2020].

[81] "ChirpStack open-source LoRaWAN® Network Server", Chirpstack.io, 2020. [Online]. Available: https://www.chirpstack.io/. [Accessed: 22- Nov- 2020].

[82] "#1 Push Service | Send Mobile & Web Push Notifications", *OneSignal*, 2020. [Online]. Available: https://onesignal.com/. [Accessed: 22- Nov- 2020].

[83] "Get Started with Docker | Docker", *Docker*, 2020. [Online]. Available: https://www.docker.com/get-started. [Accessed: 22- Nov- 2020].

[84] "Download PuTTY - a free SSH and telnet client for Windows", Putty.org, 2020. [Online]. Available: https://www.putty.org/. [Accessed: 22- Nov- 2020].

[85] Á. Robledano, "Qué es MySQL: Características y ventajas", *OpenWebinars.net*, 2019. [Online]. Available: https://openwebinars.net/blog/que-es-mysql/. [Accessed: 22- Nov- 2020].

[86] V. Code, "Documentation for Visual Studio Code", *Code.visualstudio.com*, 2020. [Online]. Available: https://code.visualstudio.com/docs. [Accessed: 22- Nov- 2020].

[87] "Arduino IDE 1.8.13", Software | Arduino, 2020. [Online]. Available: https://www.arduino.cc/en/Guide. [Accessed: 22- Nov- 2020].

[88] "Git", *Git-scm.com*, 2020. [Online]. Available: https://git-scm.com/. [Accessed: 22- Nov- 2020].

[89] "¿Qué Es GitHub Y Para Qué Se Utiliza?", *Tutoriales Hostinger*, 2019. [Online]. Available: https://www.hostinger.es/tutoriales/que-es-github/. [Accessed: 22- Nov- 2020].

[90] "GitHub Desktop Documentation - GitHub Docs", *Docs.github.com*, 2020. [Online]. Available: https://docs.github.com/en/free-pro-team@latest/desktop. [Accessed: 22- Nov- 2020].

[91] "Visualiza tus tus procesos, personal y sistemas con diagramas | Lucidchart", *Lucidchart.com*, 2020. [Online]. Available: https://www.lucidchart.com/pages/es/producto. [Accessed: 22- Nov- 2020].

[92] "Flaticon, la mayor base de datos de iconos vectoriales gratis", *Flaticon*, 2010. [Online]. Available: https://www.flaticon.es/. [Accessed: 22- Nov- 2020].

[93] "Adobe: Soluciones de creatividad, marketing y gestión de documentos", *Adobe: Creative, marketing and document management solutions*, 2020. [Online]. Available: https://www.adobe.com/la/. [Accessed: 22- Nov- 2020].

[94] "Arduino Downloads", Arduino, 2020. [Online]. Available: https://www.arduino.cc/en/software. [Accessed: 17- Nov- 2020]

[95] "HelTecAutomation/ESP32_LoRaWAN", *GitHub*, 2020. [Online]. Available: https://github.com/HelTecAutomation/ESP32_LoRaWAN. [Accessed: 26- Nov- 2020].

[96] "查询序列号", Resource.heltec.cn, 2020. [Online]. Available: https://resource.heltec.cn/search/. [Accessed: 26- Nov- 2020].

[97] "RAKWireless/ChirpStack_on_Ubuntu", GitHub, 2020. [Online]. Available: https://github.com/RAKWireless/chirpstack_on_ubuntu. [Accessed: 30- Nov- 2020].

[98] "Web Push Quickstart", *OneSignal Push Notification Service Documentation*, 2020. [Online]. Available: https://documentation.onesignal.com/docs/web-push-quickstart. [Accessed: 08- Nov- 2020].

[99] "Generate a Firebase Server Key", *OneSignal Push Notification Service Documentation*, 2020. [Online]. Available: https://documentation.onesignal.com/docs/generate-a-google-server-api-key. [Accessed: 08- Nov- 2020].

[100] S. Xalambrí, "Introducción a JSON Web Tokens", *Introducción a JSON Web Tokens*, 2017. [Online]. Available: https://medium.com/@sergiodxa/introducci%C3%B3n-a-json-web-tokens-43f22f67a122. [Accessed: 07- Nov- 2020].

[101] "Home Burglary Awareness and Prevention", Jsu.edu, 2013. [Online]. Available: http://www.jsu.edu/police/docs/Schoolsafety.pdf. [Accessed: 29- Nov- 2020].

[102] "Visita Periódica a Domicilio "Encargo de Domicilio" | Ecuador - Guía Oficial de Trámites y Servicios", Gob.ec, 2020. [Online]. Available: https://www.gob.ec/pn/tramites/visita-periodica-domicilio-encargo-domicilio. [Accessed: 27- Nov- 2020].

ANEXOS

Anexo 1. Diagramas de la tarjeta de desarrollo WiFi LoRa 32 (V2).

Anexo 2. Sprint backlog y Sprint Review

Anexo 3. Código fuente de los nodos detectores de intrusión.

Anexo 4. Códec para decodificar la carga útil del dispositivo.

Anexo 5. Mockups de las aplicaciones web y móvil.

Anexo 6. Manual de usuario.