UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACSISTEL

INGENIERÍA EN TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

Implementación de una red LoRaWAN basada en sistemas IoT para el análisis de la seguridad LPWAN

Laily Jamilex Borbor Basilio

Dirigido por:

Ing. Fernando Chamba Macas, Mgt.

La Libertad - 2026

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. Ronald Rovira Jurado, PhD.
DIRECTOR DE LA CARRERA

Ing. Fernando Chamba Macas, Mgt.
TUTOR

Ing. Luis Amaya Fariño, Mgtr.
DOCENTE ESPECIALISTA

Ing. Luis Amaya Fariño, Mgt.
DOCENTE GUÍA UIC

Ing. Corina Gonzabay De La A,Mgt.
SECRETARIA

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **Borbor Basilio Laily Jamilex**, como requerimiento para la obtención del título de Ingeniero en telecomunicaciones.

La Libertad, a los 20 días del mes de agosto del año 2026

TUTOR

Ing. Fernando Vinicio Chamba Macas, Mgt.

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Borbor Basilio Laily Jamilex**

Autorizo a la Universidad Estatal Península de Santa Elena para que haga de este trabajo de integración curricular un documento disponible para su lectura, consulta y procesos de investigación, de conformidad con las normas de la Institución.

Cedo los derechos patrimoniales en línea de este artículo profesional de alto nivel con fines de difusión pública. Asimismo, apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando dicha reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor.

La Libertad, a los 20 días del mes de agosto del año 2026.

EL AUTOR

Laily Borbor

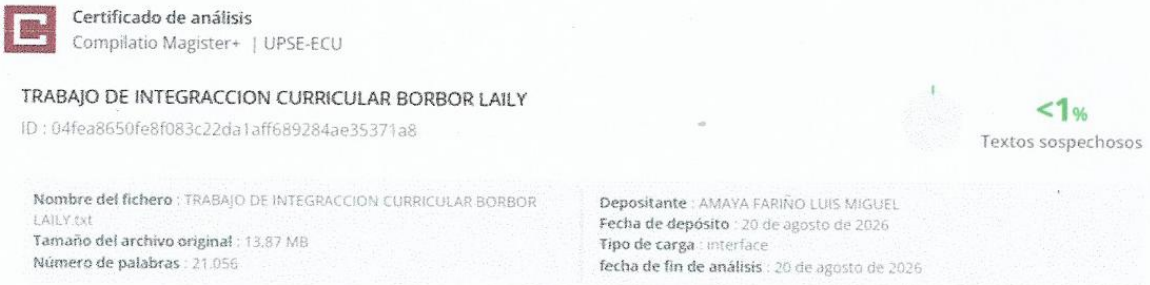
Borbor Basilio Laily Jamilex

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado **Implementación de una red LoRaWAN basada en sistemas IoT para el análisis de la seguridad LPWA**, presentado por el estudiante, **Borbor Basilio Laily Jamilex** fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 01%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.


Certificado de análisis
Compilatio Magister+ | UPSE-ECU

TRABAJO DE INTEGRACCION CURRICULAR BORBOR LAILY
ID : 04fea8650fe8f083c22da1aff689284ae35371a8

<1%
Textos sospechosos

Nombre del fichero : TRABAJO DE INTEGRACCION CURRICULAR BORBOR LAILY.txt
Tamaño del archivo original : 13,87 MB
Número de palabras : 21.056

Depositante : AMAYA FARIÑO LUIS MIGUEL
Fecha de depósito : 20 de agosto de 2026
Tipo de carga : interface
fecha de fin de análisis : 20 de agosto de 2026

TUTOR

Ing. Fernando Vinicio Chamba Macas, Mgt.

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Borbor Basilio Laily Jamilex**

DECLARO QUE:

El trabajo de integración Curricular, **Implementación de una red LoRaWAN basada en sistemas IoT para el análisis de la seguridad LPWAN,** previo a la obtención del título en Ingeniera en Telecomunicaciones, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 20 días del mes de agosto del año 2026.

EL AUTOR

Laily Borbor.

Borbor Basilio Laily Jamilex

AGRADECIMIENTO

A Dios, por concederme la sabiduría, la fortaleza y la perseverancia necesarias para superar cada desafío durante todo el proceso de mi formación universitaria.

A mis padres Luis Borbor y Azucena Basilio, por su amor incondicional, esfuerzo y sacrificio. Gracias por creer en mí, brindarme su apoyo en cada decisión y motivarme a seguir adelante incluso en los momentos en que sentí que quería darme por vencida. Este logro también les pertenece porque sin ustedes no habría sido posible alcanzar esta meta.

A mis hermanos por su cariño, apoyo y palabras de aliento que me impulsaron a continuar cuando el camino parecía complicado.

A todas las personas que de una u otra manera formaron parte de este camino, les expreso mi más profundo agradecimiento y eterno reconocimiento.

A mi tutor de tesis, Ing. Fernando Chamba, por compartir sus conocimientos, orientación y experiencia profesional. Agradezco profundamente su tiempo y paciencia, los cuales fueron fundamentales para el desarrollo y la culminación de este trabajo de investigación.

Finalmente, expreso mi sincero agradecimiento a la Universidad Estatal Península de Santa Elena, a la Facultad de Sistemas y Telecomunicaciones y a todos los docentes que formaron parte de mi preparación académica. Gracias por los conocimientos impartidos, que contribuirán significativamente a mi desarrollo como ingeniera en Telecomunicaciones.

Laily Jamilex Borbor Basilio

DEDICATORIA

Dedico este trabajo, en primer lugar, a Dios, por ser la luz que me ha guiado en este camino y me ha permitido alcanzar una de las metas más importantes en mi vida.

A mis padres, quienes con su amor, esfuerzo y ejemplo hicieron posible este objetivo. Les dedico este trabajo por enseñarme que los mayores logros se alcanzan con dedicación y humildad.

Con profundo cariño, dedico este trabajo a la memoria de mi querida hermana, quien, aunque ya no se encuentra físicamente a mi lado, permanece siempre en mi corazón. Estoy segura de que, desde el cielo, me acompaña y se siente orgullosa de este logro que siempre anheló.

Finalmente, dedico este logro a mi futuro, con el compromiso de ejercer mi profesión con responsabilidad, ética y dedicación, llevando siempre conmigo los valores y enseñanzas que han guiado mi vida.

Laily Jamilex Borbor Basilio

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo propone la implementación de una red LoRaWAN basada en sistemas IoT para analizar la seguridad de redes LPWAN. La infraestructura, conformada por un nodo ESP32 LoRa, un Gateway SenseCAP M2, las plataformas como The Things Network, Node-RED y Telegram, permite recolectar, transmitir y visualizar telemetría en condiciones normales de operación. Posteriormente, la red se somete a ataques de desautenticación e inundación de balizas mediante un dispositivo Aursinc Wi-Fi Deauther, analizando el tráfico generado con Kali Linux y Wireshark. Los resultados evidencian interrupciones en la conectividad de la red y retrasos en la transmisión de datos. Con base en estos resultados, se proponen medidas de mitigación orientadas a fortalecer la disponibilidad, integridad y confiabilidad de la red.

Palabras clave: LoRaWAN, IoT, LPWAN, desautenticación.

ABSTRACT

The present study proposes the implementation of a LoRaWAN network based on IoT systems to analyze the security of LPWAN networks. The infrastructure, consisting of an ESP32 LoRa node, a SenseCAP M2 Gateway, and platforms such as The Things Network, Node-RED, and Telegram, enables the collection, transmission, and visualization of telemetry under normal operating conditions. Subsequently, the network is subjected to deauthentication and beacon flooding attacks using an Aursinc Wi-Fi Deauther device, while the generated traffic is analyzed using Kali Linux and Wireshark. The results reveal interruptions in network connectivity and delays in data transmission. Based on these results, mitigation measures are proposed to strengthen the availability, integrity, and reliability of the network.

Keywords: LoRaWAN, IoT, LPWAN, deauthentication.

CAPÍTULO I

1.1 Antecedentes

Las implicaciones de seguridad de las redes IoT han sido estudiadas por varios autores, estableciendo una base sólida para esta investigación. Actualmente, en el sector de las telecomunicaciones, el Internet de las Cosas (IoT) se presenta como una de las tecnologías más prometedoras. Esto se debe principalmente a su capacidad para conectar diferentes dispositivos, automatizar procesos y gestionar información en tiempo real [1].

Sin embargo, esta expansión también ha generado efectos negativos en el ámbito de la seguridad digital. El aumento del número de dispositivos conectados a una red incrementa los posibles puntos vulnerables, lo que eleva el riesgo de comprometer la disponibilidad, la integridad o la confidencialidad de los datos.

Entre los estudios anteriores, el trabajo de Ricardo Plúas y Olaves Rosales resulta especialmente relevante. Este estudio se enfoca en establecer la configuración entre el router y la puerta de enlace para asegurar una conexión estable con el servidor central. Esta información es útil para el diseño de la arquitectura de comunicación de este trabajo de integración curricular, ya que proporciona lineamientos específicos para mejorar el rendimiento del enlace, seleccionar las frecuencias adecuadas y reducir la pérdida de información durante la transmisión [2].

Por otro lado, como segunda referencia, se consideró un trabajo de investigación que evidenció los efectos de los ataques de desautenticación y de la generación de redes inalámbricas falsas sobre la disponibilidad y estabilidad de las redes inalámbricas. Para ello, se utilizó el dispositivo Aursinc Wi-Fi Deauther, mediante el cual se realizaron interrupciones intencionales de la conectividad. Los hallazgos mostraron una disminución significativa en la disponibilidad de la red y un incremento en la inestabilidad del enlace, lo que resalta la necesidad de fortalecer las medidas de seguridad en las redes IoT [3].

En línea con esta investigación, Soto La Rosa y Yagual Laínez implementaron una red IoT basada en tecnología LoRa en un entorno de laboratorio. Para ello, utilizaron nodos y una puerta de enlace con el propósito de establecer la infraestructura de comunicación y evaluar parámetros técnicos, como el factor de dispersión, la potencia de transmisión y el ancho de banda, bajo diferentes condiciones de operación. Además, incorporaron un nodo interferente para analizar los efectos de condiciones adversas sobre la estabilidad del enlace. Este trabajo aportó criterios técnicos para la evaluación experimental de redes IoT y sirvió como referencia para el planteamiento de escenarios de prueba bajo condiciones controladas [4].

Desde una perspectiva más amplia, informes recientes advierten que la reducción de los costos asociados a las herramientas empleadas en ataques cibernéticos, favorecida en parte por los avances en inteligencia artificial, ha incrementado la accesibilidad para la ejecución de ataques de denegación de servicio distribuido (DDoS) a gran escala. Los datos reportados evidencian que la infraestructura del Internet de las Cosas (IoT) constituye una superficie susceptible de ser comprometida, debido a la incorporación de dispositivos IoT, routers y servidores vulnerables o previamente comprometidos en la generación y propagación de tráfico malicioso. Esta situación representa un riesgo para la disponibilidad de los servicios y resalta la necesidad de implementar mecanismos de protección y monitoreo orientados a fortalecer la seguridad de las infraestructuras IoT [5].

Los estudios mencionados abordan aspectos de conectividad y seguridad en redes IoT. Sin embargo, ninguno evalúa conjuntamente una red LoRaWAN frente a ataques en un entorno controlado. Por ello, este proyecto propone analizar su comportamiento, identificar vulnerabilidades y evaluar el impacto de los ataques sobre la comunicación.

1.2 Descripción del proyecto

El proyecto tiene como propósito implementar una red IoT basada en tecnología LoRaWAN en un entorno de laboratorio controlado, con el fin de analizar su comportamiento ante escenarios que puedan afectar la conectividad inalámbrica utilizada por el sistema. La infraestructura estará conformada por nodos sensores y una puerta de enlace LoRaWAN, encargados de recopilar, transmitir y gestionar datos de telemetría [6].

Inicialmente, se establecerá la configuración de la red y se verificará el funcionamiento de los dispositivos bajo condiciones normales de operación. Esta etapa permitirá comprobar la correcta comunicación entre los nodos, la puerta de enlace y la plataforma de gestión antes de realizar las pruebas de seguridad.

Posteriormente, se implementará un entorno controlado para evaluar el comportamiento de la red ante diferentes escenarios de ataque sobre la conectividad inalámbrica. Las pruebas permitirán observar posibles variaciones en la disponibilidad de la comunicación y en la transmisión de los datos generados por los nodos IoT [7].

Finalmente, se analizarán los resultados obtenidos durante las pruebas para identificar los principales efectos sobre el funcionamiento del sistema. A partir de este análisis, se plantearán medidas de mitigación orientadas a fortalecer la estabilidad y seguridad de la infraestructura IoT implementada.

1.3 Objetivos

1.3.1 Objetivo general

Implementar una red LoRaWAN basada en un sistema IoT para evaluar su vulnerabilidad ante ataques de deauthentication y beacon flooding sobre la conectividad Wi-Fi y la telemetría del sistema, con el fin de proponer medidas que fortalezcan la seguridad de la red.

1.3.2 Objetivos específicos

- Desarrollar una red LPWAN basada en un sistema IoT para la gestión de datos de telemetría, integrando sensores, nodo LoRa, Gateway y la plataforma TTN, garantizando la correcta entrega y visualización de la información.
- Desarrollar un escenario de evaluación de seguridad mediante la ejecución de ataques de deauthentication y beacon flooding sobre la red Wi-Fi, utilizando Kali Linux y Wireshark para analizar su efecto en la conectividad.
- Analizar el desempeño de la red IoT/LPWAN antes y durante de los escenarios de ataque, evaluando el impacto en la telemetría y conectividad del sistema, para proponer medidas de mitigación que fortalezcan su integridad, disponibilidad y confiabilidad.

1.4 Justificación

El crecimiento de las redes IoT en diversos sectores ha incrementado la necesidad de fortalecer la seguridad de las infraestructuras inalámbricas y garantizar la continuidad operativa de sus servicios. En este contexto, los ataques de desautenticación y de inundación de balizas pueden provocar la degradación o interrupción de la comunicación inalámbrica, afectando la disponibilidad y el funcionamiento de los sistemas IoT [8].

Por esta razón, resulta necesario evaluar el comportamiento de una red IoT basada en LoRaWAN frente a estos ataques bajo condiciones controladas. Para ello, se implementará un entorno de laboratorio que permita analizar las alteraciones producidas en la comunicación y en la transmisión de datos de telemetría, sin comprometer infraestructuras externas ni sistemas en producción.

A partir de las pruebas experimentales, se podrán identificar vulnerabilidades asociadas a la disponibilidad de la comunicación y determinar el impacto de los ataques sobre el funcionamiento de la red. Los resultados obtenidos servirán como base para establecer medidas de mitigación orientadas a fortalecer la estabilidad y seguridad de la infraestructura implementada.

En el ámbito académico y técnico, el estudio aportará evidencia experimental sobre el comportamiento de una red IoT basada en LoRaWAN frente a estos ataques y podrá constituir una referencia para futuras investigaciones relacionadas con la evaluación de vulnerabilidades y mecanismos de protección en redes inalámbricas.

1.5 Problemática

Las redes IoT basadas en LoRaWAN permiten recopilar y transmitir datos de telemetría para aplicaciones de monitoreo y generación de alertas. En este tipo de infraestructura, la comunicación del Gateway con los servicios de red requiere un enlace de acceso a Internet, que puede establecerse mediante una red inalámbrica Wi-Fi. La disponibilidad de este enlace es necesaria para garantizar la transmisión de los datos hacia las plataformas de gestión.

Cuando el enlace Wi-Fi presenta interrupciones o saturación como consecuencia de escenarios de ataque, la comunicación entre el Gateway y la infraestructura de Internet puede verse afectada temporalmente. Como consecuencia, pueden generarse retrasos o pérdidas de información hacia plataformas como TTN, Node-RED y Telegram, afectando la visualización de los datos y la generación de alertas.Esta situación puede comprometer la continuidad de los servicios de monitoreo que requieren comunicación constante para la transmisión de telemetría y detección de eventos.

1.6 Alcance del proyecto

El proyecto comprende el diseño, implementación y evaluación de una red IoT basada en LoRaWAN en un entorno de laboratorio bajo condiciones controladas. La infraestructura estará conformada por nodos sensores, una puerta de enlace (Gateway), un enlace Wi-Fi y plataformas para la visualización y generación de alertas, con el propósito de verificar la transmisión de datos de telemetría durante condiciones normales de operación.

La evaluación de seguridad se realizará sobre el enlace Wi-Fi utilizado por la puerta de enlace para acceder a Internet. Para ello, se empleará el dispositivo Aursinc Wi-Fi Deauther en la ejecución controlada de ataques de desautenticación e inundación de balizas (Beacon Flooding), con el fin de analizar sus efectos sobre la conectividad y la transmisión de telemetría.

Las pruebas permitirán comparar el comportamiento de la infraestructura antes, durante y después de los escenarios de ataque, considerando variables relacionadas con la disponibilidad de la comunicación y la continuidad de la transmisión de datos.

A partir de los resultados obtenidos, se establecerán medidas de mitigación orientadas a fortalecer la disponibilidad, estabilidad y seguridad de la infraestructura IoT implementada, así como una referencia para futuras investigaciones sobre la evaluación de vulnerabilidades en redes basadas en LoRaWAN.

1.7 Metodología

Este trabajo emplea una metodología mixta, con un diseño experimental y un alcance descriptivo, orientados al análisis del comportamiento de una red IoT basada en tecnología LoRaWAN ante escenarios de desautenticación e inundación de balizas.

Inicialmente, se realiza una revisión bibliográfica de libros, artículos científicos, normas técnicas y documentos especializados relacionados con IoT, LoRaWAN, ciberseguridad y ataques inalámbricos. Esta etapa permite establecer las bases teóricas y los criterios técnicos para el desarrollo de la investigación.

Posteriormente, se realiza un análisis descriptivo del comportamiento de la red en condiciones normales y durante los escenarios de ataque, considerando aspectos relacionados con la conectividad, la transmisión de telemetría y la estabilidad del sistema.

Finalmente, se desarrolla la fase experimental mediante la implementación de la infraestructura IoT en un entorno de laboratorio controlado. Durante esta etapa, se ejecutan las pruebas de seguridad y se recopilan datos para comparar el comportamiento de la red en condiciones normales y bajo ataque.

A partir de los resultados obtenidos, se analizan los efectos producidos y se plantean medidas de mitigación orientadas a mejorar la disponibilidad, estabilidad y seguridad de la comunicación en las redes LoRaWAN.

CAPÍTULO II

2.1 Marco conceptual

Este apartado reúne los fundamentos teóricos y conceptuales que sustentan el desarrollo del proyecto, relacionados con las tecnologías de comunicación utilizadas y los aspectos de seguridad involucrados en su evaluación. Por ello, se presentan los fundamentos de IoT y de las redes LPWAN, así como las características de la tecnología LoRa y del protocolo LoRaWAN, considerando su arquitectura, esquema de comunicación y componentes principales. Asimismo, se abordan los fundamentos de la seguridad en redes inalámbricas y los mecanismos de comunicación asociados al estándar

IEEE 802.11, necesarios para comprender el funcionamiento de los ataques de desautenticación e inundación de balizas (Beacon Flooding). Estos ataques constituyen los escenarios de evaluación considerados en este estudio y permiten analizar las alteraciones producidas en la conectividad y disponibilidad de la comunicación inalámbrica.

De esta forma, los temas expuestos en este apartado proporcionan el soporte conceptual y técnico necesario para comprender el diseño de la red, la metodología empleada en el desarrollo del proyecto y el análisis de los datos recopilados durante cada una de las pruebas experimentales.

2.2 Marco teórico

Para el desarrollo del estudio, se consideran los fundamentos técnicos y científicos relacionados con IoT, las redes LPWAN, la tecnología LoRa y el protocolo LoRaWAN, teniendo en cuenta su arquitectura, características de comunicación y principales componentes. Asimismo, se describen los nodos empleados en la implementación experimental, sus características técnicas y los elementos requeridos para el establecimiento de la infraestructura de comunicación. De igual manera, se desarrollan los fundamentos de ciberseguridad en entornos IoT y redes inalámbricas, con énfasis en los ataques de desautenticación y Beacon Flooding, los cuales constituyen los escenarios de prueba evaluados en este trabajo de integración curricular.

Estos fundamentos permiten establecer las bases técnicas necesarias para comprender la estructura experimental, la implementación de la infraestructura, la ejecución de las pruebas y el posterior análisis de los datos obtenidos.

2.2.1 Internet de las Cosas (IoT)

El Internet de las Cosas es un paradigma tecnológico que permite la interconexión de dispositivos físicos mediante redes de comunicación, facilitando el intercambio de información entre sensores, actuadores, plataformas de procesamiento y aplicaciones.

Estos dispositivos son capaces de captar datos del entorno, transmitirlos a través de diferentes tecnologías de comunicación y ejecutar acciones de manera automática o con mínima intervención humana [9].

El crecimiento del IoT en diversos ámbitos ha sido impulsado, en gran medida, por la integración de sensores, procesadores y tecnologías de comunicación inalámbrica, lo que ha permitido automatizar procesos, monitorear actividades de forma remota y tomar decisiones a partir de datos obtenidos en tiempo real. Por tal motivo, el IoT se ha consolidado como un componente fundamental de la transformación digital, favoreciendo el desarrollo de soluciones aplicables en diversos sectores.

De acuerdo con la Unión Internacional de Telecomunicaciones (UIT), el IoT se concibe como una infraestructura de alcance mundial orientada a proporcionar servicios avanzados mediante la interconexión de objetos físicos y virtuales. Estos objetos pueden contar con capacidades de identificación, comunicación e interacción, utilizando tecnologías de información y comunicación que permiten su integración en redes interconectadas [10].

2.2.2 Redes LPWAN

Las redes LPWAN (Low-Power Wide-Area Network) forman parte de un grupo de tecnologías de comunicación inalámbrica diseñadas para conectar dispositivos IoT que necesitan transmitir pequeños volúmenes de información a grandes distancias, con un consumo energético reducido. Estas redes permiten que sensores y dispositivos operen

durante largos períodos utilizando baterías de baja capacidad, lo que las hace ideales para aplicaciones de monitoreo remoto y recopilación de datos [11].

Una característica clave de las redes LPWAN es su capacidad para cubrir grandes áreas, alcanzando distancias de varios kilómetros en entornos urbanos o rurales, dependiendo de las condiciones de propagación. Además, se caracterizan por ofrecer bajas velocidades de transmisión, bajo consumo de energía, bajos costos de implementación y la capacidad de conectar miles de dispositivos a través de una misma red [12].

Actualmente, existen diversas tecnologías dentro de este ámbito, entre las que destacan LoRaWAN, Sigfox y NB-IoT. LoRaWAN se caracteriza por utilizar una tecnología abierta que opera en bandas de frecuencia sin licencia, permitiendo implementar redes públicas o privadas con gran flexibilidad. Por su parte, Sigfox utiliza una infraestructura gestionada por operadores y está orientada principalmente a la transmisión de pequeños mensajes con bajas velocidades de datos.

En cuanto a NB-IoT, opera sobre la infraestructura de redes celulares existente, proporcionando una amplia cobertura y confiabilidad; sin embargo, su disponibilidad depende de los servicios ofrecidos por los operadores de telefonía [13].

2.2.3 Tecnología LoRa

LoRa (Long Range) es una tecnología de comunicación inalámbrica desarrollada para la transmisión de datos a largas distancias con un bajo consumo energético. Su funcionamiento utiliza una técnica de modulación denominada Chirp Spread Spectrum (CSS), la cual proporciona una alta sensibilidad de recepción y una mayor resistencia frente al ruido y las interferencias presentes en el canal de comunicación [14].

Esta técnica permite establecer comunicaciones inalámbricas entre dispositivos IoT ubicados a varios kilómetros de distancia, dependiendo de las condiciones del entorno y de los parámetros de configuración. Debido a estas características, LoRa resulta adecuada para aplicaciones que requieren transmitir pequeñas cantidades de datos de manera periódica, como el monitoreo ambiental, la agricultura inteligente, la automatización industrial y las ciudades inteligentes.

2.2.4 Protocolo LoRaWAN

LoRaWAN (Long Range Wide Area Network) es un protocolo de comunicación diseñado para gestionar redes LPWAN que utilizan la tecnología LoRa como medio físico de transmisión. Este protocolo establece la forma en que los dispositivos se comunican a través de la red, definiendo reglas relacionadas con la autenticación, el enrutamiento, el acceso y la gestión de los datos transmitidos [15].

LoRaWAN permite conectar nodos sensores con un servidor de red mediante una puerta de enlace (Gateway), facilitando la transmisión de datos hacia plataformas de monitoreo y servicios en la nube.

Bajo este esquema, es posible gestionar una gran cantidad de dispositivos distribuidos geográficamente, manteniendo un bajo consumo energético y un amplio alcance de comunicación.

Frecuencias de operación

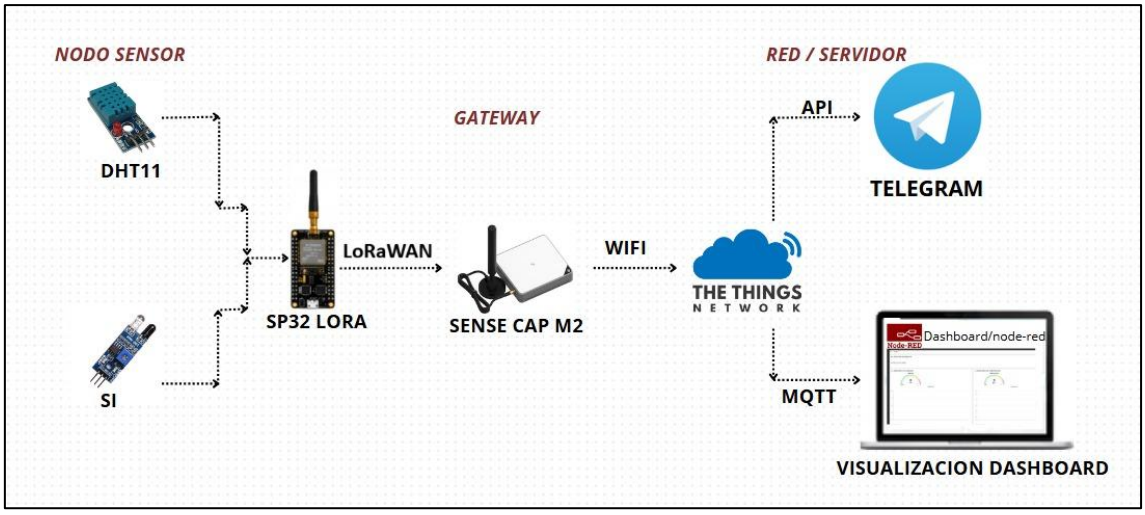
LoRaWAN opera en bandas de frecuencia sin licencia, cuyo uso se determina de acuerdo con las regulaciones establecidas en cada región. En el continente americano se utiliza principalmente la banda US915, mientras que en Europa es habitual la banda EU868. Estas bandas permiten implementar redes sin la necesidad de adquirir licencias de espectro, lo que facilita el desarrollo de aplicaciones IoT de bajo costo.

En este estudio se utiliza la banda US915, correspondiente a la configuración utilizada por el Gateway y los nodos LoRa en el entorno de prueba.

2.3 Arquitectura general del sistema IoT

Escenario experimental de la investigación basada en fases para la correcta transmisión de datos en un sistema IoT.

Figura 1. *Arquitectura general del sistema.*



Nota. Elaboración propia.

En la Figura 1 se ilustra el diseño de la red IoT que utiliza la tecnología LoRaWAN, diseñada para monitorear parámetros ambientales y detectar la presencia de objetos. La red está configurada para operar en la frecuencia de 915 MHz (US915) y comprende las etapas de recolección, transmisión, procesamiento y visualización de los datos generados por los sensores.

El procesamiento de la información en el sistema se organiza en tres niveles principales:

Nivel de recolección (nodo sensor): Este nivel está compuesto por un nodo que utiliza un ESP32 LoRa, conectado a un sensor DHT11 para medir la temperatura y la

humedad, además de un sensor infrarrojo (IR) para detectar la presencia de objetos. El microcontrolador procesa los datos obtenidos de ambos sensores y los transmite mediante el protocolo LoRaWAN.

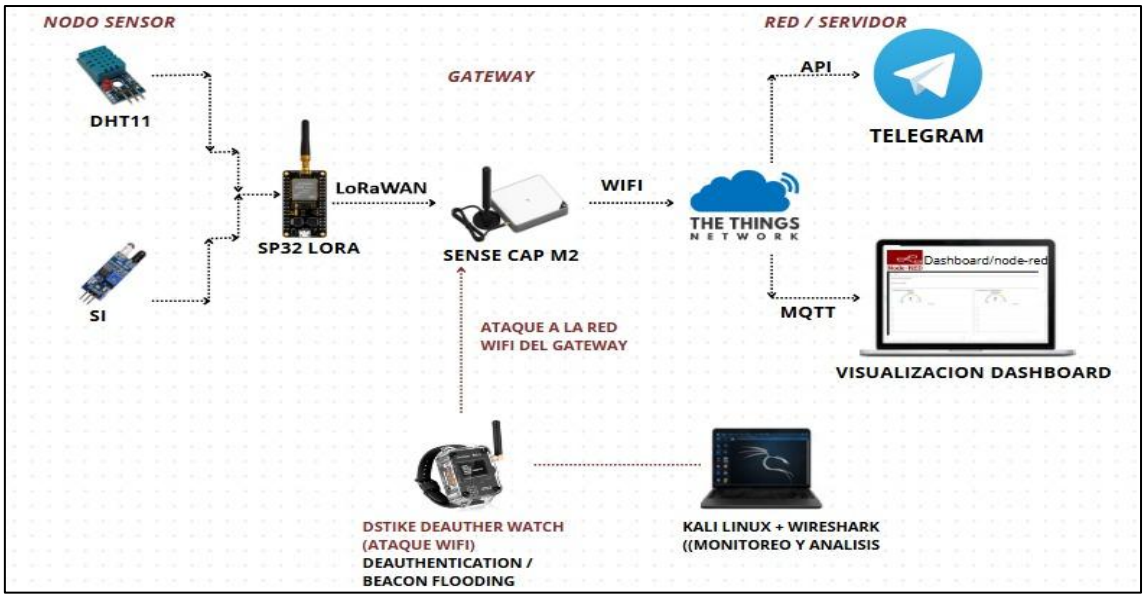
Nivel de comunicación (Gateway y servidor de red): Los datos transmitidos mediante LoRaWAN son recibidos por el Gateway SenseCAP M2, que actúa como interfaz entre la red LoRaWAN y la infraestructura de Internet. Posteriormente, el Gateway utiliza una conexión Wi-Fi para acceder a Internet y establecer la comunicación con The Things Network (TTN), donde se reciben y gestionan los datos provenientes del nodo LoRaWAN.

Nivel de aplicación (visualización y alertas): Una vez recibidos y gestionados en The Things Network, los datos se transfieren mediante el protocolo MQTT a Node-RED, donde la telemetría se presenta en un panel en tiempo real. Además, mediante la integración con la API de Telegram, el sistema envía alertas automáticas cuando los parámetros monitoreados superan los límites establecidos, lo que permite informar al usuario de manera oportuna.

2.4 Arquitectura de la red frente a escenarios de ataque

Escenario experimental de la investigación basada en fases para la evaluación ante la ejecución de ataques en un entorno controlado y análisis de seguridad.

Figura 2. *Arquitectura de escenario de ataques.*



Nota. Elaboración propia.

En la Figura 2 se ilustra el entorno experimental creado para analizar la respuesta de la red IoT frente a ataques de desautenticación e inundación de balizas (beacon flooding) que afectan la red Wi-Fi utilizada por el Gateway. El objetivo de estos ataques es evaluar su impacto en la conexión entre el Gateway y The Things Network (TTN), así como en la transmisión de datos de telemetría hacia las plataformas de supervisión. Este entorno experimental se organiza en las siguientes fases:

Ejecución de ataques: Los ataques se ejecutan utilizando el dispositivo Aursinc Wi-Fi Deauther que tiene la capacidad de generar tramas de desautenticación y múltiples tramas de beacon dirigidas a la red inalámbrica utilizada por el Gateway SenseCAP M2.

Ataque de desautenticación: Consiste en el envío de tramas de desautenticación falsificadas con el propósito de ocasionar una desconexión temporal del Gateway de la red Wi-Fi, interrumpiendo su comunicación con The Things Network.

Ataque de inundación de beacon: Consiste en generar múltiples redes Wi-Fi falsas que simulan redes legítimas, lo que dificulta la identificación de la red auténtica por parte de los dispositivos cercanos.

Vigilancia y análisis del tráfico: Durante la ejecución de ambos ataques, se utilizan Kali Linux y Wireshark para capturar y analizar el tráfico generado en la red inalámbrica. Estas herramientas permiten identificar las tramas relacionadas con cada ataque, verificar posibles pérdidas de conectividad y observar el comportamiento del Gateway antes, durante y después de cada escenario de ataque.

Análisis del impacto: Como consecuencia de los ataques realizados, la comunicación entre el Gateway y The Things Network puede verse afectada, provocando demoras o interrupciones temporales en la transmisión de datos de telemetría. Este impacto se manifiesta tanto en la actualización del dashboard de Node-RED como en el envío puntual de alertas generadas a través de Telegram, lo que permite evaluar el efecto real de los ataques en la disponibilidad y fiabilidad del sistema.

2.5 Sistema de Ataques

2.5.1 Ataque de desautenticación

El ataque de desautenticación consiste en el envío de tramas de administración falsificadas, cuyo propósito es provocar que uno o varios dispositivos se desconecten de una red Wi-Fi. Este tipo de ataque explota una vulnerabilidad presente en determinadas implementaciones del estándar IEEE 802.11, en las que las tramas de administración no cuentan con mecanismos de protección suficientes, permitiendo a un atacante suplantar la identidad del punto de acceso y provocar desconexión [16].

Al ejecutarse este ataque, el dispositivo atacante envía repetidamente tramas de tipo Deauthentication, haciendo que el dispositivo objetivo interprete estas tramas como una solicitud válida de desconexión del punto de acceso. Como resultado, el dispositivo pierde temporalmente la conexión y debe realizar nuevamente el proceso de autenticación para restablecerla.

En esta investigación, el ataque se dirigió a la red Wi-Fi utilizada por el Gateway SenseCAP M2, con el objetivo de provocar una pérdida temporal de comunicación entre el Gateway y The Things Network (TTN). De esta manera, se analizó el impacto de esta interrupción sobre la transmisión de telemetría y el envío de alertas generadas por el sistema.

Funcionamiento del ataque

- Identificación de la red inalámbrica objetivo.
- Obtención de la dirección MAC del punto de acceso y del dispositivo conectado.
- Desconexión momentánea del dispositivo afectado.
- Interrupción de la comunicación hasta que el dispositivo restablezca la conexión.

Consecuencias

- Interrupción temporal de la comunicación inalámbrica.
- Pérdida de conexión entre el gateway y la infraestructura de red.
- Retraso en la transmisión de telemetría.
- Interrupción momentánea en el envío de alertas y en la actualización del sistema de monitoreo.
- Disminución en la disponibilidad general del servicio.

2.5.2 Ataque Beacon Flooding

El ataque Beacon Flooding se caracteriza por la generación masiva de tramas Beacon falsificadas, con el objetivo de crear una gran cantidad de puntos de acceso Wi-

Fi falsos. Aunque estas redes pueden parecer auténticas, no proporcionan una conexión real a Internet ni servicios de comunicación legítimos.

La finalidad principal de este ataque es saturar el entorno inalámbrico mediante anuncios de redes ficticias, lo que puede dificultar la identificación del punto de acceso legítimo por parte de los dispositivos y aumentar la carga en el entorno inalámbrico [17].

En el desarrollo de este estudio, se aplicó este ataque para analizar cómo la presencia de múltiples redes ficticias puede influir en la estabilidad de la conexión entre el Gateway y la infraestructura de red.

Funcionamiento del ataque

- Creación automática de numerosas redes Wi-Fi ficticias.
- Transmisión constante de tramas Beacon.
- Distribución de nombres de red (SSID) que simulan redes Wi-Fi legítimas.
- Incremento del tráfico de gestión en el canal inalámbrico.
- Colapso del entorno de comunicación disponible.

Consecuencias

- Aumento de numerosas redes Wi-Fi falsas.
- Incremento notable del tráfico de gestión en el canal inalámbrico.
- Dificultad para distinguir la red real de las ficticias.
- Posibles retrasos o interrupciones en la comunicación inalámbrica.
- Disminución de la disponibilidad de la red mientras se lleve a cabo el ataque.

2.6 Kali Linux

Kali Linux es una distribución de Linux diseñada específicamente para tareas relacionadas con la ciberseguridad, el análisis forense y la identificación de vulnerabilidades en redes y sistemas informáticos. Incluye una amplia variedad de herramientas orientadas a la investigación, supervisión y evaluación de la seguridad en entornos digitales [18].

Entre sus principales características se encuentran la capacidad de realizar auditorías de redes inalámbricas, analizar el tráfico, ejecutar pruebas de penetración, identificar vulnerabilidades y supervisar las comunicaciones. Estas funcionalidades han permitido que Kali Linux se consolide como una herramienta ampliamente utilizada en el campo de la ciberseguridad.

En este estudio, se empleó Kali Linux para observar y analizar la actividad de la red durante los ataques de desautenticación y Beacon Flooding, lo que permitió verificar cómo estos eventos afectan la conexión entre el Gateway y la infraestructura de comunicación utilizada.

2.7 Wireshark

Wireshark es una herramienta de análisis de protocolos de red que permite capturar y examinar los paquetes que circulan por una red. Su funcionamiento se basa en la captura del tráfico mediante una interfaz de red, proporcionando información detallada sobre los paquetes, los protocolos utilizados, las direcciones de origen y destino, los puertos y otros parámetros asociados a la comunicación [19].

La herramienta permite analizar el contenido y las características de los paquetes capturados mediante diferentes filtros de visualización, facilitando la identificación de determinados protocolos, tipos de tráfico y eventos presentes durante una comunicación.

Además, proporciona información temporal de los paquetes, lo que permite estudiar la secuencia de los intercambios de datos y las variaciones producidas en el tráfico de red.

2.8 Protocolo MQTT

Message Queuing Telemetry Transport (MQTT) es un protocolo de mensajería ligero basado en el modelo de publicación y suscripción, orientado principalmente a aplicaciones relacionadas con el Internet de las Cosas (IoT) y dispositivos con recursos limitados. Su característica principal es facilitar la transmisión eficiente de datos mediante un bajo consumo de ancho de banda y mecanismos que permiten gestionar la entrega de mensajes entre los dispositivos y las aplicaciones involucradas.

El funcionamiento de MQTT comprende tres elementos principales: el bróker, encargado de gestionar y distribuir los mensajes entre los clientes; los publicadores, responsables de enviar la información a un tópico determinado; y los suscriptores, encargados de recibir los mensajes publicados en los tópicos a los que se encuentran suscritos. Esta estructura facilita el intercambio de información entre múltiples dispositivos y aplicaciones.

En el desarrollo de esta investigación, MQTT se empleó para permitir que Node-RED recibiera la telemetría publicada desde The Things Network (TTN). A partir de la recepción de estos datos, Node-RED realizó su procesamiento y permitió su visualización en el dashboard, además de generar alertas mediante Telegram cuando las variables monitoreadas superaban los límites establecidos.

2.9 API de Telegram

La API de Telegram es una interfaz que permite integrar servicios externos con la plataforma Telegram mediante el uso de bots. A través de esta API, es posible automatizar

el envío de mensajes, imágenes, archivos y notificaciones, facilitando la comunicación entre diferentes aplicaciones y sus usuarios.

En el contexto de las aplicaciones IoT, esta API puede utilizarse para generar alertas cuando ocurren determinados eventos o cuando las variables monitoreadas superan los límites establecidos. Su integración con plataformas como Node-RED permite automatizar el envío de estas notificaciones sin necesidad de intervención directa del usuario.

En el desarrollo de esta investigación, la API de Telegram se utilizó para enviar las alertas automáticas generadas desde Node-RED, informar sobre el estado de las variables monitoreadas y notificar cualquier irregularidad detectada durante el funcionamiento del sistema.

CAPÍTULO III

3.1 Diseño e implementación

La red IoT basada en tecnología LoRaWAN se diseña e implementa mediante la integración de los componentes de adquisición, comunicación y gestión de datos, considerando su interconexión y configuración dentro de la infraestructura propuesta. Asimismo, se incorporan los elementos requeridos para la ejecución de las pruebas de seguridad y el registro de los eventos generados durante su desarrollo.

La infraestructura está conformada por dispositivos sensores para la adquisición de variables ambientales, nodos finales para la transmisión mediante LoRa, una puerta de enlace (Gateway) como punto de interconexión entre la red LoRaWAN y la red IP, y plataformas destinadas a la recepción, procesamiento y visualización de la telemetría. La conexión Wi-Fi proporciona al Gateway el acceso a Internet necesario para el intercambio de información con los servicios de red y las plataformas externas.

La arquitectura aprovecha las características de LoRaWAN, particularmente su bajo consumo energético y capacidad de comunicación a larga distancia, para establecer el enlace entre los nodos y la puerta de enlace. Sobre la infraestructura Wi-Fi se configuran los escenarios de ataques, cuyos efectos se registran mediante la captura y análisis del tráfico generado durante las pruebas.

La selección de los componentes responde a criterios de funcionalidad, compatibilidad y características técnicas, de acuerdo con los requerimientos de comunicación, transmisión de telemetría y evaluación de seguridad establecidos para la infraestructura.

3.2 Equipos y materiales utilizados

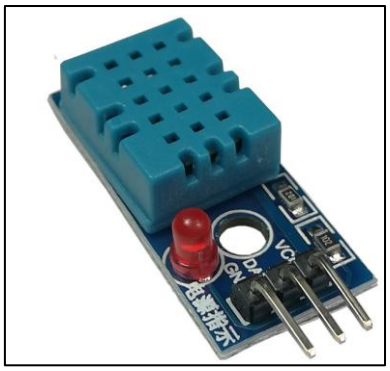
La implementación del sistema se enfoca en el uso de dispositivos específicos que permiten la construcción de una red IoT funcional.

3.2.1 Componentes de hardware

Sensor DHT11

En esta sección se presenta el sensor de temperatura y humedad , utilizado para la adquisición de las variables ambientales del sistema implementado. Asimismo, se describen sus especificaciones técnicas, la conexión realizada con el nodo ESP32 LoRa y los parámetros de funcionamiento considerados durante el desarrollo de la investigación [20].

Figura 3. *Sensor DHT11*



Nota. Imagen referencial del Sensor DHT11. Según [20].

En la Figura 3 se ilustra el sensor DHT11 utilizado para la medición de la temperatura y la humedad ambiental. Este dispositivo permite obtener datos que posteriormente son transmitidos mediante la red LoRaWAN para su análisis.

Tabla 1. *Especificaciones del sensor DHT11.*

Parámetro	Especificación
Variable monitoreada	Temperatura y humedad
Voltaje de operación	3.3 V
Pin de conexión	GPIO 4
Tipo de sensor	Digital
Frecuencia de envío desde el nodo	15 s
Estado normal de temperatura	Menor a 30 °C
Estado medio de temperatura	30 °C a 34.9 °C
Umbral de alerta de temperatura alta	≥ 35 °C
Estado normal de humedad	Menor a 60 %
Estado medio de humedad	60 % a 79.9 %
Umbral de alerta de humedad alta	≥ 80 %
Plataforma donde se generan alertas	Node-RED / Telegram
Tecnología de transmisión asociada	LoRaWAN 915 MHz

Nota. Elaboración propia.

En la Tabla 1 se presentan los parámetros que permiten configurar el monitoreo de temperatura y humedad dentro del nodo IoT. El voltaje de operación de 3,3 V garantiza la compatibilidad con el ESP32 LoRa, mientras que la conexión mediante el GPIO 4

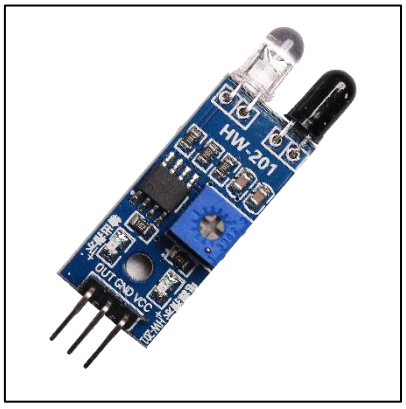
facilita la adquisición de datos ambientales. La frecuencia de muestreo de 15 segundos permite obtener información periódica del entorno sin generar una carga excesiva de procesamiento.

Los umbrales de temperatura y humedad fueron definidos para identificar condiciones que requieran la activación de alertas, mientras que los valores de recuperación permiten mantener la estabilidad del sistema, evitando activaciones y desactivaciones continuas.

Sensor IR

En esta sección se presentan las principales especificaciones técnicas del sensor infrarrojo (SI) empleado en el sistema IoT. Se describen la variable monitoreada, las especificaciones de conexión con el nodo ESP32 LoRa, el tipo de salida digital y los parámetros de funcionamiento considerados para la detección de presencia y la transmisión de la información mediante la red LoRaWAN. [21].

Figura 4. *Sensor SI.*



Nota. Imagen referencial del Sensor SI. Según [21].

En la Figura 4 se muestra el sensor infrarrojo utilizado en el nodo IoT para la detección de objetos mediante la emisión y recepción de radiación infrarroja.

La información obtenida es enviada al sistema para su posterior transmisión a través de la red LoRaWAN.

Tabla 1. *Especificaciones del sensor SI.*

Parámetro	Especificación
Variable monitoreada	Presencia de objetos
Voltaje de operación	3.3 V
Pin de conexión	GPIO 7
Tipo de salida	Digital
Estado de detección	LOW / 0
Estado normal	HIGH / 1
Frecuencia de envío desde el nodo	15 s
Alerta generada	Cuando sensor_ir = 0
Plataforma donde se genera la alerta	Node-RED / Telegram
Tecnología de transmisión asociada	LoRaWAN 915 MHz

Nota. Elaboración propia.

La Tabla 2 presenta las especificaciones del hardware utilizado para la detección de objetos o movimientos dentro del área monitoreada. El funcionamiento a 3,3 V asegura la compatibilidad eléctrica con el ESP32 LoRa, mientras que la utilización del GPIO 7 permite recibir la señal digital generada por el sensor.

Los estados lógicos HIGH y LOW facilitan la identificación de eventos de presencia, permitiendo que la información sea procesada y utilizada por el sistema para la generación de alertas o el envío de datos mediante LoRa.

Tabla 2. *Conexión de sensores y actuador al ESP32 LoRa.*

Dispositivo	VCC	GND	Señal

DHT11	**3.3 V**	**GND**	**GPIO 4**
Sensor IR	**3.3 V**	**GND**	**GPIO 7**

Nota. Elaboración propia.

La Tabla 3 presenta los datos de conexión de los sensores al ESP32 LoRa, lo que permite la adquisición de datos ambientales, la detección de presencia y la generación de alertas sonoras dentro del sistema. La alimentación a 3,3 V garantiza la compatibilidad entre los dispositivos, mientras que la asignación de pines independientes facilita su funcionamiento simultáneo y la transmisión de la información mediante tecnología LoRa.

Gateway SenseCAP M2

En esta sección se presentan las principales características técnicas del Gateway SenseCAP M2, dispositivo encargado de recibir los datos transmitidos por los nodos LoRaWAN y reenviarlos al servidor de red mediante una conexión a Internet. Asimismo, se describen sus parámetros de funcionamiento y las características consideradas para su implementación dentro de la arquitectura IoT desarrollada en esta investigación [22].

Figura 5. *Equipo SenseCAP M2.*

Nota. Imagen referencial del equipo SenseCAP M2. Según [22].

La figura 5 se ilustra el Gateway SenseCAP M2 utilizado para la recepción y gestión de los datos enviados por los nodos IoT durante la implementación experimental de la red LPWAN.

Tabla 3. *Especificaciones del SenseCAP M2.*

Parámetro	Especificación
Función	Gateway LoRaWAN
Frecuencia de operación	915 MHz
Conectividad	Ethernet / Wi-Fi
Protocolo	LoRaWAN
Rol en la red	Recepción y reenvío de datos
Plataforma de gestión	ChirpStack

Nota. Elaboración propia.

La Tabla 4 presenta las especificaciones del Gateway SenseCAP M2, el cual permite establecer la comunicación entre los nodos LoRa y la plataforma TTN mediante una conexión Wi-Fi. La operación en la banda de 915 MHz garantiza la compatibilidad con los dispositivos de la red, mientras que su función de recepción y reenvío de paquetes facilita la transferencia de información hacia la plataforma de gestión para su monitoreo y procesamiento.

Nodo LoRa ESP32

En esta sección se presentan las principales especificaciones técnicas del nodo LoRa ESP32 implementado en la investigación. Este dispositivo es el encargado de adquirir la información proveniente de los sensores [23].

Figura 6. *ESP32 Nodo LoRa.*

Nota. Imagen referencial del nodo LoRa basado en ESP32. Según [23].

La Figura 6 muestra el nodo LoRa ESP32 Heltec utilizado en la implementación experimental para la adquisición y transmisión de datos en la red IoT.

Tabla 4. *Especificaciones nodo LoRa.*

Parámetro	**Descripción**
Dispositivo	ESP32 LoRa V2
Microcontrolador	ESP32
Módulo LoRa	SX1276
Frecuencia de operación	915 MHz
Alimentación	Batería Li-ion de 3,7 V
Comunicación inalámbrica	LoRa
Sensores conectados	DHT11 y sensor IR
Actuador conectado	Buzzer
Función en el sistema	Adquisición, procesamiento y transmisión de datos

Nota. Elaboración propia.

En la Tabla 5 se presentan las especificaciones técnicas del dispositivo. El ESP32 LoRa constituye la unidad principal de procesamiento y comunicación del sistema IoT propuesto. Su integración con el módulo SX1276 permite la transmisión inalámbrica de información a larga distancia, mientras que la alimentación mediante una batería de 3,7 V proporciona autonomía energética para el funcionamiento del nodo. Además, el dispositivo permite gestionar los datos provenientes de los sensores y controlar la generación de alertas dentro de la red IoT implementada.

El nodo suplantador fue implementado utilizando un ESP32 LoRa con el propósito de simular un dispositivo malicioso dentro de la red IoT.

Dispositivo generador de tráfico adverso

Se emplea el dispositivo Aursinc Wi-Fi Deauther para la simulación de condiciones adversas en la red [24].

Figura 7. *Deauther Watch V.*

Nota. Imagen Referencial del equipo Deauther Watch. Según [24].

La Figura 7 muestra el dispositivo Deauther Watch V3 basada en ESP8266, utilizado como herramienta experimental para generar escenarios de ataque inalámbricos, como desautenticación (Deauthentication) e inundación de balizas (Beacon Flooding).

Tabla 5. *Especificaciones de deauther v3*

Parámetro	Descripción
Dispositivo	Aursinc Wi-Fi Deauther
Microcontrolador	ESP8266
Frecuencia de operación	2.4 GHz
Conectividad	Wi-Fi IEEE 802.11 b/g/n
Alimentación	Batería recargable integrada
Función en el sistema	Herramienta de evaluación de seguridad
Pruebas realizadas	Desautenticación y Beacon Flooding
Entorno de uso	Red IoT controlada

Nota. Elaboración propia.

En la Tabla 6 se presentan las especificaciones técnicas del dispositivo Aursinc Wi-Fi Deauther, incorporado para la ejecución de pruebas de seguridad sobre la red inalámbrica del sistema IoT.

Sus capacidades de generación y análisis de tramas Wi-Fi permiten simular escenarios de ataque controlados, facilitando la evaluación del comportamiento de la red frente a amenazas que afectan la disponibilidad y el rendimiento de las comunicaciones.

3.2.2 Software y herramientas utilizadas

Arduino IDE

Arduino IDE es un entorno de desarrollo integrado empleado para la creación, compilación y carga del firmware en microcontroladores como el ESP32 LoRa y el ESP8266. Este software proporciona un entorno simplificado para la programación en lenguaje C/C++, facilitando la implementación de funciones orientadas a la adquisición de datos, control de periféricos y la comunicación inalámbrica [25].

Dentro del sistema IoT propuesto, Arduino IDE permite la integración de librerías específicas para sensores como el DHT11 y módulos de comunicación LoRa, así como la

configuración de pines digitales y analógicos del microcontrolador. Además, incorpora herramientas de depuración mediante el monitor serial, lo que facilita el análisis del comportamiento del sistema durante su ejecución.

Figura 8. *Logotipo de Arduino ID.*

Nota. Representación gráfica del logotipo de Arduino IDE. Según [25].

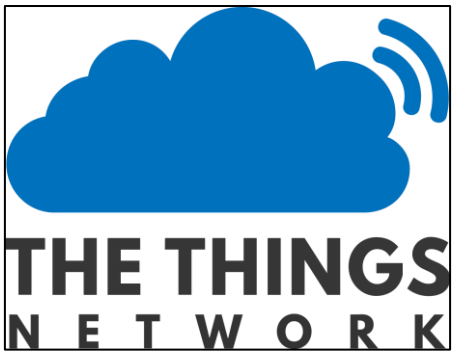
La figura 8 muestra el Software Arduino IDE utilizado para la programación del nodo LoRa ESP32 para la correcta implementación de la comunicación inalámbrica LoRaWAN.

The Things Network (TTN)

The Things Network (TTN) es una plataforma de red LoRaWAN utilizada para gestionar de manera centralizada dispositivos IoT y controlar el flujo de datos proveniente de redes de amplia cobertura. Esta plataforma permite registrar y administrar nodos, autenticar dispositivos mediante claves criptográficas y gestionar el enrutamiento de los paquetes hacia las aplicaciones finales. [26].

En el sistema implementado, TTN actúa como la capa de red que recibe los datos enviados desde el Gateway SenseCAP M2, facilitando su decodificación y posterior visualización en una interfaz de usuario. Además, ofrece la posibilidad de monitorear el estado de los dispositivos y supervisar el flujo de datos en la red IoT.

Figura 9. *Logotipo de la Plataforma the Things Network (TTN).*



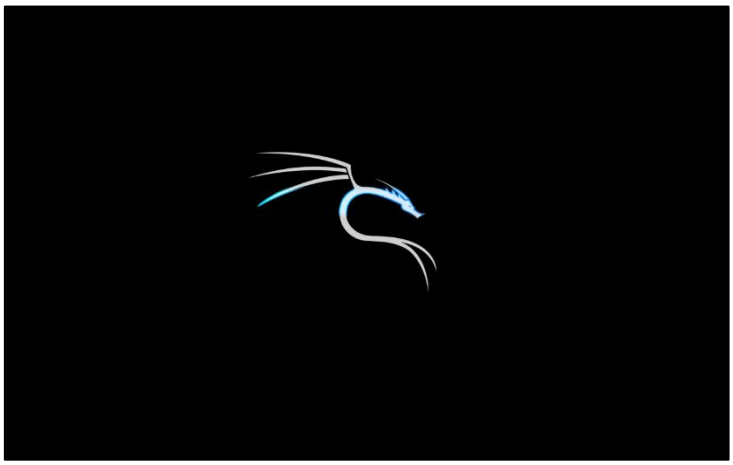
Nota. Imagen referencial de (TTN). Según [26].

La Figura 9 presenta la plataforma The Things Network, utilizada como servidor de red LoRaWAN, donde se configuraron los parámetros de autenticación y comunicación del nodo IoT.

Kali Linux

Durante el desarrollo de la investigación, se empleó Kali Linux para realizar pruebas de seguridad, como ataques de desautenticación y generación de tráfico malicioso, con el objetivo de evaluar la resistencia de la red LoRa y explorar posibles áreas de vulnerabilidad en la estructura de comunicación [27].

Figura 10. *Acceso a Kali Linux.*



Nota. Imagen referencial Kali Linux. Según [27].

La Figura 10 presenta la interfaz de Kali Linux utilizada como herramienta de apoyo en los escenarios experimentales de seguridad.

Wireshark

Wireshark se empleó como herramienta de verificación para evaluar la transmisión de datos y detectar posibles irregularidades en el tráfico, contribuyendo al análisis del rendimiento y la seguridad del sistema implementado [28].

Figura 11. *Logotipo de la herramienta Wireshark.*

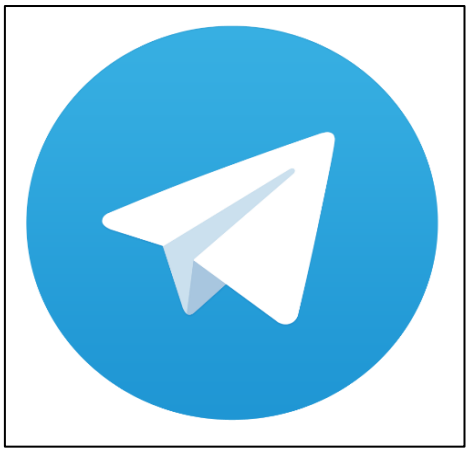
Nota. Logotipo de Wireshark. Según [28].

En la Figura 11 se ilustra la herramienta Wireshark utilizada para analizar el comportamiento del tráfico de red durante las pruebas de evaluación de seguridad.

Telegram para notificaciones en tiempo real

En el sistema IoT propuesto, la integración con servicios externos se realiza con el propósito de habilitar el almacenamiento, la visualización y la notificación en tiempo real de los datos generados por los nodos de monitoreo. Para ello, se emplean la plataforma The Things Network, ThingSpeak, el protocolo MQTT y la API de Telegram, los cuales, en conjunto, permiten extender las capacidades del sistema hacia un entorno de monitoreo basado en la nube.

Figura 12. *Logotipo de Telegram.*



Nota. Logotipo de telegram, Según [28].

La Figura 12 presenta la aplicación Telegram empleada como sistema de visualización y alerta remota.

De manera complementaria, se implementó la API de Telegram como un sistema de notificación inmediata. Este servicio permite el envío automático de alertas a un usuario o grupo previamente configurado cuando el sistema detecta condiciones anómalas, como un aumento de temperatura por encima de un umbral definido o la activación del sensor de presencia.

Las notificaciones enviadas incluyen información relevante sobre el estado del sistema, permitiendo una supervisión remota eficiente y una respuesta rápida ante eventos críticos. Esta integración mejora significativamente la capacidad de monitoreo del sistema IoT, al incorporar un canal de comunicación directo con el usuario final.

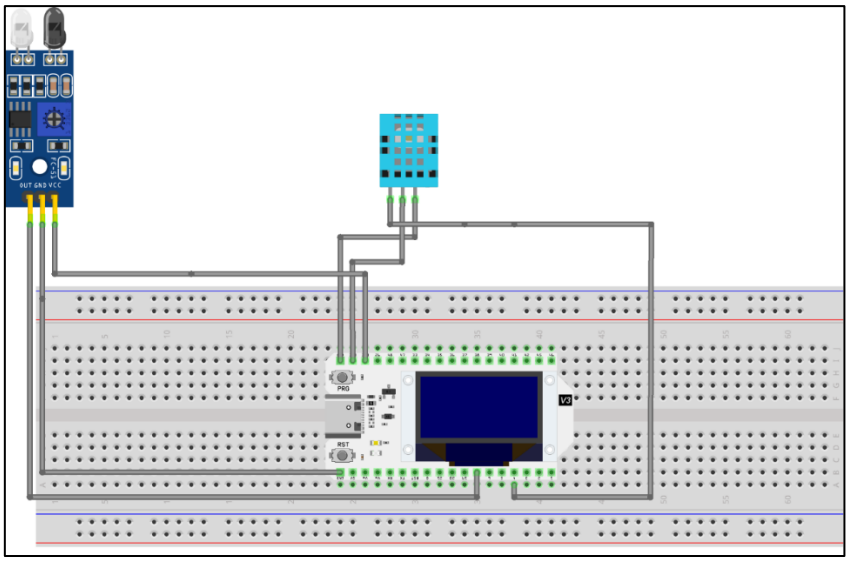
3.3 Diseño de conexión

En esta sección se presenta el proceso de diseño y validación del nodo IoT implementado mediante tres representaciones complementarias: el diagrama de conexiones en protoboard, el esquema eléctrico y el diseño final de la placa PCB. Estas etapas permitieron verificar la integración de los sensores con la placa Heltec Wi-Fi LoRa

V3, definir la asignación de pines y establecer una estructura de montaje estable para el prototipo final.

3.3.1 Diagrama de conexiones

Figura 13. *Diagrama en Protoboard.*



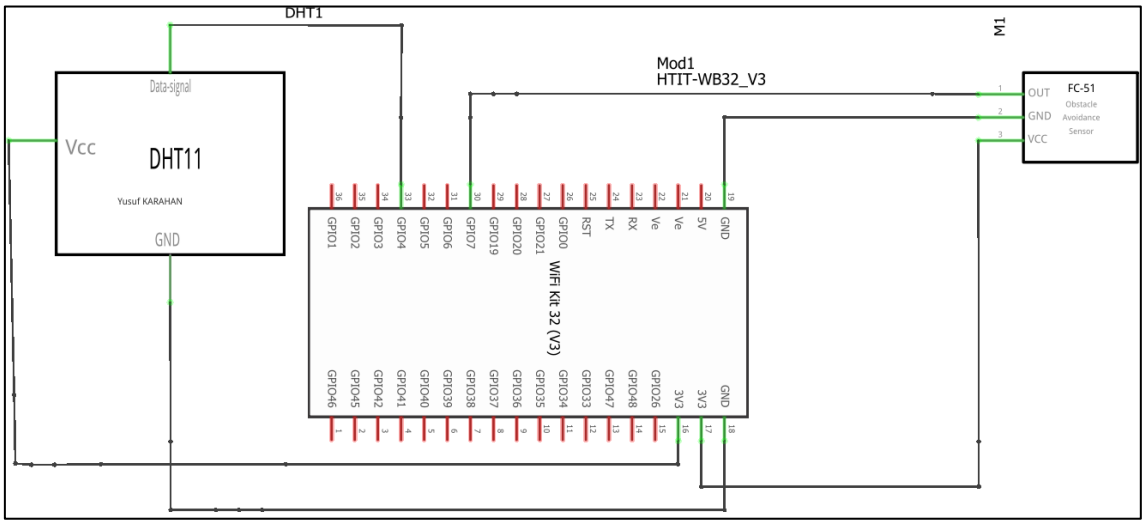
Nota. Elaboración Propia.

En la Figura 13 se presenta la disposición física de los componentes sobre una protoboard, correspondiente a una etapa previa al montaje definitivo del nodo IoT. En esta configuración se integran el sensor infrarrojo (IR), el sensor de temperatura y humedad DHT11 y la pantalla OLED con la placa Heltec WiFi LoRa 32 V3. El sensor IR dispone de las conexiones de alimentación VCC, tierra GND y señal de salida OUT, mientras que el DHT11 incorpora las líneas de alimentación y señal conectadas a los pines asignados de la placa.

La pantalla OLED utiliza el bus de comunicación I2C para el intercambio de datos con el microcontrolador, por lo que su conexión se establece mediante las líneas correspondientes de alimentación y comunicación. La distribución de los conductores sobre la protoboard permite verificar la asignación de los componentes y sus respectivas conexiones antes del ensamblaje definitivo del prototipo.

3.3.2 Diagrama esquemático

Figura 14. *Diagrama Esquemático.*



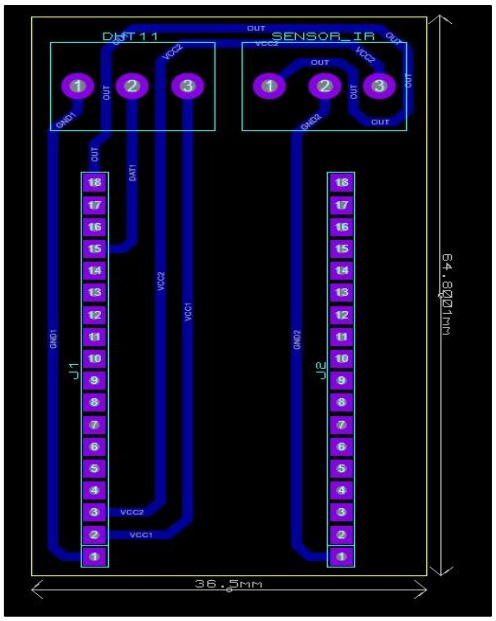
Nota. Elaboración Propia.

En la Figura 14 se muestra el diagrama correspondiente a la representación esquemática del sistema, en el cual se detalla la asignación específica de los pines GPIO utilizados por cada dispositivo conectado al módulo Heltec LoRa 32 V3 Node. El sensor DHT11 se vincula mediante su línea de datos, además de sus terminales VCC y GND, mientras que el sensor infrarrojo (SI) se conecta a través de sus tres terminales: OUT, GND y VCC.

A diferencia de la vista de protoboard, este esquema no representa la ubicación física de los componentes, sino las conexiones eléctricas entre ellos, lo que facilitó la identificación exacta del GPIO correspondiente a cada sensor para su posterior programación en Arduino IDE.

3.3.3 Diseño PCB

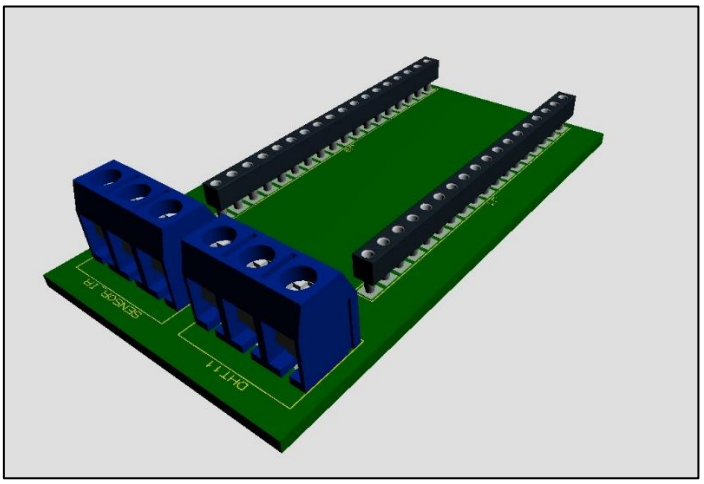
Figura 15. *Diagrama PCB.*



Nota. Elaboración Propia.

La Figura 15 corresponde al diseño de la placa de circuito impreso (PCB), generado a partir del esquema eléctrico previamente definido. En esta representación se observa el ruteado de las pistas que conectan cada componente sobre la superficie de la placa, sustituyendo los cables sueltos utilizados en la protoboard por conexiones soldadas de manera permanente.

Figura 16. *Modelo tridimensional (3D).*



Nota. Elaboración Propia.

En la Figura 16 se muestra un modelo tridimensional (3D) cuyo propósito es organizar la distribución espacial de los elementos, optimizar el espacio disponible dentro de la carcasa y reducir el riesgo de desconexiones que suelen presentarse en montajes temporales sobre protoboard. De esta manera, se aporta mayor estabilidad eléctrica y mecánica al prototipo final.

3.4 Diseño técnico de la propuesta

Para mejorar la organización, protección y estabilidad de los componentes eléctricos del sistema IoT, se propuso utilizar una cubierta fabricada mediante técnicas de impresión 3D como solución para encapsular el prototipo. Este tipo de diseño es común en dispositivos integrados e IoT debido a su flexibilidad en cuanto a formas, bajo costo y facilidad de personalización.

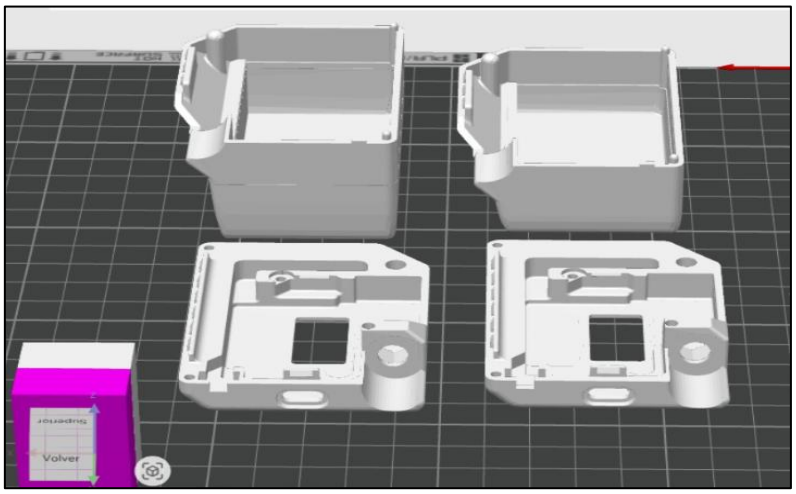
La elección de este modelo respondió a la necesidad de organizar de manera adecuada el nodo compuesto por el ESP32 LoRa, la pantalla OLED, el sistema de alimentación y los elementos de control del prototipo. La estructura se adaptó a las dimensiones de los módulos, asegurando la compatibilidad física y una correcta disposición interna de los componentes.

El diseño incluyó un espacio específico para la batería de 3,7 V, lo que facilitó su sujeción y conexión segura al sistema de alimentación del nodo. También se incorporaron aberturas para el interruptor de encendido y apagado, permitiendo controlar el dispositivo sin necesidad de abrir la carcasa. Además, se diseñaron aperturas para la ubicación de los sensores, permitiendo su exposición al entorno exterior y facilitando la medición de temperatura, humedad y detección de presencia. Asimismo, se consideraron orificios para la ventilación y el acceso a los puertos de programación, lo que simplificó el mantenimiento y la supervisión del sistema.

El objetivo de esta implementación fue proporcionar un entorno físico más estable y eficiente para el prototipo IoT, reduciendo los riesgos de desconexiones, daños mecánicos y exposición de los circuitos eléctricos. Esto permitió evaluar el sistema en condiciones más cercanas a un entorno real de uso, aumentando su confiabilidad durante las pruebas de funcionamiento y el análisis de su comportamiento.

3.4.1 Vista superior

Figura 17. *Vista Superior 3D.*

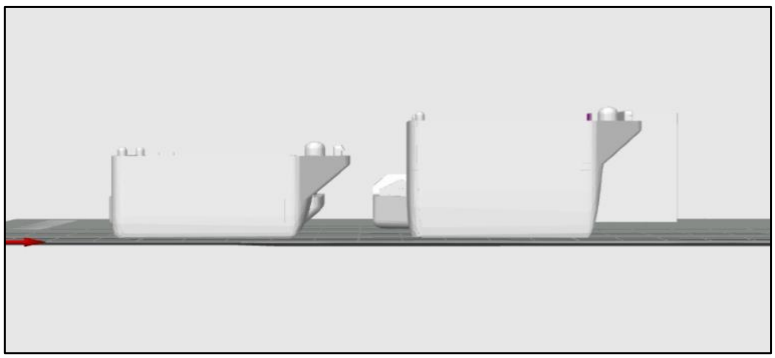


Nota. Elaboración Propia.

En la figura 17 se aprecia la organización interna del prototipo, incluyendo el espacio destinado para el ESP32 LoRa, la batería y el cableado del sistema.

3.4.2 Vista lateral

Figura 18. *Vista Lateral en 3D.*

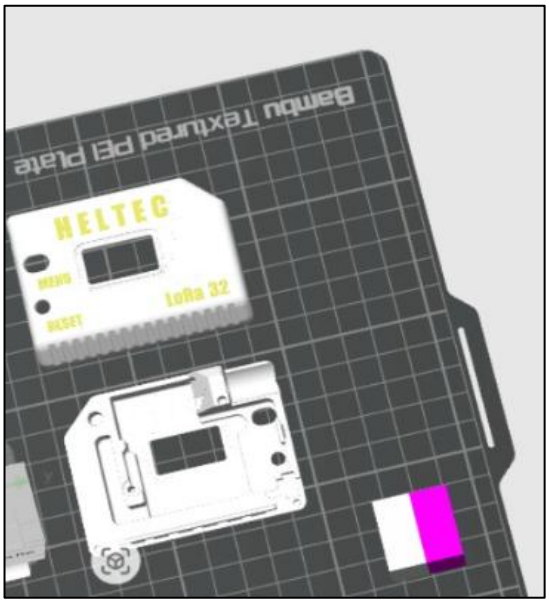


Nota. Elaboración Propia.

En la Figura 18 se observa el perfil lateral del diseño de la carcasa, en el cual se evidencian dos compartimentos de diferentes dimensiones. Uno de ellos presenta aberturas destinadas al paso de cables, sensores y elementos de conexión, mientras que el otro está destinado al alojamiento del ESP32 LoRa y su respectiva batería.

3.4.3 Vista frontal

Figura 19. *Vista Frontal.*



Nota. Elaboración Propia.

En la figura 19 se visualiza el panel frontal del diseño, donde se ubican elementos de interacción como la pantalla OLED y las aberturas para sensores y control del dispositivo.

3.5 Entorno de pruebas de seguridad

El entorno de pruebas de seguridad fue definido con el propósito de evaluar el comportamiento de la red inalámbrica del sistema IoT bajo condiciones controladas. Este entorno, denominado “Red-Lab”, corresponde a una red local configurada en modo hotspot, utilizada exclusivamente para la realización de pruebas de conectividad, estabilidad y análisis de seguridad.

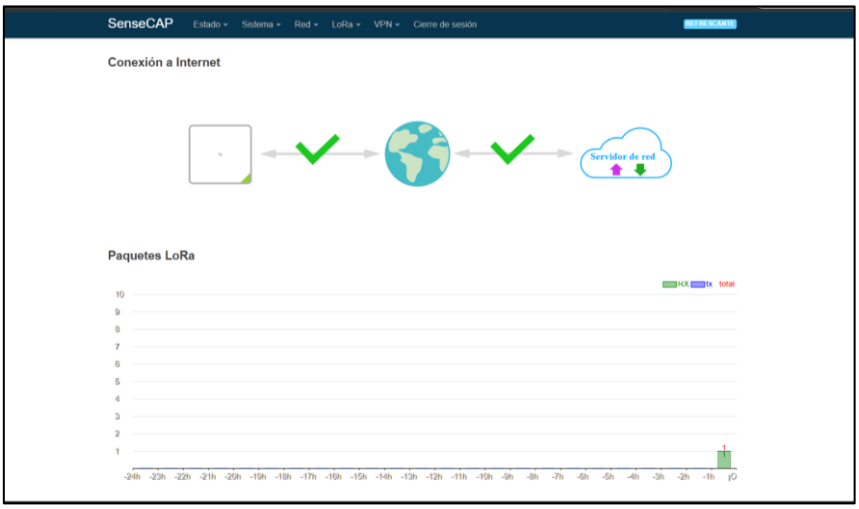
El entorno permitió ejecutar pruebas sin afectar redes externas ni infraestructuras en producción, garantizando que los resultados obtenidos correspondieran únicamente al sistema implementado. De esta manera, se estableció un ambiente controlado para la evaluación del comportamiento de los dispositivos IoT frente a variaciones en la disponibilidad del servicio y posibles escenarios de interferencia en la red inalámbrica.

Asimismo, este entorno permitió la integración de herramientas de análisis y monitoreo de red, facilitando la observación del tráfico y el estudio del desempeño de la comunicación entre los dispositivos del sistema IoT y la red inalámbrica utilizada como medio de transmisión.

3.6 Configuración del Gateway SenseCAP M2

En esta sección se presenta el acceso a la interfaz de administración del gateway SenseCAP M2, desde donde se realizan las configuraciones necesarias para su funcionamiento.

Figura 20. *Acceso a la interfaz web del Gateway.*



Nota. Elaboración Propia.

En la Figura 20 se muestra la interfaz principal de administración del Gateway SenseCAP M2, utilizada para verificar el estado de la conexión a Internet y supervisar el funcionamiento general del sistema LoRaWAN. Mediante esta interfaz, accesible a través de la dirección IP 192.168.100.222, fue posible acceder a las configuraciones de red necesarias para el entorno experimental de la tesis.

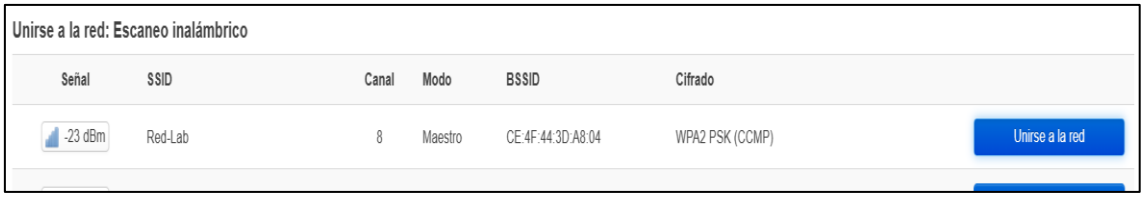
Usuario: Admin

Contraseña: KCXTa9pY

3.6.1 Escaneo de la red

En esta sección se describe el proceso de búsqueda de redes inalámbricas disponibles realizado desde el Gateway SenseCAP M2.

Figura 21. *Escaneo de red Wi-Fi.*



Nota. Elaboración Propia.

En la figura 21 se presenta el proceso de escaneo de redes inalámbricas disponibles realizado desde el Gateway SenseCAP M2. Se identificó la red “Red-Lab”, correspondiente al hotspot utilizado en el entorno de pruebas, permitiendo establecer la conectividad inalámbrica del Gateway hacia Internet.

Para esto se debe tener en cuenta datos de ingreso de la red:

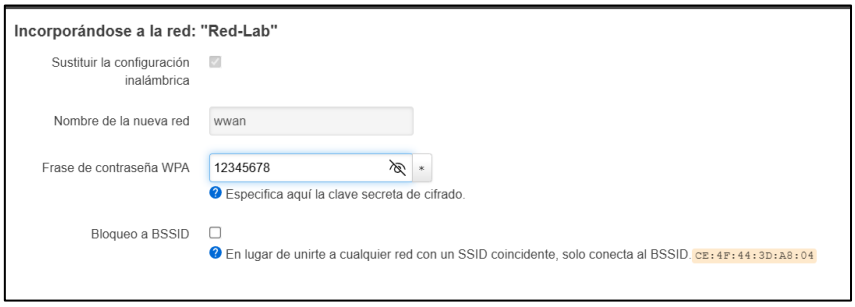
Usuario: Red-Lab

Contraseña:12345678

3.6.2 Enlazamiento con la red WI-FI

En esta sección se presenta el proceso de conexión del Gateway SenseCAP M2 a la red Wi-Fi seleccionada.

Figura 22. *Enlazamiento con la red.*



Nota. Elaboración Propia.

En la figura 22 se observa el estado de la conexión inalámbrica del Gateway SenseCAP M2, por el cual se enlazó correctamente a la red Wi-Fi denominada **"Red-Lab"**, funcionando en modo cliente para permitir el acceso a Internet y la comunicación con la plataforma The Things Network (TTN). Asimismo, se observa el uso del estándar de seguridad **WPA2-PSK**, empleado para proteger la autenticación y el intercambio de datos dentro de la red inalámbrica.

3.6.3 Conexión inalámbrica

En esta sección se verifica el estado de la conexión inalámbrica establecida por el gateway. Asimismo, se presentan los indicadores de intensidad de señal y otros parámetros que permiten comprobar la estabilidad de la comunicación con la red Wi-Fi.

Figura 23. *Confirmación de conexión inalámbrica.*

Nota. Elaboración Propia.

En la Figura 23 se verificó una intensidad de señal aproximada de **−45 dBm**, valor que indica una recepción de señal adecuada para mantener una comunicación estable entre el Gateway y la infraestructura de red.

Esta condición resulta favorable para reducir las interrupciones de conectividad y asegurar la continuidad en la transmisión de datos provenientes de los nodos IoT implementados.

3.6.4 Verificación de la conectividad

Dentro de la red local utilizada durante el entorno de pruebas, se empleó un comando de exploración mediante solicitudes **ICMP (ping)**, aplicado sobre un rango de direcciones IP, con el objetivo de identificar dispositivos activos y comprobar la disponibilidad de comunicación en la red.

Figura 24. *Ip dinámica.*

```
C:\Users\jamilex>
C:\Users\jamilex>for /L %i in (2,1,254) do @ping -n 1 -w 20 10.16.169.%i | f
ind "TTL="
Respuesta desde 10.16.169.21: bytes=32 tiempo=4ms TTL=64
Respuesta desde 10.16.169.78: bytes=32 tiempo<1m TTL=128
Respuesta desde 10.16.169.160: bytes=32 tiempo=55ms TTL=64

C:\Users\jamilex>
```

Nota. Elaboración Propia.

En la Figura 24 se aprecian respuestas provenientes de varias direcciones IP, en las que se evidencian los tiempos de respuesta y los valores **TTL (Time To Live)**

asociados a los dispositivos detectados. Esta verificación permitió confirmar la presencia de equipos conectados y la operatividad de la red antes de continuar con las pruebas de comunicación y la evaluación del sistema IoT implementado.

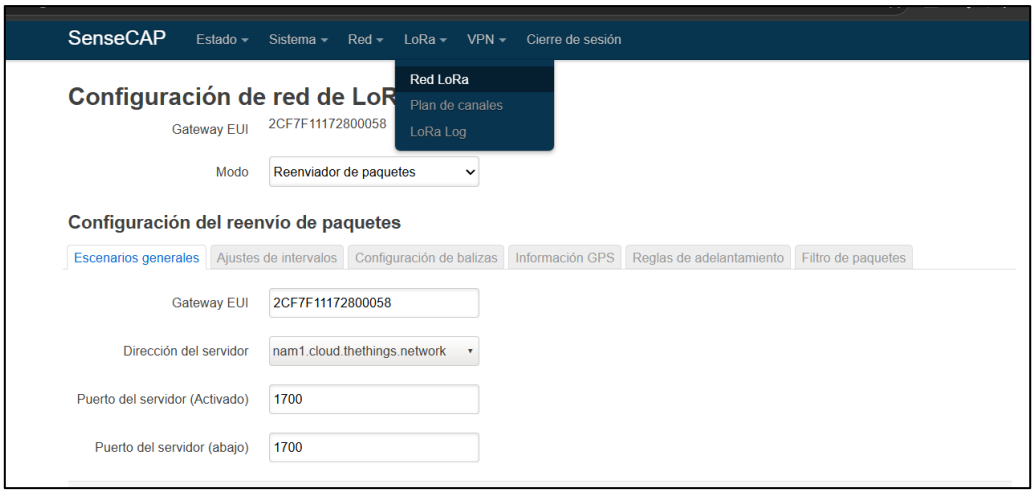
Las direcciones IP identificadas dentro del entorno experimental son:

- La IP del hotspot (10.16.169.21).
- La IP de tu laptop (10.16.169.78).
- La IP del SenseCAP (10.16.169.160).

3.6.5 Configuración de red LoRaWAN del Gateway

La configuración del Gateway SenseCAP M2 dentro de la arquitectura LoRaWAN, se emplea el modo de reenvío de paquetes (Packet Forwarder) para establecer la comunicación con The Things Network (TTN).

Figura 25. *Configuración de red LoRaWAN.*



Nota. Elaboración Propia.

En la Figura 25 se muestra la configuración del sistema de pruebas desarrollado bajo la banda de frecuencia **US915**, utilizada en el entorno experimental en Ecuador. Se definió el servidor **nam1.cloud. thethings.network** junto con el puerto **1700**, lo que

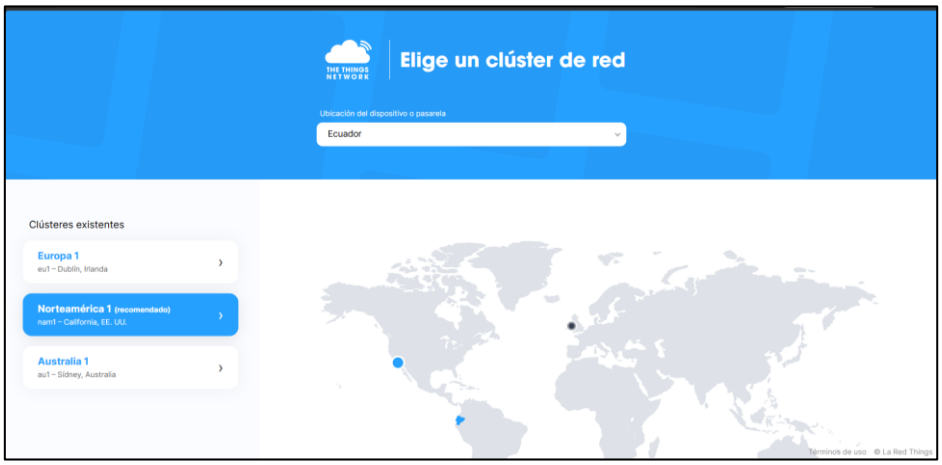
permitió la transferencia de paquetes entre el Gateway y los dispositivos IoT implementados.

3.7 Acceso en la plataforma The Things Network

Durante esta fase se llevó a cabo la selección del clúster de red asociado a la ubicación del proyecto, tomando en consideración la región geográfica de implementación y la cercanía de los servidores disponibles, con el propósito de favorecer una comunicación más estable y eficiente.

Para el desarrollo de este proyecto se utilizó el clúster **North America 1 (nam1 – California, Estados Unidos)**, sugerido por la plataforma para dispositivos implementados en Ecuador. Esta elección permitió establecer la comunicación inicial entre el Gateway SenseCAP M2, los nodos LoRaWAN y la plataforma The Things Network (TTN), posibilitando posteriormente el registro, la autenticación y la administración de los dispositivos dentro de la red IoT implementada.

Figura 26. *Acceso a la plataforma TTN.*



Nota. Elaboración Propia.

En la Figura 26 se aprecian los parámetros de identificación correspondientes al SenseCAP M2, entre ellos el Gateway ID, el Gateway EUI y el plan de frecuencias

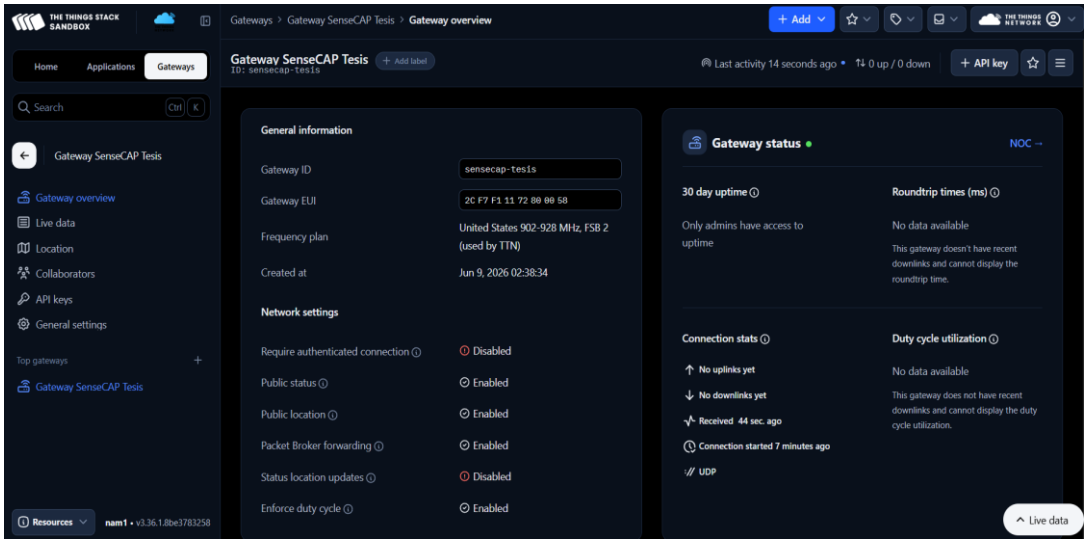
configurado, utilizado para asegurar la compatibilidad con los nodos IoT implementados dentro de la red LoRaWAN.

Asimismo, la plataforma evidenció actividad reciente del Gateway, reflejada mediante indicadores asociados al estado de conexión, la recepción de información y la disponibilidad operativa. Estos parámetros permitieron confirmar que el Gateway se encontraba correctamente vinculado a TTN, quedando habilitado para la recepción y retransmisión de datos provenientes de los nodos LoRaWAN integrados en el sistema IoT desarrollado.

3.7.1 Conexión estable del Gateway con TTN

Esta configuración favoreció una comunicación eficiente y de baja carga de procesamiento, facilitando la transferencia continua de información desde los nodos IoT hacia la infraestructura en la nube.

Figura 27. *Conexión con TTN.*



Nota. Elaboración Propia.

En la Figura 27 se identifica el uso del protocolo UDP (User Datagram Protocol) en la comunicación entre la puerta de enlace (Gateway) y el servidor de red. Este

protocolo permite el transporte de los mensajes recibidos por la puerta de enlace hacia la infraestructura de red, mediante el intercambio de datagramas sin establecer una conexión previa.

Gateway ID: SenseCAP-tesis

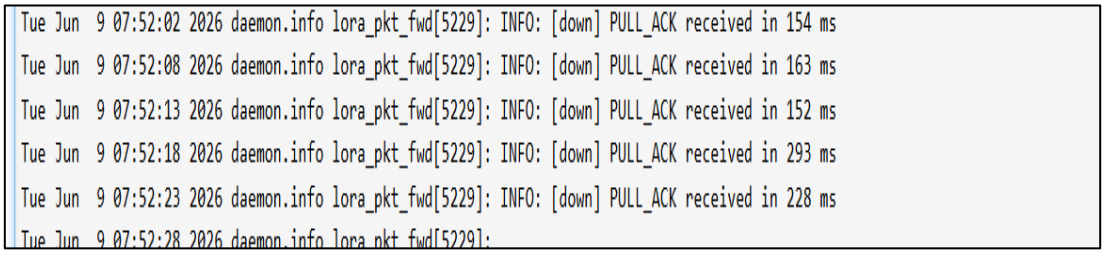
Gateway EUI:2C F7 F1 11 72 80 00 58

Frequency plan: United States 902-928 MHz, FSB 2(used by TTN)

3.7.2 Monitoreo del estado de comunicación interno del gateway

Durante el proceso de integración de la red LoRaWAN, se realizó el monitoreo desde la sección **Red Log** del Gateway, utilizada para supervisar el intercambio de información con la plataforma **The Things Network (TTN)**. En dicho apartado se registraron eventos asociados al funcionamiento del **packet forwarder**, encargado de reenviar los paquetes de datos desde el Gateway hacia los servidores de red.

Figura 28. *Registro de confirmación de conectividad.*



Nota. Elaboración Propia.

En la Figura 28 se identifican múltiples mensajes **PULL_ACK**, correspondientes a las respuestas enviadas por el servidor de red al Gateway en respuesta a las solicitudes **PULL_DATA**.

La presencia continua de estos mensajes permitió verificar que el Gateway mantenía una comunicación activa con la plataforma TTN, evidenciando el correcto

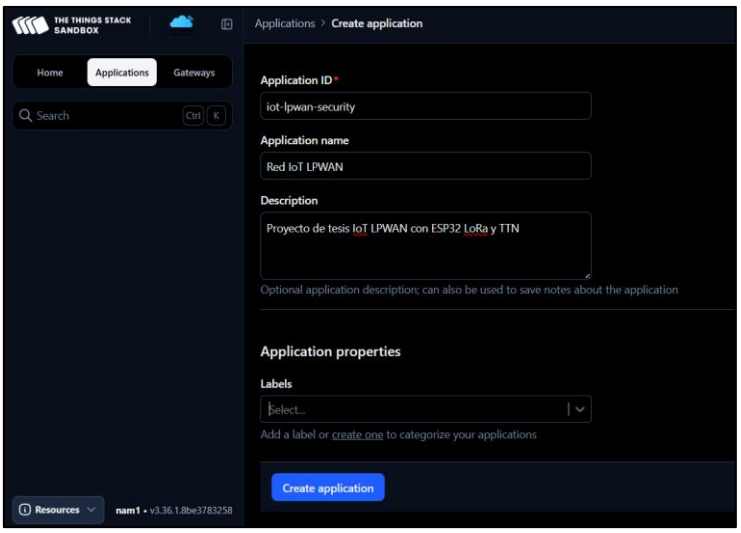
funcionamiento del mecanismo de intercambio de datos dentro de la infraestructura implementada.

Asimismo, los tiempos de respuesta expresados en milisegundos permitieron comprobar la estabilidad de la conexión establecida entre el Gateway y el servidor de red, reflejando un comportamiento adecuado durante el intercambio de información.

3.8 Configuración de los nodos en la aplicación TTN

En esta etapa se procedió con la creación de una aplicación dentro de la plataforma TTN, la cual permite administrar los dispositivos finales que forman parte de la red LoRaWAN.

Figura 29. *Configuración de nodos en TTN.*



Nota. Elaboración Propia.

En la Figura 29 se ilustra la asignación del identificador, el nombre y una descripción relacionada con el proyecto. Esta aplicación funciona como el espacio principal donde se gestionan los nodos, las credenciales, los datos recibidos y las integraciones posteriores del sistema IoT.

Se muestran los datos utilizados para la creación de la aplicación:

Application ID: iot-lpwan-security

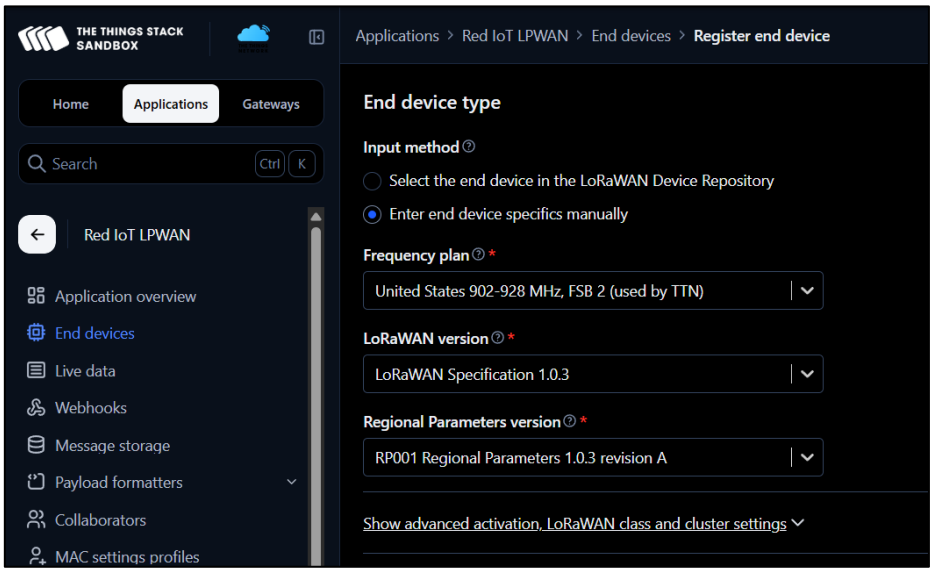
Application name: Red IoT LPWAN

Description: Proyecto de tesis iot LpWAN con ESP23 LoRa y TTN

3.8.1 Registro del nodo de forma manual

En esta sección se realizó el registro del nodo debido a que la placa utilizada corresponde a un módulo **Heltec LoRa ESP32 SX1276 868–915 MHz**.

Figura 30. *Registro de nodo.*



Nota. Elaboración Propia.

En la Figura 30 se observa el plan de frecuencias, compatible con la banda **US915** utilizada por el Gateway SenseCAP M2. Además, se configuraron la versión de LoRaWAN y los parámetros regionales, garantizando la compatibilidad entre el nodo, el Gateway y la plataforma TTN.

Frecuency plan: United States 902–928 MHz, FSB 2,

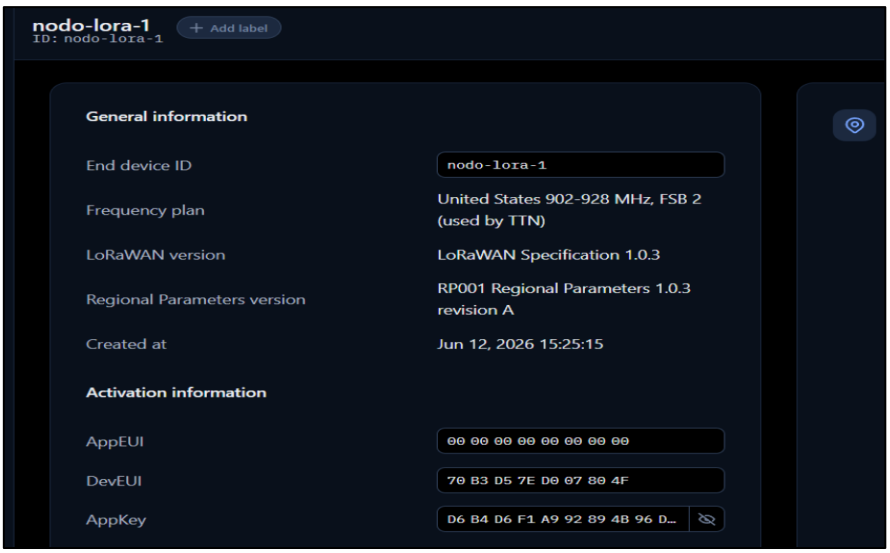
LoRaWAN version: Specification 1.0.3

Regional parameters version: RP001 Regional Parameters 1.0.3 revision A

3.8.2 Autenticación del nodo dentro de la red

Una vez definidos estos parámetros de comunicación, se generaron las credenciales necesarias como AppEUI, DevEUI y AppKey los cuales permiten identificar el dispositivo y establecer el proceso de activación mediante OTAA.

Figura 31. *Autenticación del nodo.*



Nota. Elaboración Propia.

En la Figura 31 se ilustran las credenciales necesarias para que el nodo pueda unirse a la red LoRaWAN y transmitir los datos capturados por los sensores hacia TTN.

Finalmente, se verificó que el nodo **nodo-LoRa-1** quedara registrado correctamente dentro de la aplicación **Red IoT LPWAN**. En la información general se observan el plan de frecuencias configurado, la versión de LoRaWAN, los parámetros regionales y las credenciales de activación. Con esta configuración, el nodo quedó preparado para ser programado desde Arduino IDE y posteriormente enviar datos de telemetría hacia el Gateway SenseCAP M2 y la plataforma TTN.

Se muestran los datos de activación considerados necesarios para establecer la comunicación entre el nodo, el Gateway y la plataforma TTN.

AppEUI :0000000000000000

DevEUI: 70B3D57ED007804F

AppKey: D6B4D6F1A992894B96D6A29A2C728E44

3.9 Configuración y uso de Kali Linux y Wireshark

Como parte del entorno de pruebas de seguridad de la red IoT, se llevó a cabo la configuración inicial del sistema operativo Kali Linux, empleado para la ejecución de herramientas destinadas al análisis y monitoreo de red. Esta distribución fue seleccionada debido a las funcionalidades integradas orientadas a las pruebas de seguridad informática y al análisis de tráfico, lo que permitió establecer un entorno adecuado para el desarrollo de las evaluaciones planteadas en la investigación.

Durante esta etapa se realizó la configuración básica del sistema, incluyendo la modificación de las credenciales de acceso y la personalización de los parámetros del usuario, con el fin de disponer de un entorno funcional para la ejecución de las pruebas. Posteriormente, se empleó Wireshark como herramienta de captura y análisis del tráfico de red, lo que permitió supervisar el comportamiento de la conectividad y observar los efectos generados durante los escenarios de evaluación implementados.

3.9.1 Acceso al sistema kali linux.

Una vez finalizada la instalación del sistema operativo, se realizó el ingreso inicial al entorno de Kali Linux mediante la autenticación del usuario previamente configurado.

Figura 32. *Acceso a Kali.*

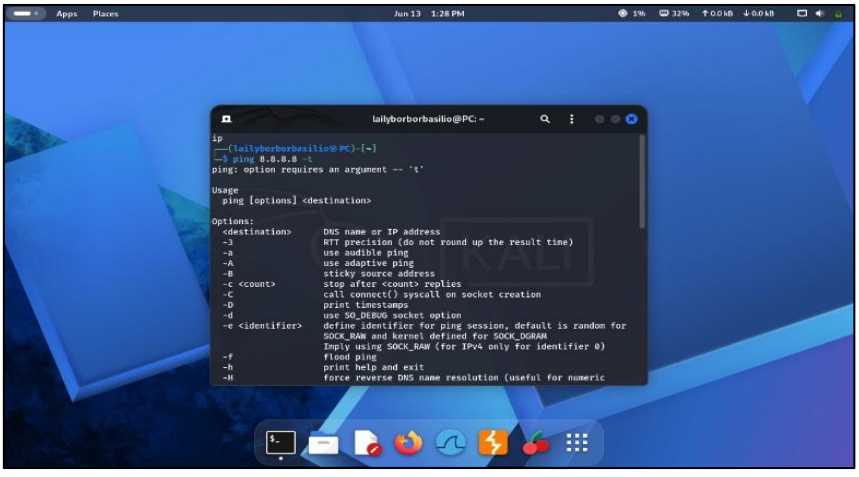
Nota. Elaboración Propia.

En la Figura 32 se verificó el correcto funcionamiento del sistema y la disponibilidad del entorno de trabajo para la ejecución de herramientas de análisis y monitoreo de red.

3.9.2 Verificación de conectividad de la red en kali Linux.

Con el propósito de comprobar la disponibilidad de conexión en el entorno virtualizado, se ejecutaron pruebas básicas de conectividad desde la terminal del sistema.

Figura 33. *Verificación de conectividad.*



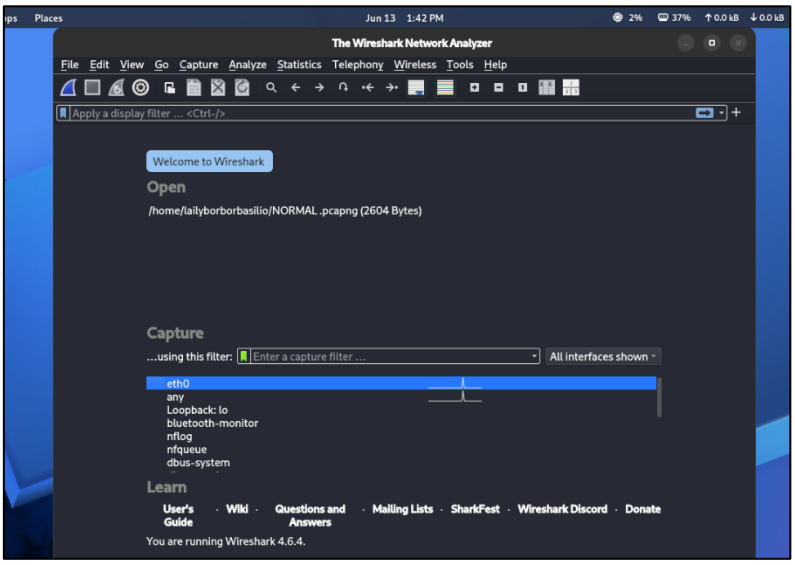
Nota. Elaboración Propia.

En la Figura 33 se observa la verificación de conectividad. Dicho procedimiento permitió comprobar el estado de la comunicación de red necesaria para el desarrollo de las pruebas de seguridad y el análisis de tráfico.

3.9.3 Apertura inicial de Wireshark.

Posteriormente, se realizó la ejecución de Wireshark como herramienta de captura y análisis de tráfico de red, seleccionando las interfaces disponibles para el monitoreo de paquetes.

Figura 34. *Inicio de Wireshark.*



Nota. Elaboración Propia.

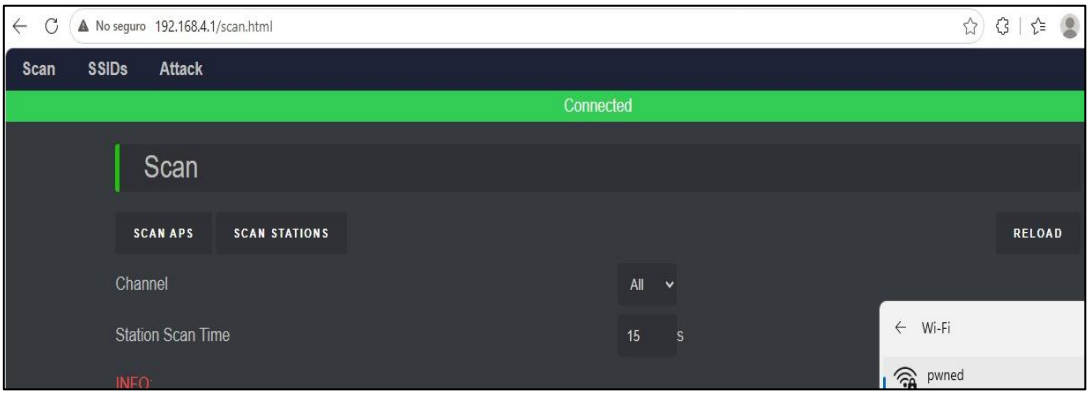
En la Figura 34 se muestra la ejecución de Wireshark, utilizado para la captura y análisis del tráfico de red durante las pruebas experimentales. La herramienta permite visualizar los paquetes intercambiados a través de la interfaz de comunicación y examinar los protocolos involucrados en las conexiones establecidas durante los escenarios de evaluación.

3.10 Acceso y configuración de la interfaz del reloj Deauther

Con el propósito de desarrollar las pruebas de seguridad sobre la red inalámbrica del entorno IoT, se realizó la puesta en funcionamiento del reloj Deauther Wi-Fi.

El dispositivo fue energizado mediante una conexión USB al puerto de la PC, lo que permitió el encendido del sistema y la habilitación de sus funciones internas orientadas al análisis y evaluación de redes inalámbricas.

Figura 35. *Ilustración de 1a Interfaz del reloj.*



Nota. Elaboración Propia.

En la figura 35 se observa el acceso a la interfaz de administración del dispositivo, se efectuó la conexión a la red inalámbrica generada por el reloj, identificada de forma predeterminada por el firmware Deauther.

Red del reloj:

Usuario: pwend

Contraseña: deauther

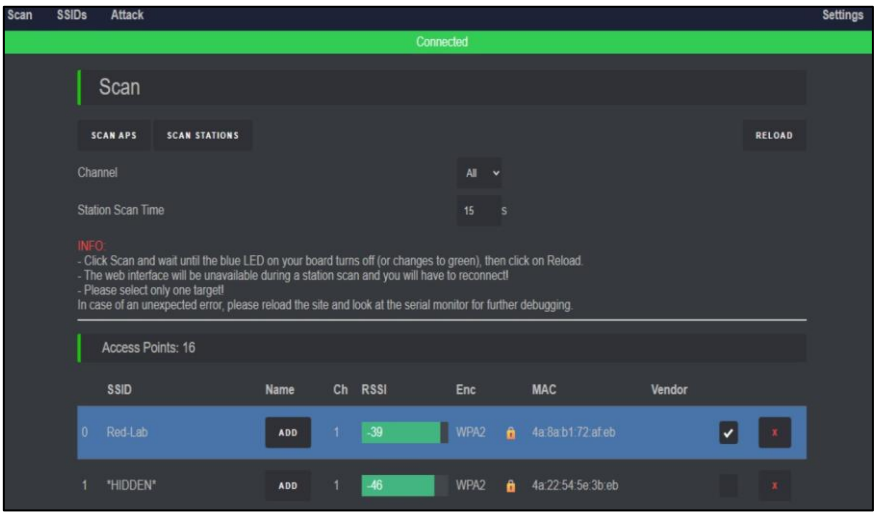
Una vez establecida la conexión, se ingresó a la interfaz web de configuración desde un navegador. La interfaz web del reloj Deauther, accesible mediante la dirección

IP **192.168.4.1**, permite administrar funciones relacionadas con el análisis de redes inalámbricas.

3.10.1 Escaneo de redes inalámbricas

Una vez establecido el acceso a la interfaz de administración del reloj Deauther, se procedió a utilizar la función de escaneo de las redes disponibles en el entorno de prueba. Mediante la opción **Scan APs**, el dispositivo realizó la detección de los puntos de acceso cercanos, permitiendo visualizar parámetros asociados a cada red, tales como el identificador de red (SSID), el canal de operación, la intensidad de señal (RSSI), el tipo de cifrado y la dirección MAC del punto de acceso detectado.

Figura 36. *Escaneo de red.*



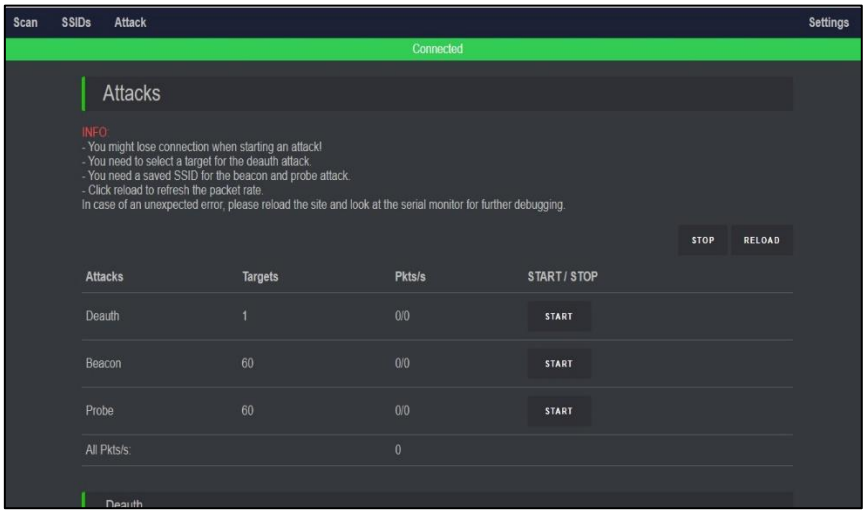
Nota. Elaboración Propia.

En la Figura 36 se observa el proceso de identificación de la red inalámbrica utilizada dentro del laboratorio de pruebas, denominada **Red-Lab**, la cual fue seleccionada como objetivo para el desarrollo de los escenarios de evaluación de la disponibilidad de la conectividad.

3.10.2 Monitoreo de sección de ataques

Posteriormente, se accedió a la sección de ataques disponible en la interfaz del reloj Deauther, donde se visualizaron las funcionalidades destinadas a la ejecución de escenarios de evaluación sobre redes inalámbricas. Dentro de esta interfaz se presentaron opciones relacionadas con ataques de desautenticación (Deauth), generación masiva de balizas falsas (Beacon) y envío de solicitudes de exploración (Probe), las cuales permiten analizar el comportamiento de la conectividad de una red frente a diferentes condiciones de interferencia.

Figura 37. *Visualización de ataques.*



Nota. Elaboración Propia.

En la Figura 37 se muestran las principales funciones asociadas a los ataques de desautenticación (Deauthentication) e inundación de balizas (Beacon Flooding) disponibles en el dispositivo Deauther. Estas opciones permiten seleccionar los puntos de acceso identificados y configurar la ejecución de los escenarios de prueba sobre la red inalámbrica. La utilización de estas funciones permitió evaluar las variaciones producidas en la conectividad del enlace Wi-Fi durante las pruebas experimentales.

3.11 Pruebas de conexión en Wireshark

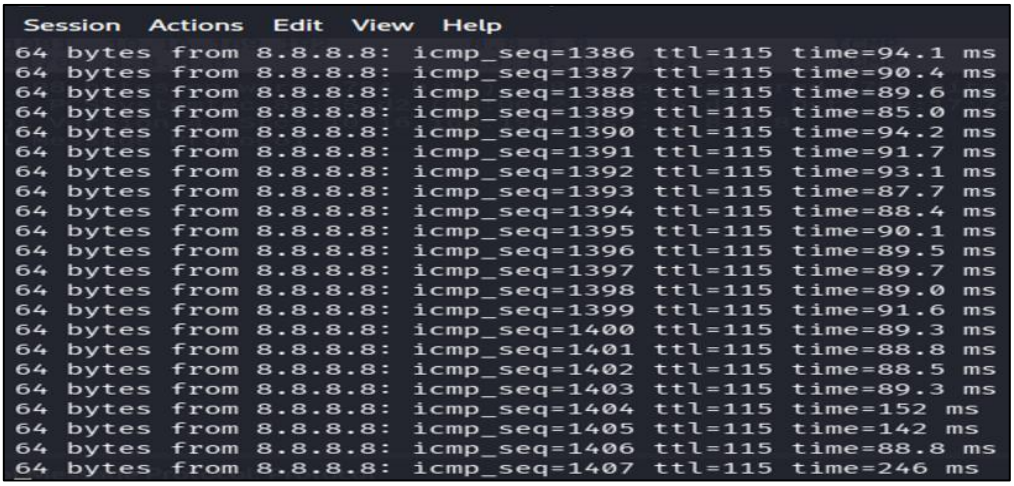
Con el fin de analizar el comportamiento de la red inalámbrica ante diferentes escenarios de ataque, se llevaron a cabo pruebas de conectividad empleando Wireshark como herramienta para la captura y el análisis del tráfico de red. Mediante el monitoreo de paquetes **ICMP** generados a través del comando **ping**, se evaluó el estado de la comunicación entre el dispositivo y la red antes, durante y después de la ejecución de los ataques.

Este análisis permitió identificar cambios en los tiempos de respuesta, pérdida de paquetes e interrupciones en la conectividad, evidenciando el impacto de los ataques sobre la disponibilidad del sistema IoT.

3.11.1 Ataque de Deauthentication

En esta sección. Se aprecian el comportamiento de la red antes y durante el ataque de deauthentication.

Figura 38. *Estado Normal sin ataque.*

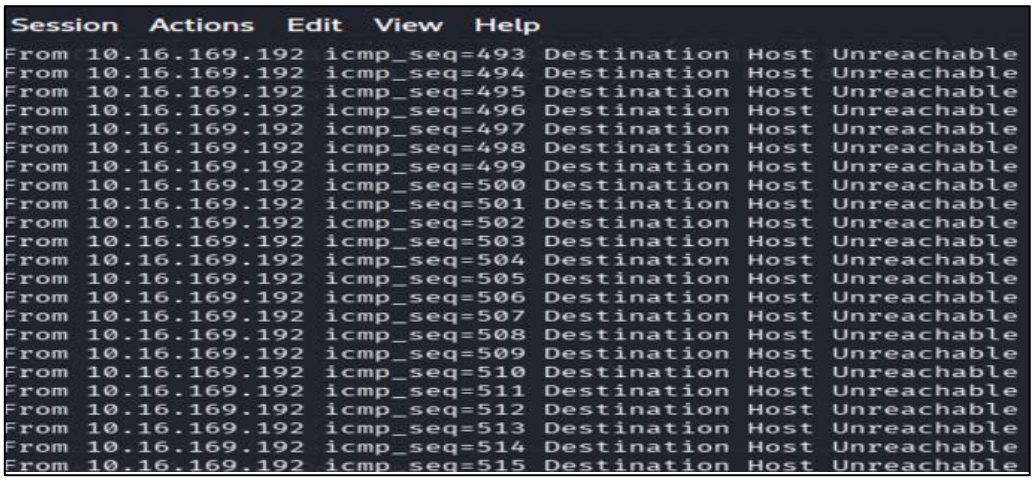


Nota. Elaboración Propia.

En la figura 38 se muestra la sección del comando ping hacia la dirección 8.8.8.8 en condiciones normales, sin ataque. Se observa que las respuestas se reciben de forma

consistente, con tiempos de respuesta estables, lo que evidencia que la conexión Wi-Fi funciona correctamente antes de iniciar el ataque.

Figura 39. *Estado Anormal durante ataque.*



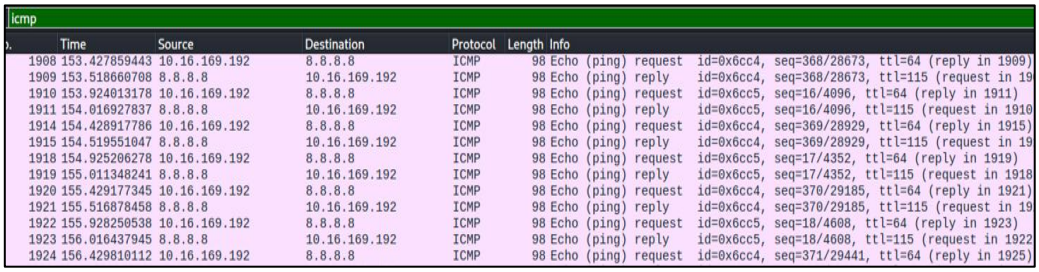
Nota. Elaboración Propia.

En la figura 39 se ilustra que, durante el ataque de desautenticación ejecutado con el dispositivo Deauther, se evidenció una pérdida total de conectividad, reflejada en el mensaje "Destination Host Unreachable". Esto indica que el dispositivo perdió por completo el acceso a la red, afectando la comunicación entre el host y el punto de acceso.

3.11.2 Ataque de Beacon Flooding

El ataque provocó una degradación de la conectividad Wi-Fi, generando interrupciones en la comunicación entre el host y el destino.

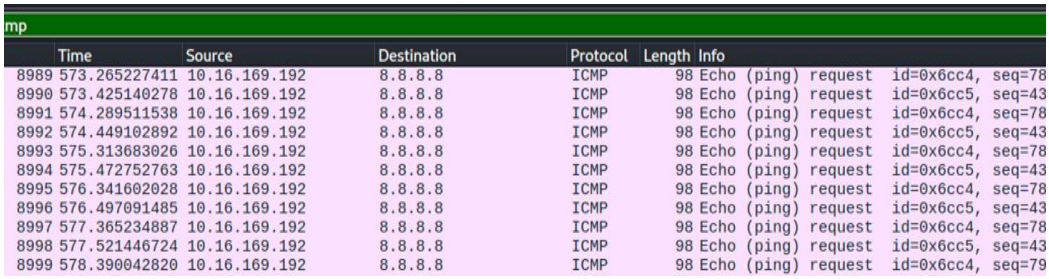
Figura 40. *Estado normal sin ataque de Beacon.*



Nota. Elaboración Propia.

En la figura 40 se ilustra el estado normal de la conexión antes del ataque de Beacon Flooding, evidenciado por el intercambio correcto de solicitudes y respuestas ICMP (echo request / echo reply) entre el host 10.16.169.192 y el destino 8.8.8.8 lo que confirma la disponibilidad normal de la red.

Figura 41. *Estado Anormal durante el ataque de Beacon.*



icmp

	Time	Source	Destination	Protocol	Length	Info
8989	573.265227411	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc4, seq=78
8990	573.425140278	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc5, seq=43
8991	574.289511538	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc4, seq=78
8992	574.449102892	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc5, seq=43
8993	575.313683026	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc4, seq=78
8994	575.472752763	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc5, seq=43
8995	576.341602028	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc4, seq=78
8996	576.497091485	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc5, seq=43
8997	577.365234887	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc4, seq=78
8998	577.521446724	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc5, seq=43
8999	578.390042820	10.16.169.192	8.8.8.8	ICMP	98	Echo (ping) request id=0x6cc4, seq=79

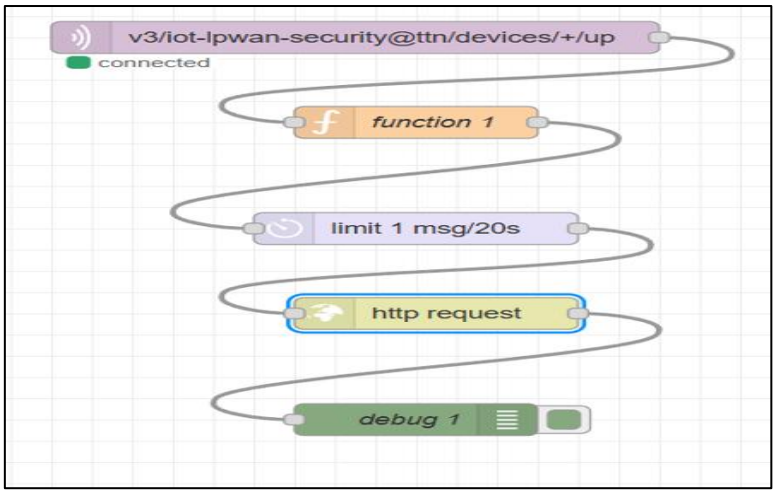
Nota. Elaboración Propia.

En la figura 41 se observa el estado anormal de la ejecución durante el ataque de Beacon Flooding, evidenciado por múltiples solicitudes ICMP (echo request) sin sus respectivas respuestas, lo que indica pérdida de paquetes y degradación de la disponibilidad del sistema IoT afectado.

3.12 Diseño del flujo en Node-RED

Para gestionar el procesamiento de los datos recibidos desde la red LoRaWAN, se diseñó un flujo en Node-RED conectado al servidor MQTT de TTN mediante un tópico de suscripción. A partir de este nodo de entrada, se desprenden distintas funciones encargadas de interpretar la información correspondiente a la temperatura, la humedad y el sensor infrarrojo, así como de generar las alarmas correspondientes y enviar los datos hacia las plataformas de visualización y notificación configuradas para el proyecto.

Figura 42. *Diseño de flujo Node-RED.*



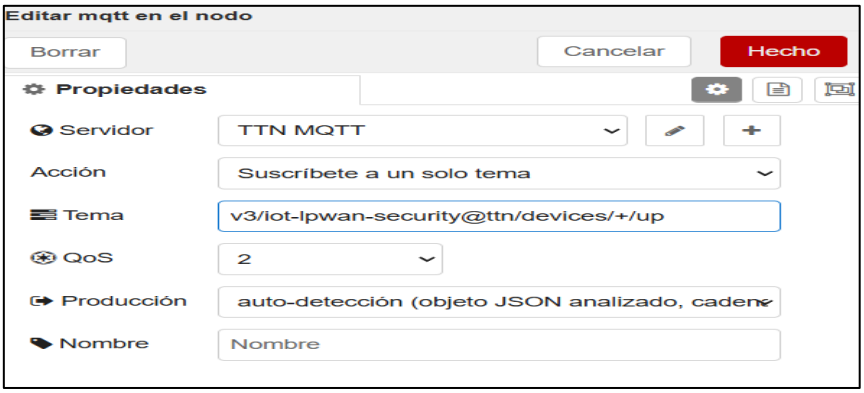
Nota. Elaboración Propia.

En la Figura 42 se muestra el flujo de procesamiento implementado en Node-RED, estructurado de acuerdo con el tipo de variable recibida. Cada flujo dirige los datos hacia los nodos correspondientes para su registro histórico, generación de alarmas y visualización en el panel de control del dashboard.

3.12.1 Configuración del nodo de entrada MQTT (conexión con TTN)

En esta sección se explica la configuración del nodo ubicado al inicio del flujo corresponde a la entrada de datos mediante el protocolo MQTT, encargado de recibir la información transmitida por los dispositivos LoRaWAN a través de la plataforma The Things Network.

Figura 43. *Configuración del nodo MQTT de entrada.*



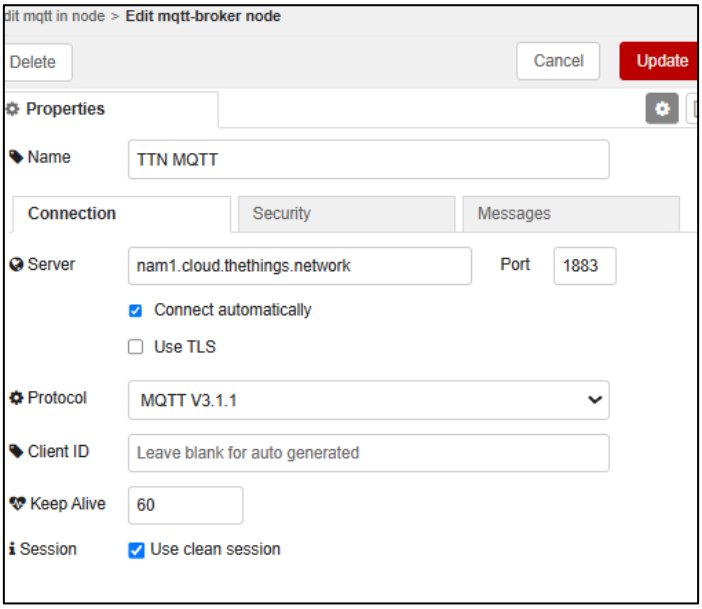
Nota. Elaboración Propia.

En esta figura 43 se observa la configuración del modo suscripción a un único tema (topic), definido como v3/iot-lpwan-security@ttn/devices/+/up, el cual corresponde a la ruta estándar utilizada por TTN para publicar los mensajes ascendentes (uplinks) de todos los dispositivos registrados dentro de la aplicación del proyecto.

El nivel de calidad de servicio (QoS) se estableció en 2, garantizando que cada mensaje fuera entregado exactamente una vez, sin duplicados ni pérdidas durante la transmisión.

El formato de salida se configuró en modo de autodetección, permitiendo que Node-RED interpretara automáticamente el contenido recibido como un objeto JSON estructurado.

Figura 44. *Configuración del nodo bróker TTN MQTT.*

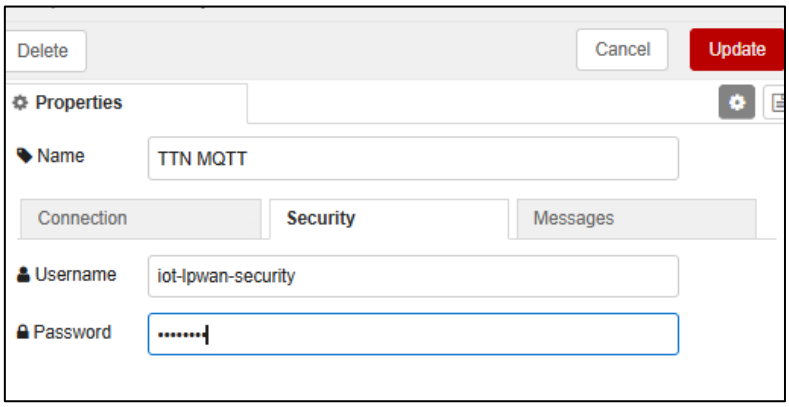


Nota. Elaboración Propia.

En la figura 44 se ilustra la conexión hacia el servidor donde se gestionó mediante un nodo de tipo bróker MQTT, identificado como "TTN MQTT", configurado con la dirección del servidor nam1.cloud. thethings. network a través del puerto 1883, utilizando la versión 3.1.1 del protocolo MQTT. Se habilitó la opción de conexión automática, así

como el uso de sesión limpia (clean session), con un intervalo de keep alive de 60 segundos, destinado a mantener activa la conexión entre Node-RED y el servidor de TTN.

Figura 45. *Configuración de seguridad del bróker.*



Nota. Elaboración Propia.

En la figura 45 se ilustra la autenticación hacia el bróker se realizó mediante credenciales de seguridad, conformadas por un nombre de usuario y una contraseña asociados a la aplicación registrada en la plataforma.

USUARIO: iot-lpwan-security

CONTRASEÑA:12345678

3.12.2 Verificación de conexión

En esta sección se realizó una prueba de envío para confirmar que la plataforma recibía correctamente la información generada por el nodo.

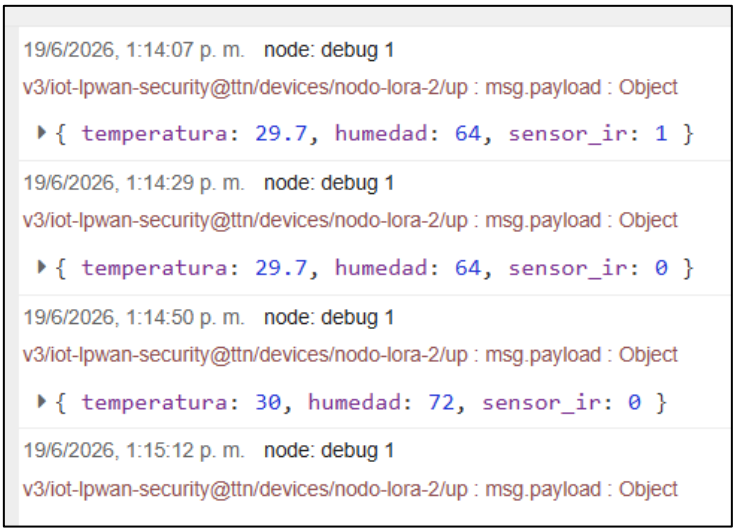
Figura 46. *Respuesta de datos en Node-RED.*

Nota. Elaboración Propia.

En la Figura 46 se observa que, a través del nodo **Debug**, se pudo verificar que la solicitud HTTP devolvió un código de estado **200**, lo que indicó que los datos de temperatura, humedad y estado de la alarma se estaban enviando correctamente al canal correspondiente.

Correcciones realizadas

Figura 47. *Verificación de Datos.*



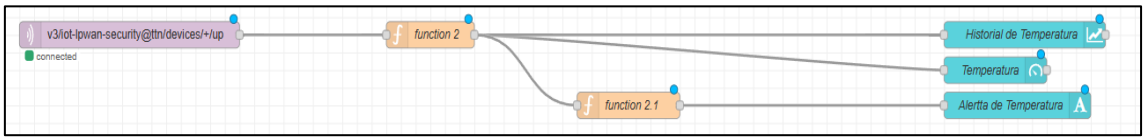
Nota. Elaboración Propia.

En la Figura 47 se observa la verificación de los datos, lo que permitió validar su recepción en tiempo real y confirmar que esta alternativa podía emplearse como mecanismo complementario al monitoreo realizado dentro del entorno de Node-RED.

3.12.3 Configuración y monitoreo de la temperatura en el Dashboard

A partir de la función 2, encargada de extraer el valor de temperatura recibido desde la red LoRaWAN, se distribuyó la información hacia tres nodos del dashboard, destinados a representar gráficamente la variable monitoreada.

Figura 48. *Bloque para el sensor de temperatura en el dashboard.*



Nota. Elaboración Propia.

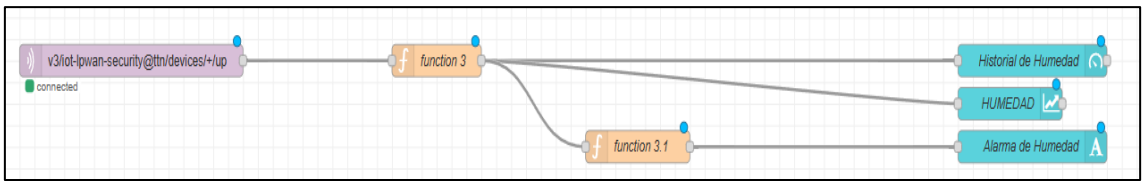
En la Figura 48 se observa el diagrama de flujo correspondiente a la temperatura. El primero corresponde a un nodo de tipo gráfico (**Chart**), denominado **Historial de Temperatura**, utilizado para registrar el comportamiento de esta variable a lo largo del tiempo mediante una línea de tendencia. El segundo corresponde a un nodo de tipo medidor (**Gauge**), configurado con un rango de 0 a 50 °C y segmentado por colores para identificar visualmente los niveles normales, de advertencia y crítico de temperatura.

El tercero corresponde a un nodo de texto denominado **Alerta de Temperatura**, empleado para mostrar el estado actual de la variable y la alerta generada cuando esta supera el umbral definido.

3.12.4 Configuración y monitoreo de la humedad en el Dashboard

La función 3 procesa el mensaje recibido desde la red LoRaWAN y extrae el valor correspondiente a la variable de humedad. El dato obtenido se distribuye hacia tres nodos del dashboard configurados para su representación gráfica y visualización.

Figura 49. *Bloque para el sensor de humedad en el dashboard.*



Nota. Elaboración Propia.

En la figura 49 se observa que el primer diagrama corresponde a un nodo de tipo medidor (gauge) como Historial de Humedad, ajustado con un rango de 0 a 100 % y segmentado por colores para diferenciar los niveles normales, de advertencia y críticos de humedad.

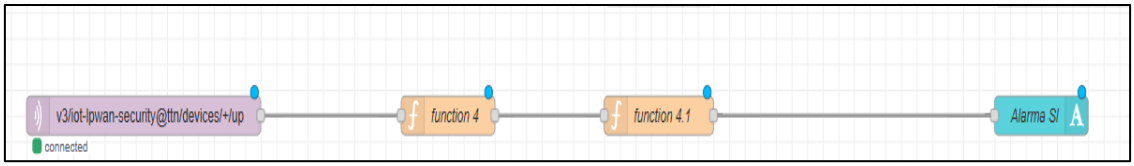
El segundo es un nodo de tipo gráfico (chart) como Humedad, configurado para registrar el historial de la variable a lo largo del tiempo dentro del rango de 60 a 100 %, permitiendo observar su comportamiento de manera continua.

El tercero es un nodo de texto, empleado para mostrar el estado actual de la humedad junto con la alerta correspondiente cuando el valor supera el umbral establecido.

3.12.5 Configuración de la alarma por detección de obstáculos (sensor IR)

Esta configuración permitió visualizar de manera inmediata si el sensor IR detectaba la presencia de un objeto dentro del área monitoreada, complementando el seguimiento de las variables ambientales con un indicador adicional orientado a la seguridad física del entorno.

Figura 50. *Bloque para el sensor infrarrojo en el dashboard.*



Nota. Elaboración Propia.

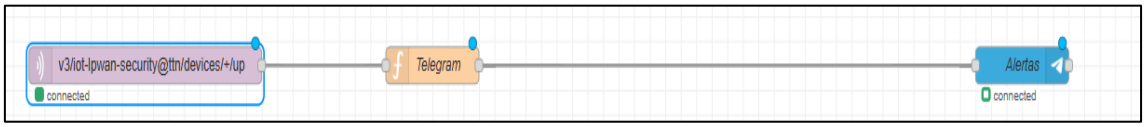
En la figura 50 se observa la función 4, encargada de procesar el valor reportado por el sensor infrarrojo del nodo IoT, la información se canalizó hacia la función 4.1, donde se evaluó la condición de presencia para determinar el estado de alarma correspondiente.

El resultado de este procesamiento se envió finalmente a un nodo de texto denominado "Alarma SI", ubicado dentro del grupo "Detector de Obstáculos" del dashboard, configurado para mostrar el estado actual de la alarma con una etiqueta personalizada, fuente Arial y texto en color rojo, con el fin de resaltar visualmente cualquier evento de detección frente al resto de indicadores del panel.

3.12.6 Integración del sistema de notificaciones mediante la API de Telegram

Para complementar el monitoreo visual del dashboard, se incorporó un canal de notificaciones automáticas mediante la API de Telegram, integrada directamente en el flujo de Node-RED a través de un nodo de tipo sender.

Figura 51. *Integración de alerta.*

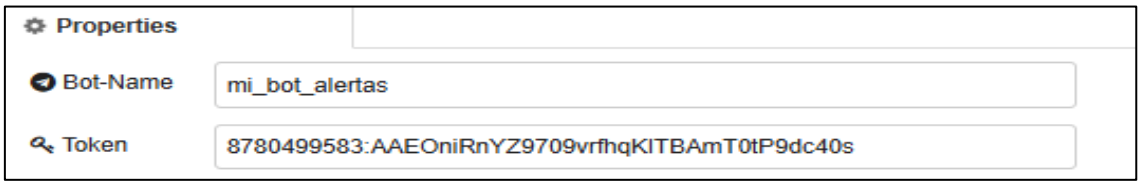


Nota. Elaboración Propia.

En la figura 51 se observa el modo de integración de alerta mediante la actualización se estableció en Polling, mecanismo mediante el cual el bot consulta

periódicamente los servidores de Telegram en busca de nuevos eventos, con un intervalo de verificación configurado en 300 milisegundos.

Figura 52. *Configuración del bots en Node-RED.*



Nota. Elaboración Propia.

En la figura 52 se muestra la configuración del bots personalizado creado previamente a través de BotFather, el servicio oficial de Telegram destinado a la generación y administración de bots. A partir de este proceso se obtuvo un token de autenticación, el cual permite que Node-RED se comunique de manera autorizada con la API de Telegram para el envío de mensajes. Este token fue registrado dentro del nodo de configuración del bot, identificado en el flujo bajo el nombre "mi_bot_alertas".

Token: 8780499583: AAEOniRnYZ9709vrfhqKITBAmT0tP9dc40s

ChatIds: 5505655147

Figura 53. *Chatlds en node red.*

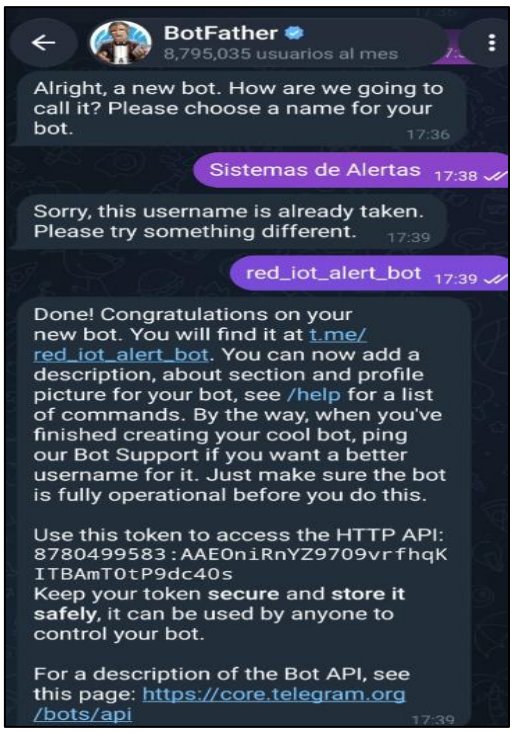
Nota. Elaboración Propia.

En la figura 53 se ilustra el chatlds en node red para tomar en consideración la autenticación para la correcta transmisión de datos hacia aplicaciones externas.

3.13 Creación del bot de Telegram

Para habilitar el envío de notificaciones automáticas hacia el usuario final, se creó un bot personalizado utilizando BotFather, el servicio oficial de Telegram destinado al registro y administración de bots.

Figura 54. *Integración de BotFather.*



Nota. Elaboración Propia.

En la figura 54 se ilustra la integración de BotFather. Durante este proceso se asignó el nombre "Sistemas de Alertas" y el nombre de usuario "@red_iot_alert_bot", quedando el bot disponible para su uso dentro de la plataforma. Una vez generado el bot, Telegram entregó un token de autenticación, requerido posteriormente para vincular el bot con el nodo correspondiente en Node-RED y habilitar el envío de mensajes mediante la API.

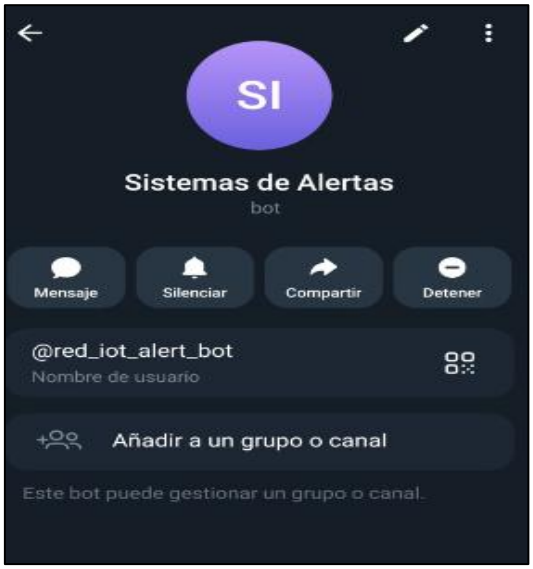
3.13.1 Identificador del chat

Con el bot ya creado, se procedió a obtener (Chat ID) necesario para dirigir las notificaciones hacia la cuenta de usuario correspondiente. Para ello, se envió un mensaje inicial al bot desde la cuenta de Telegram del usuario y, posteriormente, se realizó una consulta a la API de Telegram mediante el método getUpdates, accediendo a través del navegador a la URL del bot con dicho parámetro. La respuesta obtenida, en formato JSON, incluyó los datos del mensaje enviado junto con el identificador numérico del chat, el cual fue extraído del campo "chat":"id" para ser registrado posteriormente en la configuración del nodo de Telegram dentro de Node-RED.

Token: 8780499583: AAEOniRnYZ9709vrfhqKITBAmT0tP9dc40s

Chatt ds:5505655147

Figura 55. *Perfil del bot creado en Telegram.*



Nota. Elaboración Propia.

En la figura 55 se aprecia el perfil del bot como Sistema de Alertas creado en Telegram, cuyo nombre de usuario red_iot_alert_bot.

Figura 56. *Respuesta del método getUpdates con el Chat ID obtenido.*

Nota. Elaboración Propia.

En la figura 56 se observa la respuesta del método getUpdates con el Chat ID obtenido Este procedimiento permitió establecer la comunicación entre el sistema IoT y la cuenta de Telegram del usuario, garantizando que las alertas generadas por las funciones del flujo llegaran de manera directa al destinatario configurado.

CAPÍTULO IV

4.1 Implementación del prototipo IoT

El modelo diseñado para este estudio incluye un nodo LoRa ESP32 que cuenta con sensores de temperatura, humedad y un detector de movimiento.

También se incorpora un Gateway SenseCAP M2, que se encarga de recibir y transmitir los datos a través de la red LoRaWAN. Además, se utiliza un dispositivo móvil para monitorear la información enviada.

Figura 57. *Prototipo IoT.*

Nota. Elaboración Propia.

En la figura 57 se ilustra el prototipo IoT. Esta implementación permitió validar de manera efectiva el intercambio de datos entre los diferentes componentes del sistema antes de llevar a cabo las pruebas experimentales.

4.2 Desempeño de la red en condiciones normales de operación

Como primera etapa para dar cumplimiento al objetivo de analizar el desempeño de la red IoT/LPWAN, se procedió a evaluar el comportamiento del sistema bajo condiciones normales de operación, sin la presencia de interferencias externas ni eventos adversos sobre el canal de comunicación. Para ello, se mantuvo el nodo LoRa transmitiendo de manera continua durante un periodo aproximado de una hora, registrando simultáneamente la telemetría capturada en el dashboard de Node-RED, los eventos recibidos en la consola de la aplicación de The Things Network y el estado operativo del Gateway.

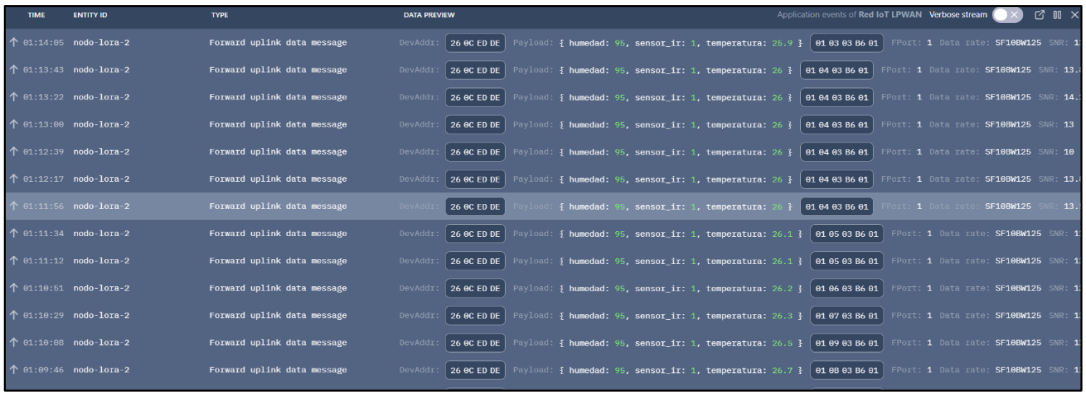
Estos resultados constituyen la línea base del sistema, la cual será empleada posteriormente como parámetro de comparación frente al comportamiento observado durante los escenarios de ataque que se ejecutarán en una etapa posterior de la investigación.

4.3 Recepción de datos del nodo en TTN

A nivel de la plataforma de red, la consola de eventos de la aplicación en The Things Network permitió verificar, mensaje por mensaje, la recepción de los paquetes ascendentes (uplinks) enviados por el dispositivo "nodo-loRa-2", identificado mediante la dirección de dispositivo (DevAddr) 26 0C ED DE.

Cada evento registrado correspondió a un mensaje de tipo "Forward uplink data message", conteniendo el payload decodificado con las tres variables monitoreadas (humedad, sensor_ir y temperatura), además de parámetros propios de la capa física de LoRaWAN, tales como el contador de trama (FCnt), incrementado de manera secuencial en cada transmisión, el puerto de aplicación (FPort 1), la tasa de transmisión de datos (Data rate SF10BW125, correspondiente al factor de dispersión 10 con un ancho de banda de 125 kHz) y la relación señal-ruido (SNR), la cual se mantuvo en un rango favorable de entre 10 y 14 dB.

Figura 58. *Consola de eventos de la aplicación.*



Nota. Elaboración Propia.

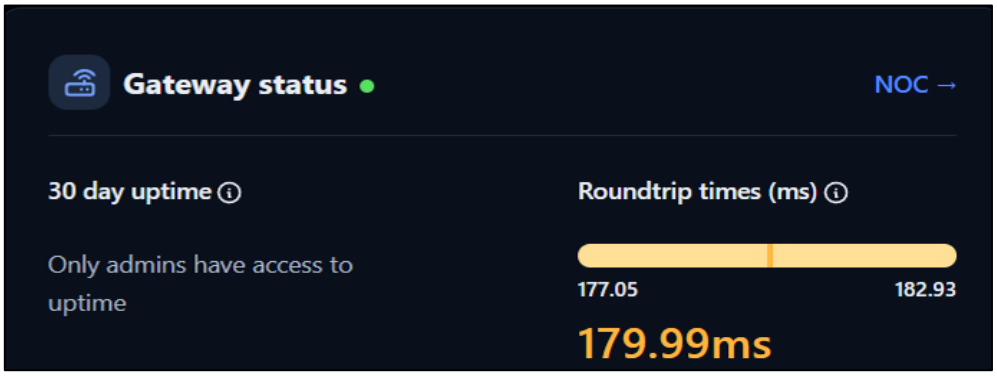
En la figura 58 se observa como los intervalos de tiempo entre mensajes consecutivos, son calculados a partir de las marcas de tiempo registradas, se mantuvieron consistentemente entre 20 y 22 segundos, lo cual coincidió con el intervalo de transmisión

configurado en el firmware del nodo y confirmó la ausencia de mensajes perdidos o retransmitidos durante el periodo observado.

4.4 Estado y conectividad del Gateway

En cuanto a la infraestructura de conectividad, el panel de administración del Gateway "SenseCAP Tesis" (identificado mediante el EUI 2C F7 F1 11 72 80 00 58) reportó un estado activo de manera ininterrumpida, operando bajo el plan de frecuencia 902-928 MHz correspondiente a la región utilizada por la red de TTN.

Figura 59. *Conectividad del Gateway.*



Nota. Elaboración Propia.

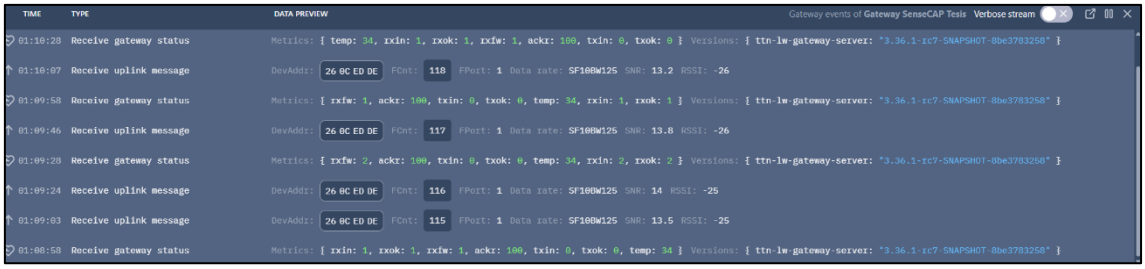
En la figura 59 se observa como el indicador de tiempo de actividad (uptime) de los últimos 30 días mostró un comportamiento estable, mientras que los tiempos de ida y vuelta (roundtrip) de los mensajes se mantuvieron en un rango acotado entre 177.05 ms y 182.93 ms, valores compatibles con una comunicación de baja latencia entre el Gateway y el servidor de red.

4.4.1 Registro de mensajes recibidos

El registro de eventos del propio Gateway, por su parte, permitió observar la alternancia periódica entre mensajes de tipo "Receive uplink message" y reportes de estado ("Receive gateway status"), estos últimos incluyendo métricas internas de calidad

como el porcentaje de paquetes recibidos correctamente (rxok), reenviados (rxfw) y reconocidos (ackr).

Figura 60. *Panel de información general del SenseCAP.*



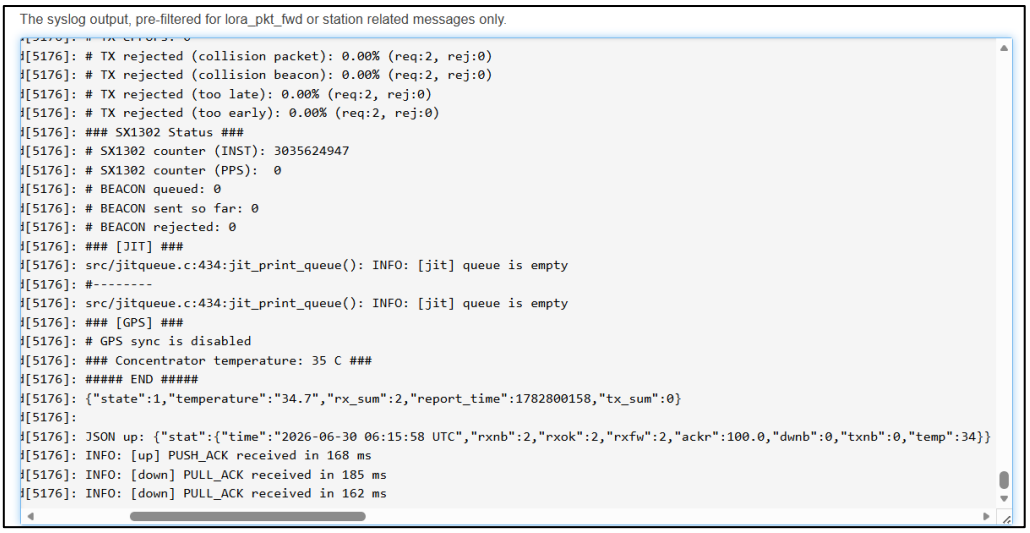
Nota. Elaboración Propia.

En la figura 60 se observa el panel de información general del SenseCAP por lo que se mantuvieron consistentemente en valores óptimos durante todo el periodo de observación, sin evidencia de pérdidas en la capa de concentración.

4.4.2 Registro de bajo nivel del concentrador (syslog del gateway)

A nivel de hardware, el análisis del registro syslog generado por el concentrador SX1302 del Gateway permitió profundizar en el comportamiento de la capa física de la red. Las estadísticas de transmisión reportadas en dicho registro mostraron que los porcentajes de rechazo por colisión de paquetes, colisión con baliza, envíos tardíos y envíos tempranos se mantuvieron en 0.00 % durante el periodo analizado, lo cual indicó la ausencia de conflictos de acceso al canal y de saturación en la cola de transmisión del concentrador. Se evidenciaron valores de temperatura, humedad en °C, valor consistente con un régimen de operación continua dentro de parámetros normales de funcionamiento.

Figura 61. *Salida del syslog del Gateway.*



Nota. Elaboración Propia.

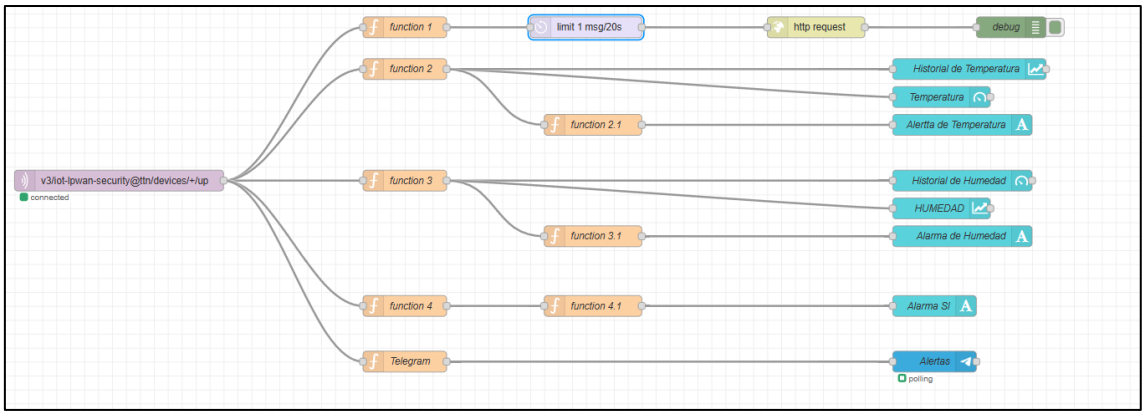
En la figura 61 se ilustra la salida del syslog donde se registra evidencia de la recepción oportuna de las confirmaciones PUSH_ACK y PULL_ACK, correspondientes al protocolo de comunicación entre el Gateway y el servidor de red, en tiempos inferiores a 200 milisegundos en todos los casos observados, lo que reforzó la conclusión de una comunicación estable y de baja latencia entre ambos extremos del enlace.

4.5 Diseño final del flujo de procesamiento en Node-RED

Cada una de estas ramas conserva la estructura descrita previamente en el capítulo, compuesta por una función de extracción y/o evaluación del dato, seguida de los nodos correspondientes de visualización en el dashboard o de notificación externa.

Esta vista general del flujo permite comprobar que la totalidad de los componentes desarrollados, tanto de adquisición como de visualización y alertamiento, quedaron correctamente integrados dentro de un único entorno de trabajo, garantizando un funcionamiento coordinado entre las distintas funciones del sistema.

Figura 62. *Diagrama de flujo.*



Nota. Elaboración Propia.

En la figura 62 se ilustra el flujo completo resultante de la integración de los módulos desarrollados a lo largo del capítulo. A partir del nodo de entrada MQTT, suscrito al tópico de la aplicación en The Things Network, los datos recibidos se distribuyen hacia cinco ramas de procesamiento independientes: el envío de información hacia dashboard mediante solicitud HTTP, el procesamiento de la temperatura, el procesamiento de la humedad, la evaluación del estado del sensor infrarrojo y el envío de notificaciones mediante el bot de Telegram.

4.6 Resultados del monitoreo en el Dashboard

A continuación, se presentan los resultados obtenidos a través del panel de monitoreo configurado en Node-RED, organizados según cada una de las variables registradas por el sistema durante el periodo de prueba: detección de objetos, temperatura y humedad relativa. Estos resultados se obtuvieron directamente desde la interfaz del dashboard accesible mediante la dirección ,127.0.0.1:1800/dashboard/page 1, donde se presentan de forma organizada los indicadores correspondientes a cada grupo configurado previamente en el flujo.

4.6.1 Detección de objetos (sensor infrarrojo)

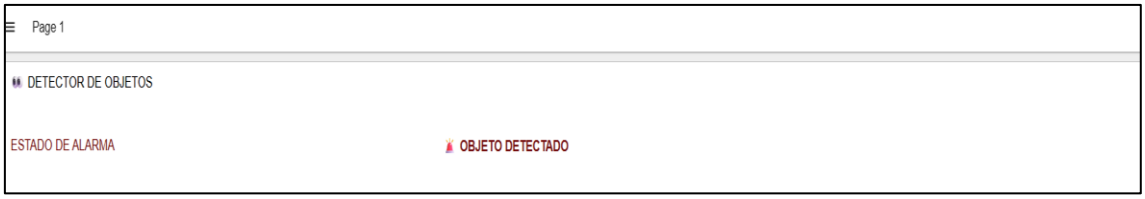
En esta sección se evidenció el correcto funcionamiento de toda la cadena de procesamiento asociada a esta variable, desde la lectura física realizada por el sensor IR conectado al nodo, pasando por la transmisión vía LoRaWAN, hasta la evaluación lógica efectuada en la función correspondiente del flujo y su posterior representación en la capa de visualización. Cabe señalar que, a diferencia de las variables de temperatura y humedad, el sensor IR entrega una señal de tipo binaria (presencia o ausencia de objeto), por lo que su validación se limita a confirmar la correspondencia entre el estado físico esperado durante la prueba y el valor mostrado en el panel.

Figura 63. *Estado normal sin detención.*

Nota. Elaboración Propia.

En la figura 63 se observa el estado normal sin detección. Dentro del grupo "Detector de Objetos", el nodo de tipo text encargado de mostrar el estado de la alarma reportó, en el instante registrado, la detección de un objeto dentro del área de cobertura del sensor infrarrojo, representado mediante la etiqueta "Sin objeto Detectado" resaltada en color rojo.

Figura 64. *Estado con detención.*



Nota. Elaboración Propia.

En la figura 63 se observa el estado con detección. Dentro del grupo "Detector de Objetos", representado mediante la etiqueta " objeto Detectado" resaltada en color rojo.

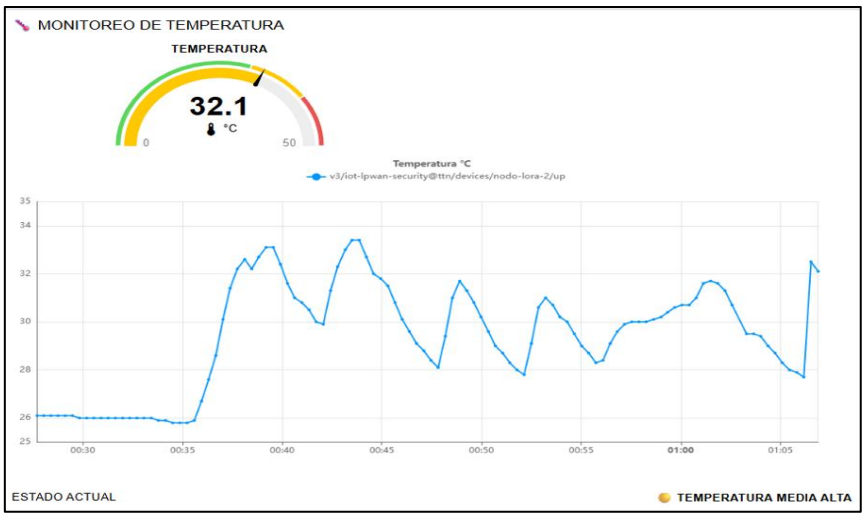
4.6.2 Temperatura

En el mismo instante, el indicador de tipo gauge asociado a la temperatura se ubicó en 32.1 °C, valor representado dentro del segmento amarillo correspondiente al rango medio-alto definido en la configuración del nodo, cuyos límites fueron establecidos previamente entre 0 °C y 50 °C con segmentación por colores en 30 °C y 40 °C.

El historial de esta variable, registrado por el nodo chart del grupo "Monitoreo de Temperatura", mostró una transición desde un valor estable cercano a los 26 °C, mantenido hasta aproximadamente las 00:35, hacia un patrón oscilante con picos de hasta 33.4 °C registrados entre las 00:38 y las 01:05.

La periodicidad de los máximos y mínimos observados en la curva sugiere un comportamiento cíclico de la variable, posiblemente asociado a las condiciones específicas del entorno donde se efectuó la prueba, tales como cambios en la exposición del sensor a fuentes de calor cercanas.

Figura 65. *Monitoreo de Temperatura.*



Nota. Elaboración Propia.

En la figura 65 se observa el monitoreo de temperatura, donde se evidencia la continuidad de la curva, sin interrupciones ni saltos abruptos no justificados por el comportamiento físico de la variable, permitió confirmar que la transmisión de datos entre el nodo sensor y el dashboard se mantuvo estable durante la totalidad del periodo evaluado, sin evidencia de pérdida de muestras en la serie temporal.

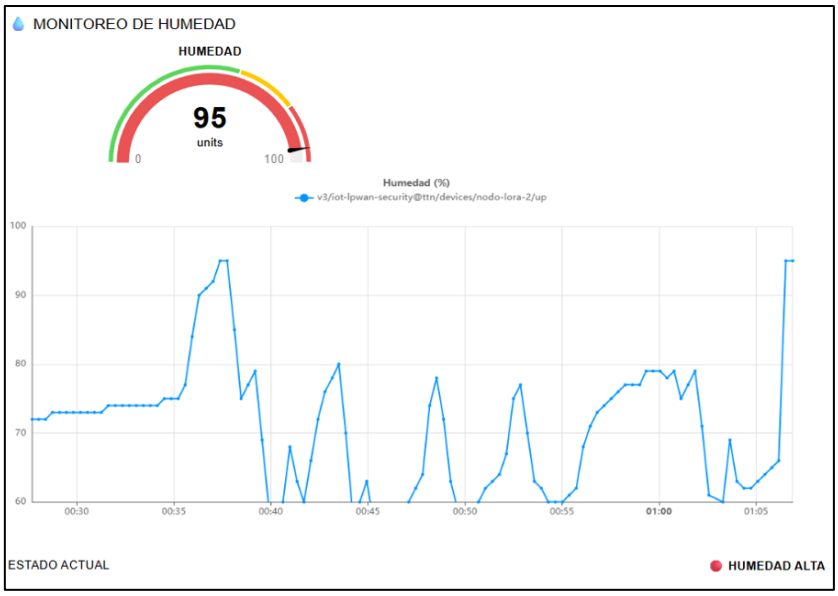
4.6.3 Humedad

De forma simultánea, el indicador de humedad muestra un valor de 95 unidades, clasificado dentro del segmento de color rojo definido en la configuración del nodo gauge, cuyo rango fue establecido entre 0 % y 100 %, con umbrales de segmentación en 60 % y 80 % para diferenciar los niveles normales, de advertencia y crítico.

El registro histórico de esta variable, capturado mediante el nodo chart configurado con un eje temporal y un límite de visualización de una hora, evidenció un comportamiento marcadamente fluctuante a lo largo del periodo de observación, con valores comprendidos entre un mínimo de 60 % y un máximo de 95 %. Entre las 00:40 y las 01:00 se identificaron múltiples picos de corta duración y rápida recuperación, atribuibles a variaciones puntuales en las condiciones del ambiente de prueba, mientras que a partir de las 01:05 se observa un incremento sostenido que llevó a la variable a estabilizarse en su valor máximo durante aproximadamente diez minutos, antes de iniciar un descenso progresivo hacia valores cercanos al 70 %.

Este comportamiento permitió constatar que el sensor DHT11 mantuvo una resolución adecuada para capturar tanto cambios graduales como variaciones abruptas en la humedad del entorno, y que el sistema de adquisición no presentó vacíos ni discontinuidades significativas en la serie de datos registrada.

Figura 66. *Monitoreo de Humedad.*



Nota. Elaboración Propia.

En la figura 66 se ilustra el monitoreo de Humedad evidenciando los valores numéricos mostrados en cada indicador y la codificación por colores empleada en los nodos gauge permitió validar que la lógica de segmentación configurada respondía adecuadamente a los umbrales establecidos durante la fase de diseño del sistema.

Asimismo, la correspondencia temporal entre los valores puntuales observados en los medidores y las tendencias reflejadas en los gráficos de historial permitió confirmar la consistencia interna del sistema de monitoreo, en el sentido de que ambos tipos de visualización (instantánea e histórica) reflejaron de manera coherente el mismo conjunto de datos recibidos desde el nodo IoT.

4.6.4 Notificaciones recibidas en Telegram

Como complemento al monitoreo realizado en el dashboard, se verificó el funcionamiento del canal de notificaciones configurado mediante el bot "Sistemas de Alertas" en Telegram.

Durante el periodo de prueba, el bot envió de manera automática tres mensajes correspondientes a cada una de las variables monitoreadas por el sistema, cada uno con su valor registrado, el estado clasificado según los umbrales definidos y la hora exacta del evento.

Figura 67. *Sistemas de Alertas.*

Nota. Elaboración Propia.

En la figura 67 se muestran los sistemas de alertas en el caso de la temperatura, el mensaje reportó un valor de 32.5 °C con un estado clasificado como "Media", recibido a las 01:06:31 a.m. Para la humedad, se notificó un valor de 95 %, clasificado como estado "Alta", registrado en el mismo minuto. Por su parte, el sensor infrarrojo generó una notificación independiente indicando la detección de un objeto, con estado "Detectado" y hora de envío 01:06:53 a.m., apenas veintidós segundos después de los dos mensajes anteriores.

La recepción oportuna de estas notificaciones, junto con la correspondencia entre los valores reportados por Telegram y los observados de manera simultánea en el dashboard de Node-RED, permitió confirmar que el canal de alertas funcionó de forma sincronizada con el resto del sistema, garantizando que el usuario pudiera recibir información relevante sobre el estado del entorno monitoreado sin necesidad de mantener abierta la interfaz gráfica del dashboard.

4.7 Resultados obtenidos durante la ejecución del ataque Beacon Flooding

Para analizar el desempeño de la conexión inalámbrica frente a una inundación de tramas Beacon, Se ejecuto ensayos controlados empleando el hardware Aursinc Wi-Fi Deauther. Durante esta práctica, se configuró múltiples nombres de red falsos semejantes a la señal real del laboratorio, emulando la suplantación de identidades de redes mediante la constante difusión de paquetes Beacon.

Dicha estrategia facilitó observar cómo la proliferación de redes inexistentes afecta a los equipos receptores, notando la presencia de muchos puntos de acceso ficticios al buscar señales y el incremento en datos operativos por este ataque. Seguidamente, se presentó los resultados obtenidos en el proceso, incluyendo el ajuste del ataque, la emisión de señales falsas, detección por parte de usuarios y el estudio del tráfico capturado en Wireshark.

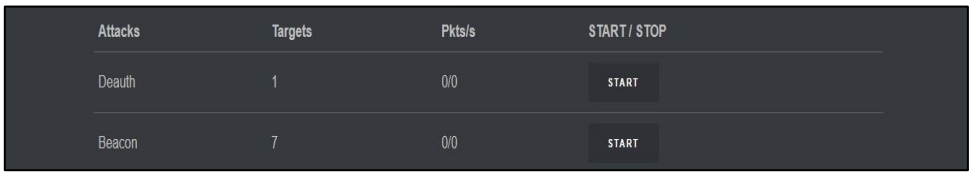
4.7.1 Configuración del ataque Beacon Flooding

En esta sección se muestra la configuración del ataque Beacon. Este tipo de tramas tiene como objetivo notificar la existencia de redes inalámbricas, aunque eso no implica que los puntos de acceso relacionados realmente existan.

La emisión continua de este tráfico busca aumentar de forma artificial el número de SSID visibles en el área de prueba, simulando la presencia concurrente de múltiples

redes y complicando la identificación clara de cuál es la red legítima del laboratorio por parte de los dispositivos clientes.

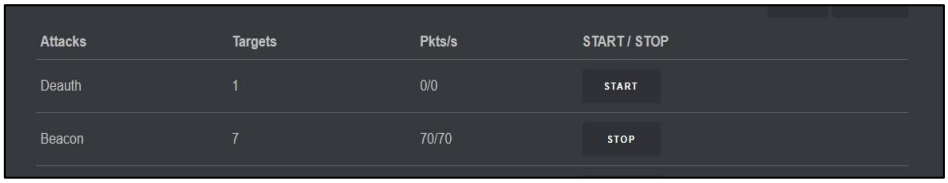
Figura 68. *Función de Beacon inactiva.*



Nota. Elaboración Propia.

En la figura 68 se observa el estado inactivo, sin ataque Beacon. Dando como resultado sin ninguna selección de tramas.

Figura 69. *Función de Beacon activa.*



Nota. Elaboración Propia.

En la figura 69 se observa la configuración del ataque de Beacon Flooding se llevó a cabo utilizando el dispositivo Deauther Watch V3.

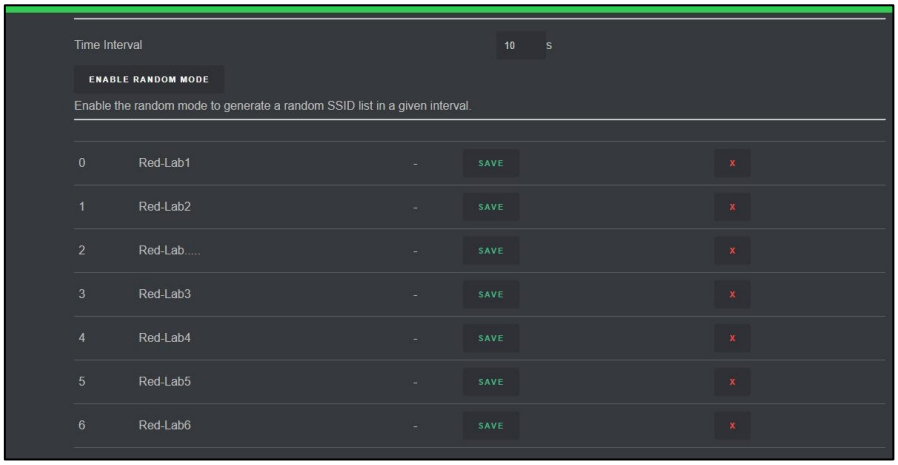
En la pantalla del dispositivo se puede observar que la función de ataque Beacon está activa, generando un promedio de 70 tramas Beacon cada segundo.

4.7.2 Configuración de los SSID utilizados durante el ataque

Esta táctica tuvo como objetivo simular la presencia de varias redes que parecían estar conectadas a la infraestructura original, lo que causó confusión en el usuario al intentar identificar cuál de ellas era el acceso real.

La propagación al mismo tiempo de estos identificadores se realizó a través de tramas Beacon, enviadas de manera ininterrumpida por el dispositivo que atacaba.

Figura 70. *Tramas falsas de Beacon.*



Nota. Elaboración Propia.

En la figura 70 se presenta la configuración de los identificadores de red utilizados durante la realización del ataque. En esta evaluación se establecieron siete nombres de red que eran similares al de la red auténtica del laboratorio, incluyendo Red-Lab1, Red-Lab2, Red-Lab3, Red-Lab4, Red-Lab5 y Red-Lab6.

4.7.3 Redes inalámbricas detectadas durante la ejecución del ataque

La detección simultánea de estos SSID evidencia la recepción de múltiples tramas falsas generadas durante el escenario de Beacon Flooding, provocando que el sistema operativo del cliente identificara y mostrara diversos puntos de acceso correspondientes a redes inalámbricas no pertenecientes a la infraestructura original.

Este hallazgo muestra cómo un ataque de Beacon Flooding puede incrementar la cantidad de redes que se encuentran y generar confusión para el usuario al momento de elegir la red inalámbrica a la que desea conectarse.

Figura 71. *Visualización de Redes Beacon.*

Nota. Elaboración Propia.

En la figura 71 se muestra los hallazgos del análisis de redes inalámbricas realizado desde un dispositivo cliente mientras se ejecutaba el ataque de Beacon Flooding. Se puede notar que, además de la red legítima llamada Red-Lab, aparecen varias otras redes con nombres parecido a la red principal.

4.7.4 Comparación entre la red legítima y las redes falsas

Esta sección corresponde a los datos relacionados con la red válida Red-Lab, que sirve como referencia durante el proceso de la evaluación.

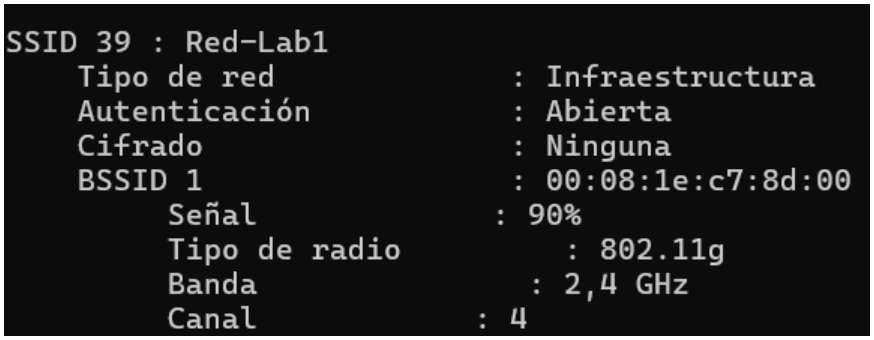
Figura 72. *Red Legitima.*

```
SSID 1 : Red-Lab
    Tipo de red             : Infraestructura
    Autenticación           : WPA2-Personal
    Cifrado                 : CCMP
    BSSID 1                 : 2e:2f:23:09:e7:b2
         Señal              : 95%
         Tipo de radio      : 802.11ac
         Banda              : 2,4 GHz
         Canal              : 13
```

Nota. Elaboración Propia.

En la figura 72 se observa la red legitima que a diferencia de los SSID falsos que surgen del ataque de Beacon Flooding, esta red tiene autenticación WPA2-Personal y cifrado CCMP, lo que proporciona un acceso seguro a los usuarios que están autorizados.

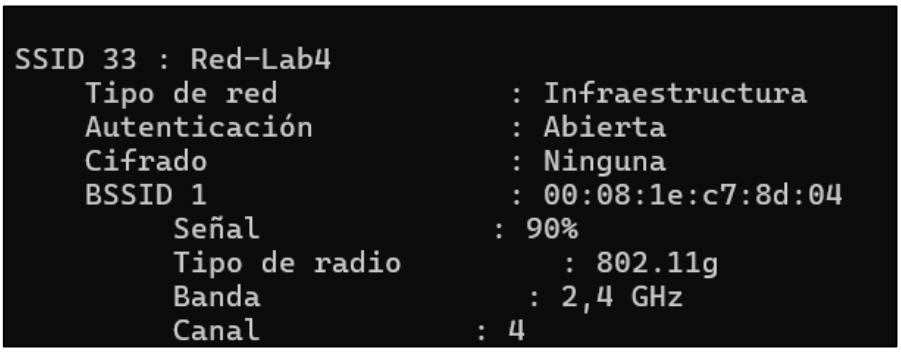
Figura 73. *Red-Lab1 Beacon.*



Nota. Elaboración Propia.

En la figura 73 se observa una comparación de red-Lab1, se pueden identificar claramente las distinciones entre la infraestructura inalámbrica auténtica y las redes simuladas creadas por el atacante.

Figura 74. *Red-Lab4 Beacon.*



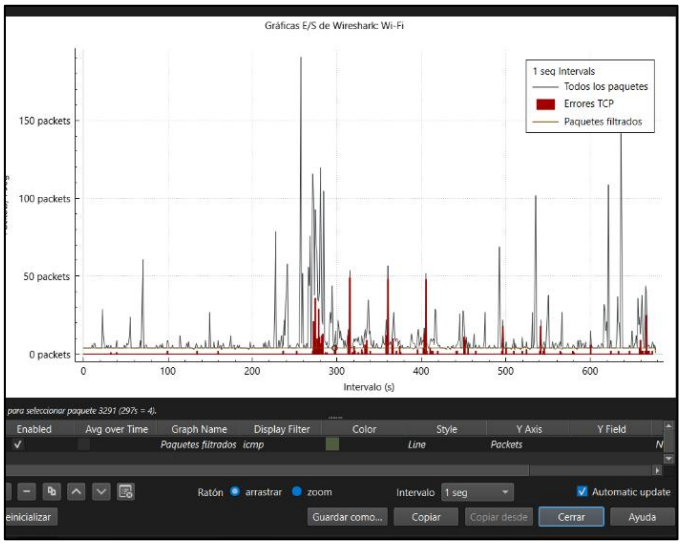
Nota. Elaboración Propia.

En la figura 74 se observa una comparación de red-Lab4, se pueden identificar claramente las distinciones entre la infraestructura inalámbrica legitima y la red falsa.

4.7.5 Análisis del tráfico generado durante el ataque mediante Wireshark

En esta sección se observa como el incremento en el tráfico de administración confirma que el ataque generó una distribución constante de anuncios de redes inalámbricas fraudulentas, lo que elevó considerablemente la actividad registrada en el canal inalámbrico durante el tiempo de evaluación.

Figura 75. *Comportamiento de Beacon en Wireshark.*



Nota. Elaboración Propia.

En la Figura 75 se presenta el I/O Graph obtenido a través de Wireshark durante la realización del ataque de Beacon Flooding. En este gráfico se pueden observar numerosas fluctuaciones en el número de paquetes capturados, representadas por picos en el tráfico que corresponden a la emisión continua de tramas Beacon originadas por el dispositivo atacante.

4.8 Resultados durante la ejecución del ataque de Desautenticación

Para estudiar el impacto del ataque de desautenticación en la conectividad, se ejecutó este proceso mediante un equipo Deauther, monitoreando la red con Wireshark, el Gateway SenseCAP M2, The Things Network y Node-RED. Durante estas

evaluaciones, se recopilaron evidencias que facilitaron la comparación del funcionamiento del esquema previo y durante la intrusión, detectando fallos en la transmisión entre los diferentes componentes del ecosistema IoT.

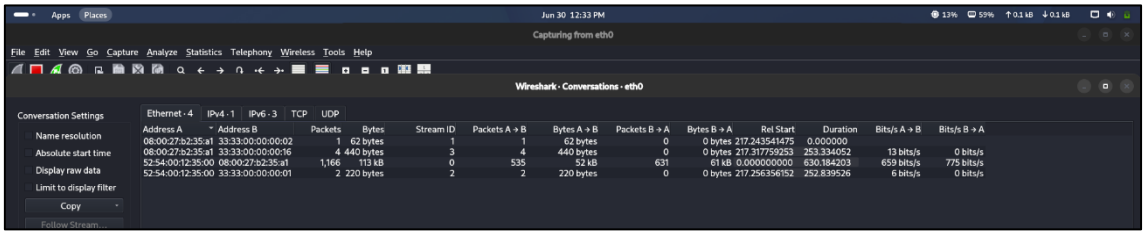
Los resultados indicaron que el ataque mantuvo intactos los ajustes del gateway y de los terminales; no obstante, generó un bloqueo momentáneo del enlace inalámbrico, afectando la operatividad del sistema. Como consecuencia, el gateway interrumpió temporalmente la entrega de reportes hacia TTN, provocando una breve interrupción en el flujo de información de Node-RED y en las herramientas digitales destinadas al monitoreo y control.

4.8.1 Análisis en Wireshark

En este apartado se detalla cómo estos datos permitieron verificar la existencia de conectividad previa al incidente y analizar posteriormente las irregularidades presentadas tras la interrupción de la señal.

Aunque esta vista no permite evidenciar de manera explícita la ejecución del ataque de desautenticación, proporciona evidencias del intercambio habitual de información entre los dispositivos involucrados y facilita el análisis del tráfico registrado durante los experimentos.

Figura 76. Conversaciones registradas en Wireshark.



Nota. Elaboración Propia.

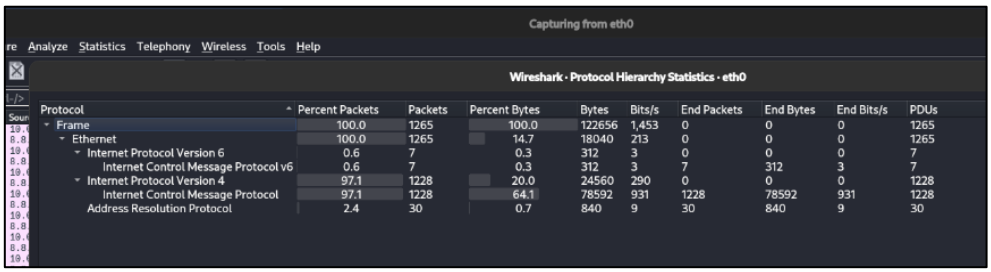
En la figura 76 se observa el panel Conversations en Wireshark donde facilitó identificar los equipos que interactuaron mediante el análisis del tráfico capturado. Aquí

se logró observar la transferencia de datos entre las conexiones activas, junto con el volumen global de paquetes y bytes movidos en ambos sentidos.

4.8.2 Análisis de protocolos en Wireshark

La pantalla de Jerarquía de Protocolos de Wireshark muestra cómo se reparten los protocolos hallados al capturar la actividad en la red. Esta utilidad permite ver fácilmente qué protocolos están activos y cuántos paquetes forman parte de cada uno.

Figura 77. *Jerarquía de protocolos (Protocol Hierarchy).*



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	1265	100.0	122656	1,453	0	0	0	1265
Ethernet	100.0	1265	14.7	18040	213	0	0	0	1265
Internet Protocol Version 6	0.6	7	0.3	312	3	0	0	0	7
Internet Control Message Protocol v6	0.6	7	0.3	312	3	7	312	3	7
Internet Protocol Version 4	97.1	1228	20.0	24560	290	0	0	0	1228
Internet Control Message Protocol	97.1	1228	64.1	78592	931	1228	78592	931	1228
Address Resolution Protocol	2.4	30	0.7	840	9	30	840	9	30

Nota. Elaboración Propia.

En la figura 77 se observa la captura de datos, donde se obtuvieron 1265 paquetes, sumando el 100% del tráfico estudiado. Del total, IPv4 significó el 97,1 %, mientras que IPv6 apenas alcanzó un 0,6 %. Esto indica que la conexión en el entorno fue mayormente para IPv4, pues la red y los equipos usados dependían de este formato.

La poca presencia de IPv6 no evidencia un menor desempeño ni una desventaja tecnológica si no que se centra que este protocolo tuvo una participación limitada durante la captura realizada.

Asimismo, el protocolo ICMP sumó el 97,1 % del tráfico total (1228 paquetes), ya que se hicieron pruebas constantes de conexión con mensajes Echo Request y Echo Reply para verificar el estado durante el ataque de desautenticación. Respecto al protocolo ARP, supuso el 2,4 % del tráfico (30 paquetes), dado que resulta clave para traducir direcciones IP a direcciones MAC, logrando así enlazar los equipos conectados a la red.

Además, el sistema entrega cifras como el porcentaje de bytes enviados y el ritmo de transferencia (Bits/s), lo cual ayuda a estudiar el flujo del tráfico durante las pruebas y comparar cómo funcionaba la red antes y durante el ataque.

4.8.3 Análisis de Información de expertos

Una pieza clave de evidencia se obtuvo mediante la función **Expert Information** de Wireshark. Durante la ejecución del ataque, se generaron numerosas alertas relacionadas con la falta de respuesta y varios avisos de **"Destination unreachable"**, lo que indicó que una parte significativa de los paquetes enviados no alcanzó su destino.

Figura 78. *Información de expertos (Expert Information).*

Nota. Elaboración Propia.

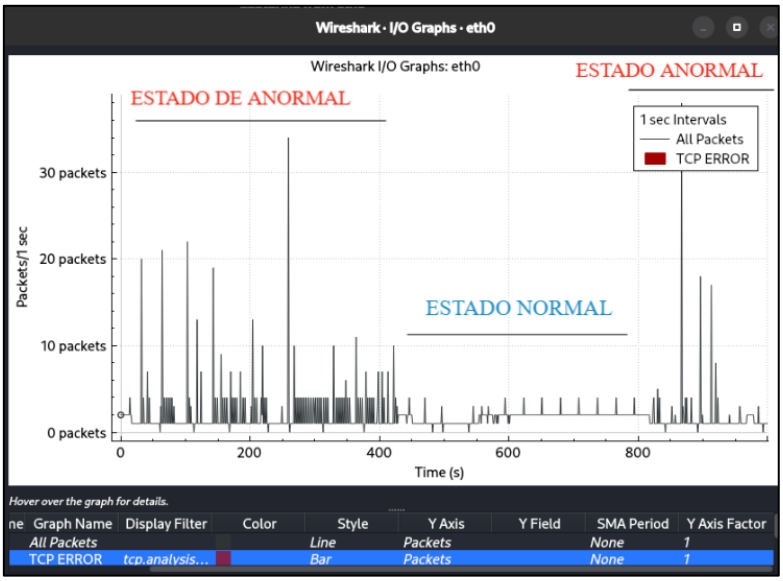
En esta figura 78 se ilustra hechos donde prueban que hubo cortes momentáneos en la conexión por el ataque de desautenticación, pues el equipo no logró intercambiar información con la red brevemente. Reiterar estos avisos es una prueba contundente de la caída de conectividad que sucedió durante esta evaluación.

4.8.4 Análisis de (I/O Graph) en Wireshark

La gráfica sobre el flujo de paquetes ayuda a visualizar el movimiento de la red durante el proceso de captura. Se puede observar variaciones en el volumen de datos por segundo, reflejando altibajos en la carga conforme el ataque ocurría.

Cabe aclarar que esta imagen presenta el panorama global del tráfico, así que no constituye un indicio absoluto de la intrusión. Sin embargo, al contrastarla con pruebas extra, como fallos de acceso o avisos de Wireshark, se vinculó con anomalías con el ataque de desautenticación.

Figura 79. *Gráfico de entrada y salida de paquetes (I/O Graph).*



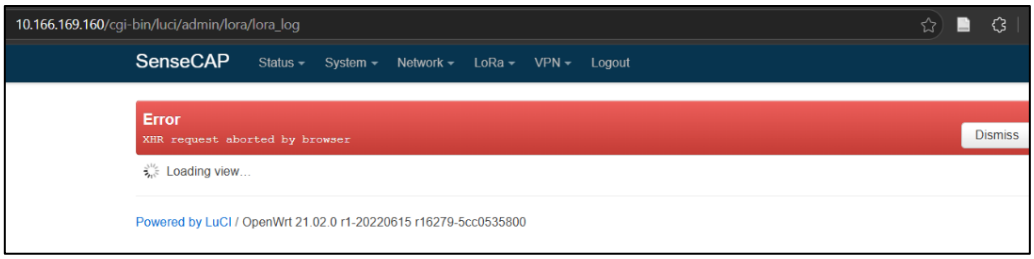
Nota. Elaboración Propia.

En la figura 79 se observa el Gráfico de entrada y salida de paquetes, evidenciando el estado normal y anormal del tráfico de la red en Wireshark.

4.8.5 Análisis de la interrupción de la comunicación con el SenseCAP

En esta sección de muestra la imposibilidad de entrar al sistema ratificó que el ataque provocó un corte en la señal inalámbrica del Gateway, impidiendo el tráfico normal de datos mientras la desconexión seguía operando.

Figura 80. *Interrupción del SenseCAP.*



Nota. Elaboración Propia.

En la figura 80 se muestra cómo actúa el dispositivo ante ataques, el panel de control del Gateway lanzó un mensaje informando que la petición del explorador falló. Esto confirmó que el enlace entre el equipo gestor y el Gateway resultó momentáneamente afectado.

4.8.6 Análisis de Interrupción del flujo de eventos en The Things Network

De acuerdo con el análisis se verifica el corte en el flujo de información evidenció que el equipo perdió contacto con la red LoRaWAN, provocando una pausa obligada en la entrega de paquetes del sensor.

Figura 81. *Interrupción en TTN.*

Nota. Elaboración Propia.

En la figura 81 se observa que dentro de la consola TTN surgió un aviso de conexión finalizada, lo cual indicaba que la puerta de enlace dejó de transmitir reportes mientras duró la ofensiva.

4.8.7 Análisis de Interrupción en la recepción de datos en Node-RED

Debido a la desconexión del Gateway, Node-RED dejó de recibir nuevas tramas de TTN. Durante este tiempo, el flujo se mantuvo sin cambios, lo que provocó que no aparecieran datos recientes en el panel y que no se enviara información a las plataformas

de monitoreo. Una vez que finalizó el ataque y se restableció la conexión inalámbrica, el flujo de datos volvió a la normalidad, reestableciendo la comunicación entre todos los componentes del sistema.

Figura 82. Interrupción de datos en Node-RED.

Nota. Elaboración Propia.

En la figura 82 se observa la interrupción o pérdida de información, evitando la correcta transmisión de datos.

4.8.8 Análisis del monitor de paquetes del reloj Deauther

Es relevante destacar que esta cifra refleja el tráfico inalámbrico registrado por el dispositivo y no se limita únicamente a los paquetes de desautenticación.

No obstante, sirve como prueba de que el reloj estaba ejecutando un monitoreo activo del canal durante la prueba y avala la existencia de actividad inalámbrica mientras el ataque se estaba llevando a cabo.

Figura 83. *Paquetes del reloj Deauther.*

Nota. Elaboración Propia.

En la figura 83 se observa que, durante la realización del ataque, se empleó el monitor de paquetes que viene integrado en el reloj Deauther para analizar el tráfico inalámbrico en el canal elegido. En la pantalla se capturó un flujo de aproximadamente 687 paquetes por segundo, junto con una representación gráfica en tiempo real de la actividad detectada.

4.9 Propuesta de medidas de mitigación

Basándose en los hallazgos de las pruebas experimentales, se propone estrategias de mitigación que pueden ayudar a mejorar la integridad, disponibilidad y fiabilidad de la red LoRaWAN que se ha implementado.

Estas acciones están diseñadas para reducir el impacto de posibles ataques en la infraestructura de comunicación y optimizar el rendimiento general del sistema IoT.

4.9.1 Refuerzo de la seguridad de la red Wi-Fi

Para mejorar la seguridad de la red Wi-Fi que utiliza el Gateway SenseCAP M2, se recomienda adoptar protocolos de seguridad como WPA2 o WPA3, establecer contraseñas robustas y desactivar el WPS. Estas medidas dificultan que dispositivos no autorizados intenten conectarse a la red o exploten configuraciones vulnerables para comprometer la comunicación.

Aunque los ataques de desautenticación se aprovechan de fallos de gestión del estándar IEEE 802.11 y no dependen directamente de la contraseña de la red, mantener una configuración segura reduce las oportunidades de ataque, previene accesos no autorizados y fortalece la seguridad general de la red inalámbrica en la que opera el Gateway.

4.9.2 Vigilancia constante de la red

Se sugiere realizar un monitoreo continuo del tráfico con herramientas como Wireshark, con el fin de detectar a tiempo comportamientos anómalos, interrupciones en la conectividad o la aparición de paquetes que puedan estar relacionados con ataques potenciales.

4.9.3 Actualización regular de los dispositivos

Es importante mantener el firmware del Gateway SenseCAP M2, del nodo ESP32 LoRa y de los demás dispositivos que forman parte de la infraestructura siempre actualizado, para corregir vulnerabilidades conocidas y mejorar el funcionamiento del sistema.

4.9.4 Verificación de la información recibida

Es recomendable implementar métodos para verificar la información que se recibe en Node-RED. Esto es fundamental para asegurarnos de que los datos provenientes de The Things Network (TTN) sean valores válidos, especialmente en lo que respecta a temperatura, humedad y detección de presencia.

4.9.5 Salvaguardar credenciales y claves de acceso

Es crucial proteger las claves de autenticación utilizadas por LoRaWAN y las credenciales de acceso a TTN, Node-RED y Telegram, evitando su divulgación y reduciendo el riesgo de accesos no autorizados a la infraestructura implementada.

4.9.6 Fortalecer el sistema de notificaciones

Se propone optimizar el sistema de notificaciones que se ha implementado a través de Telegram. La idea es incluir alertas adicionales que informen de inmediato sobre cualquier desconexión entre el Gateway y The Things Network (TTN), la ausencia de recepción de datos o la detección de actividades inusuales en la red. De esta manera, el

usuario podrá estar al tanto del estado del sistema en tiempo real y reaccionar rápidamente ante posibles problemas que puedan afectar la disponibilidad y continuidad del servicio.

CONCLUSIONES

La implementación de una red de IoT utilizando la tecnología LoRaWAN ha permitido establecer una comunicación fluida entre el nodo ESP32 LoRa, el Gateway SenseCAP M2 y plataformas como TTN, Node-RED, y Telegram.

Los resultados obtenidos confirmaron que la arquitectura desarrollada es capaz de adquirir, transmitir, visualizar y alertar sobre la información generada por los sensores, validando así el correcto funcionamiento del sistema en condiciones operativas normales.

Las pruebas de seguridad realizadas con el dispositivo Aursinc Wi-Fi Deauther mostraron que los ataques de desautenticación y beacon flooding impactan la disponibilidad de la red Wi-Fi utilizada por el Gateway, provocando interrupciones momentáneas en la comunicación con The Things Network y retrasos en la transmisión de telemetría a las plataformas de monitoreo. Esto subraya la importancia de la conectividad inalámbrica como un componente esencial en la infraestructura implementada. El análisis del tráfico de la red mediante Kali Linux y Wireshark permitió evaluar su impacto en la conectividad del sistema.

La información recopilada durante las capturas facilitó el estudio del comportamiento de la red antes, durante de los incidentes de ataque, lo que permitió evaluar la respuesta de la infraestructura instalada.

A partir de los resultados obtenidos, se propusieron acciones de mitigación para fortalecer la seguridad de la infraestructura de IoT, tales como mejorar la red Wi-Fi, realizar una vigilancia continua del Gateway, validar la información recibida en Node-

RED y reforzar el sistema de notificaciones. Estas medidas contribuyen a mejorar la disponibilidad, integridad y confiabilidad de las redes LoRaWAN en entornos de IoT.

RECOMENDACIONES

Se recomienda realizar evaluaciones de seguridad de forma regular en la infraestructura LoRaWAN, utilizando herramientas como Wireshark y Kali Linux. Esto ayudará a identificar vulnerabilidades antes de que puedan afectar la conectividad y el correcto funcionamiento del sistema. Además, es fundamental mejorar la seguridad de la red Wi-Fi del Gateway SenseCAP M2. Esto se puede lograr mediante configuraciones adecuadas, el uso de protocolos de seguridad actualizados, contraseñas robustas y actualizaciones frecuentes del firmware de los dispositivos.

Todo esto tiene como objetivo reducir la exposición a amenazas que podrían comprometer la infraestructura inalámbrica. También se sugiere implementar mecanismos adicionales para verificar la información procesada en Node-RED.

Es importante asegurarse de que los datos provenientes de The Things Network sean consistentes antes de almacenarlos o enviarlos a través de Telegram, lo que fortalecerá la integridad de los datos. Por último, se aconseja ampliar las investigaciones futuras para incluir un mayor número de nodos LoRaWAN, explorar diferentes escenarios de comunicación y considerar nuevos tipos de ataques dirigidos a redes LPWAN. Esto permitirá evaluar el comportamiento de estas tecnologías en situaciones más reales y desarrollar más estrategias futuras.

BIBLIOGRAFÍAS

[1] K. A. Cajas Tapia, «Redes de sensores inalámbricos para IoT : automatización de redes inalámbricas de sensores.,» febrero 2022, Quito, 2022.

[2] S. Ricardo Pluas y A. Olaves Rosales, «IMPLEMENTACIÓN DE UN SISTEMA DE ALARMA COMUNITARIA APLICANDO TECNOLOGÍA LPWAN E IOT PARA EL BARRIO 5 DE JUNIO DEL CANTÓN LA LIBERTAD,» 2025, La Libertad,Ecuador, 2025.

[3] A. P. F. Figueroa, «Evaluación de la efectividad en la autenticación y el cifrado de redes de la normativa 802.11 b/g/n analizando las vulnerabilidades y pruebas de penetración en la categoría pentesting,» 2025, La Libertad-Ecuador, 2025.

[4] R. I. Soto La Rosa y J. E. Yagual Laínez, «Desarrollo de un entorno de pruebas controlado para el estudio de redes IoT LoRa implementado en el laboratorio de telecomunicaciones de FACSISTEL,» 2025, La Libertad-Ecuador , 2025.

[5] J. M. Díaz, Artist, *Riesgos y vulnerabilidades de la denegación de servicio distribuidos en internet de las cosas.* [Art]. Universidad El Bosque. Colombia, 2025.

[6] I. Moreno Carre, «Análisis integral de seguridad en dispositivos IoT,» 06-2024, España, 2024.

[7] securitylab, «AURSINC Wi-Fi Deauther: reloj inteligente para comprobar redes Wi-Fi,» 25 June 2024. [En línea]. Available: https://www.securitylab.lat/blog/personal/XiaomiFanES/357715.php.

[8] E. O. De Lima Rosado, Fortalecimiento de la seguridad en el Internet de las cosas (IoT): estrategias y propuestas para mitigar riesgos de vulnerabilidad en sensores de agua con tecnología IoT, santa Marta: 2025, 2025-10-16.

[9] Fortinet, «Definición y explicación IoT (IoT),» 2026. [En línea]. Available: https://www.fortinet.com/lat/resources/cyberglossary/iot.

[10] UIT-T, SERIE Y: INFRAESTRUCTURA MUNDIAL DE LA, Suiza: 2013, 2012.

[11] A. L. rosa, Artist, *LPWAN como base de comunicaciones para IoT.* [Art]. España.

[12] G. c. s. r. LPWAN, Guía completa sobre redes LPWAN, 2023, 2023.

[13] TEKTELICO, «LoRaWAN frente a NB-IoT, Sigfox y LTE: ¿Qué tecnología IoT es la adecuada para usted?,» 2 junio 2025. [En línea]. Available: https://tektelic.com/es/expertise/loRaWAN-vs-nb-iot-sigfox-and-lte-comparison/.

[14] tecnoloblog, «LoRa y LoRaWAN: La tecnología que conecta el futuro del IoT,» 2026. [En línea]. Available: https://www.tecnoloblog.com/que-es-LoRa-y-LoRaWAN/.

[15] innovaciondigital360, Artist, *LoRaWAN: qué es, para qué sirve, por qué utilizarla*. [Art]. innovaciondigital360.

[16] D. Santiago-Cernadas, Artist, *Estudio de la reconexión automática a redes Wi-Fi como puerta de entrada a ataques de suplantación*. [Art]. Universidade da Coruña. Facultade de Informática.

[17] C. A. Amaya Tarazona, Artist, *Modelo para generación de alertas de prevención de ataques TCP/SYN Flooding*. [Art]. Universidad Autónoma de Bucaramanga.

[18] A. D. Vinueza Gualotuña, Artist, *Elaboración de guías de prácticas de laboratorio para la asignatura seguridad de redes empleando Kali Linux*. [Art]. Escuela Politecnica nacional .

[19] U. F. d. P. Santander, Artist, *Análisis de una red en un entorno IPv6: una mirada desde las intrusiones de red y el modelo TCP/IP*. [Art]. Universidad Francisco de Paula Santander.

[20] J. R. Montalvo, Artist, *Sensor de temperatura y humedad DHT11*. [Art].

[21] m. sanchezel, Artist, *Sensores Infrarrojos: Funcionamiento y Aplicaciones*. [Art]. Tegnologico de estudios Superiores de Jocotitlan .

[22] seeedstudi, «Descripción general del SenseCAP M2 Light Gateway,» 2023. [En línea]. Available: https://wiki.seeedstudio.com/es/Network/SenseCAP_Network/SenseCAP_M2_Light_Gateway/SenseCAP_M2_Light_Gateway_Overview/.

[23] M. S. Hernández Espinoza, Artist, *Propuesta de diseño y desarrollo de una red de nodos sensores aplicando tecnología LoRa con ESP32 para monitoreo de temperatura y humedad IOT para un invernadero en la provincia del Guayas.*. [Art]. Universidad Católica de Santiago de Guayaquil.

[24] securitylab., «AURSINC Wi-Fi Deauther: reloj inteligente para comprobar redes Wi-Fi,» 25 June 2024. [En línea]. Available: https://www.securitylab.lat/blog/personal/XiaomiFanES/357715.php.

[25] TheLazyMotobugel, «Introducción a Arduino y su IDE,» 06 2025. [En línea]. Available: https://es.scribd.com/document/836164168/Libro-del-Arduino-docx-20250304-194939-0000.

[26] J. Lopez Vicario, Artist, *Red sensores multiservicio LPWAN, LoRa, LoRaWAN, TTN y MQTT*. [Art]. Universidad Cataluya.

[27] A. A. S. S. Jorge Veloz, Artist, *Ethical hacking, una metodología para descubrir fallas de seguridad en sistemas informáticos mediante la herramienta KALI-LINUX*. [Art].

[28] J. E. ÁLAVA CRUZATTY, Artist, *ANÁLISIS DE TRÁFICO DE DATOS EN LA CAPA DE ENLACE DE REDES LAN, PARA LA DETECCIÓN DE POSIBLES ATAQUE O INTRUCCIONES SOBRE TECNOLOGÍAS ETHERNET Y WI-FI 802.11 EN LA CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES DE LA UNIVERSIDAD ESTATAL DEL SUR DE MANAB*. [Art]. UNESUM.

[29] D. F. J. González, «Estudio monografía: Detección y prevención de ataques en redes Wi-Fi públicas:,» 2024. [En línea]. Available: https://repository.unad.edu.co/bitstream/handle/10596/66814/dfjimenezg.pdf?sequence=3.

ANEXOS: Instalación y Configuración**.**

Anexo: Kali Linux

https://www.kali.org/

Figura 84. *Instalador Kali Linux*

Figura 85. *Archivo de instalación 2026.*

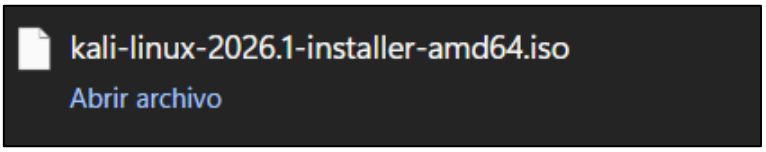


Figura 86. *Selección de idioma.*

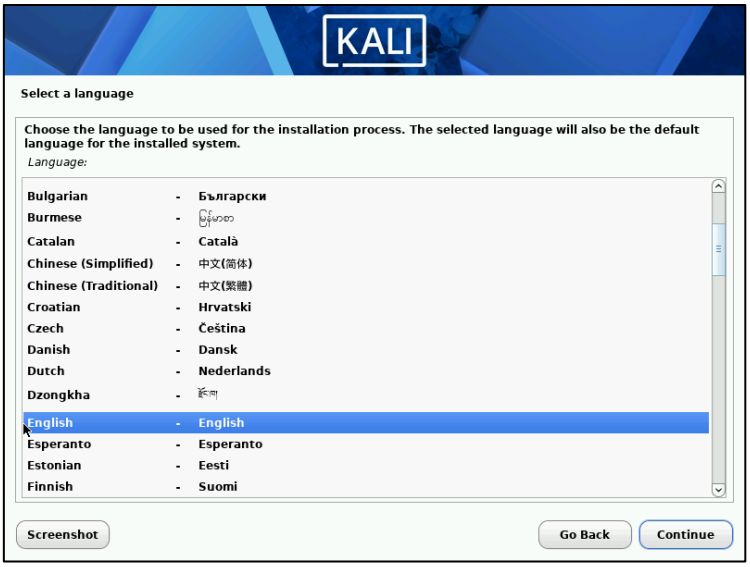


Figura 87. *Nombre de Usuario.*

Figura 88. *Contraseña de ingreso.*

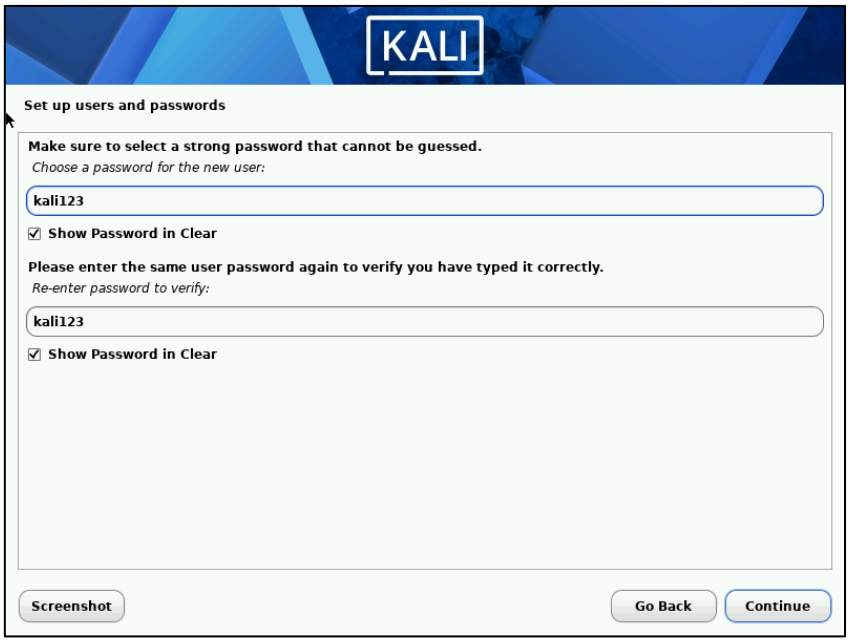


Figura 89. *Software de selección.*

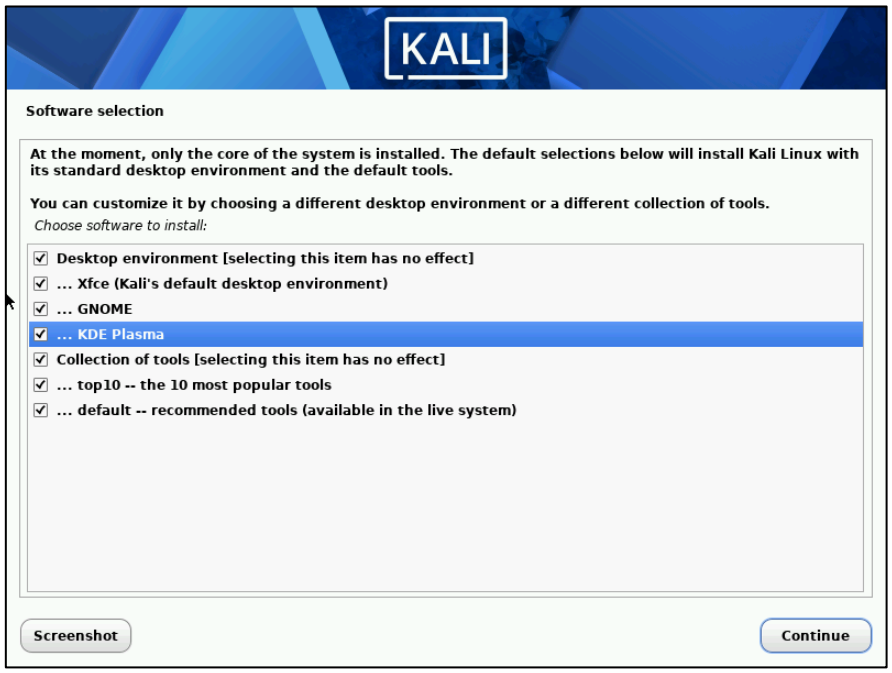
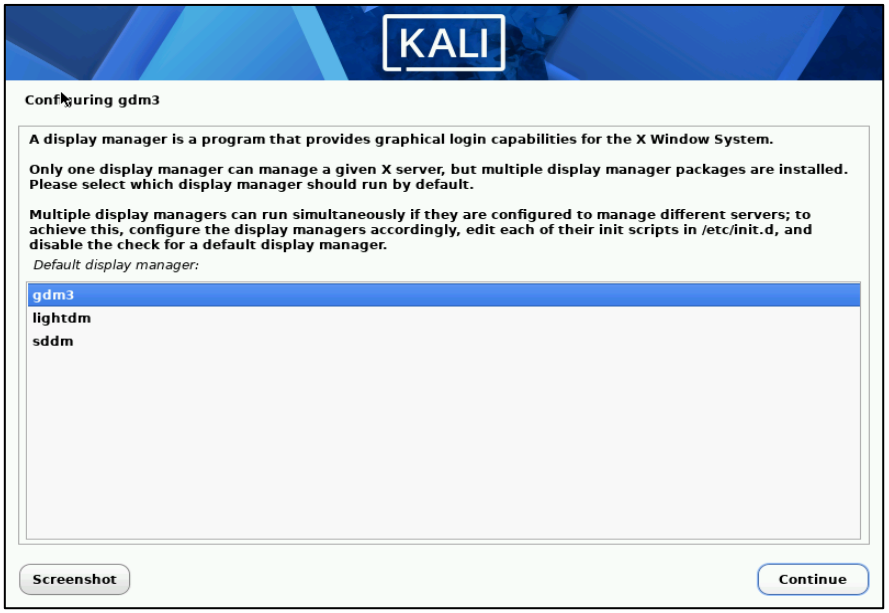


Figura 90. *Configuración gdm3.*

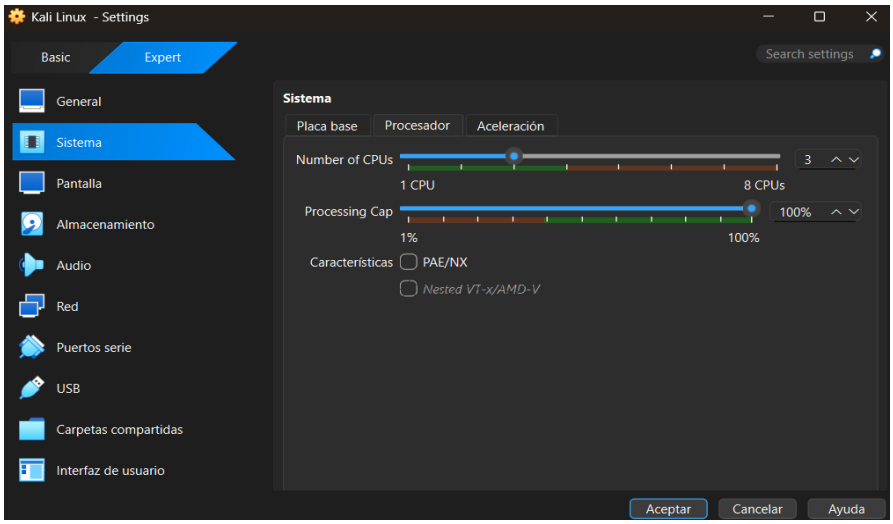


Oracle VirtualBox

Figura 91. *VirtualBox.*

Figura 92. *Hosts de window.*

Figura 93- Configuración del sistema



Anexo: Node-RED

Figura 94. *Instalación para acceso de Node-RED.*

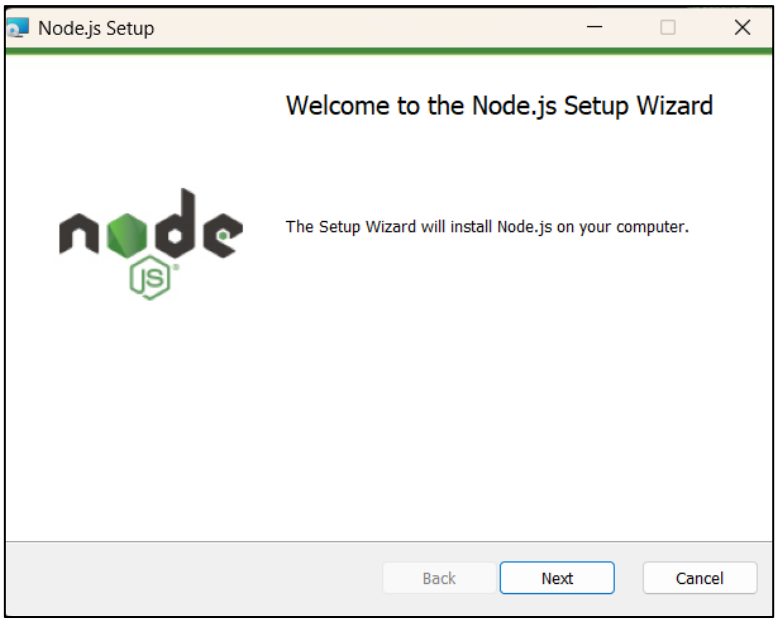


Figura 95. *Acceso a Node-RED.*

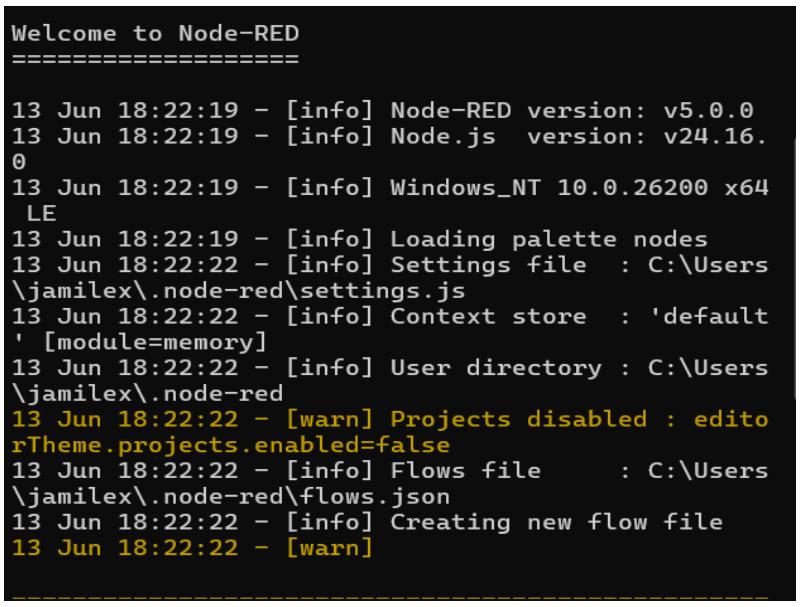


Figura 96. *Instalador de Node-RED para diagrama.*

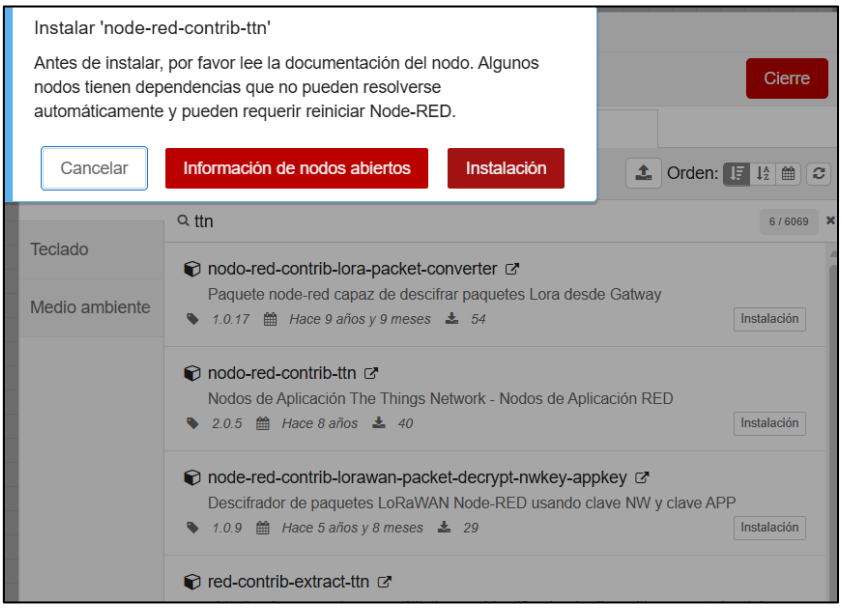


Figura 97. Configuración de Protocolo MQTT.

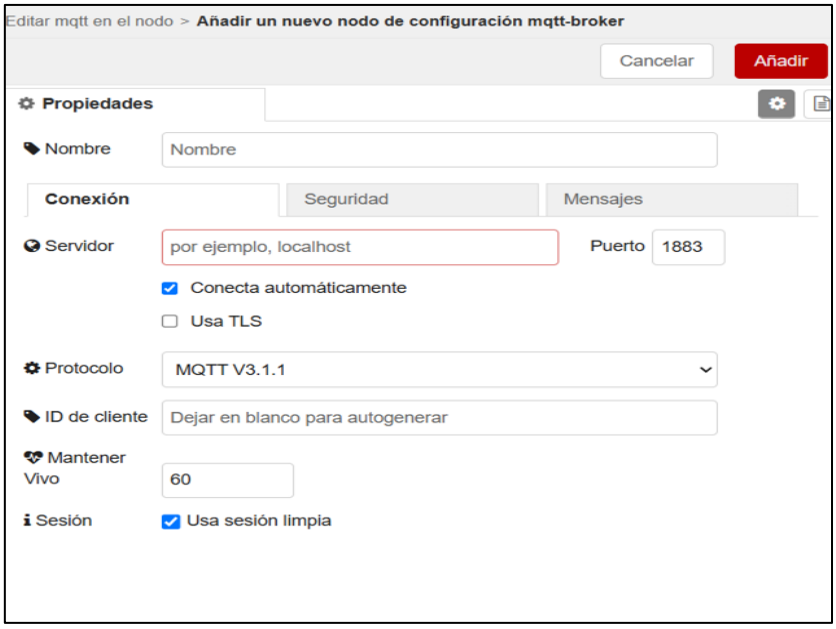


Figura 98. Configuración de Función para temperatura.

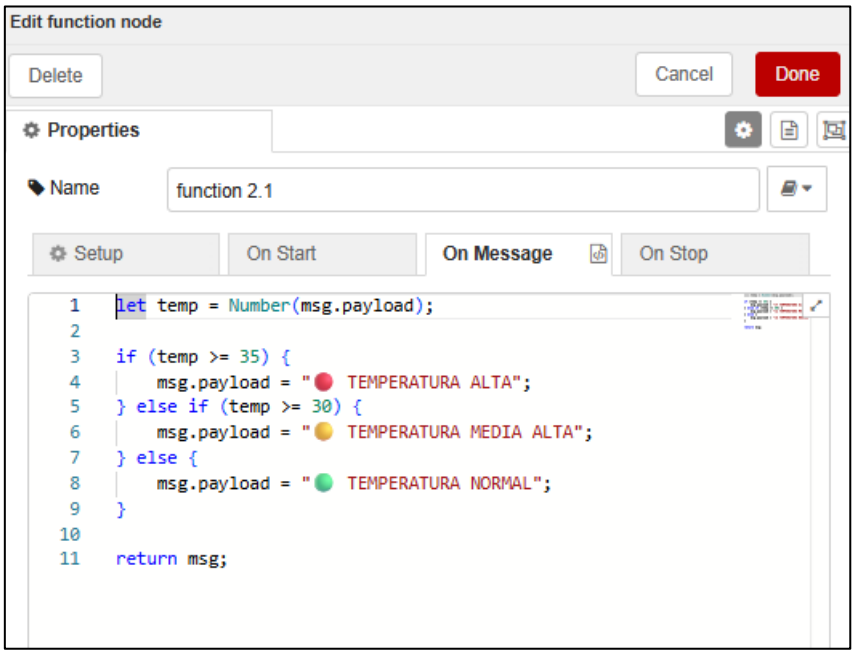


Figura 99 - Configuración del nodo "Historial de Temperatura" (chart)

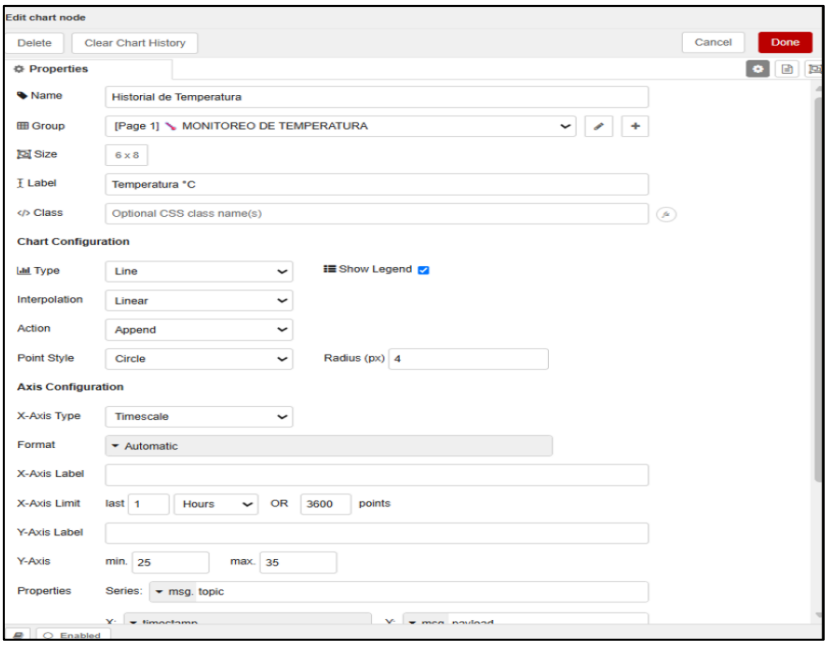


Figura 100. *Configuración del nodo "Temperatura" (gauge).*

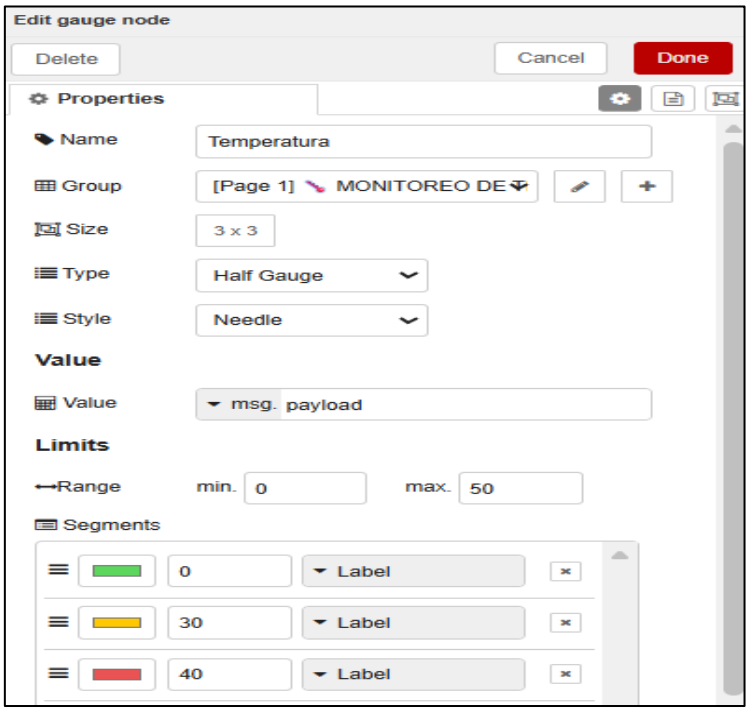


Figura 101. *Configuración del nodo "Alerta de Temperatura" (text)*

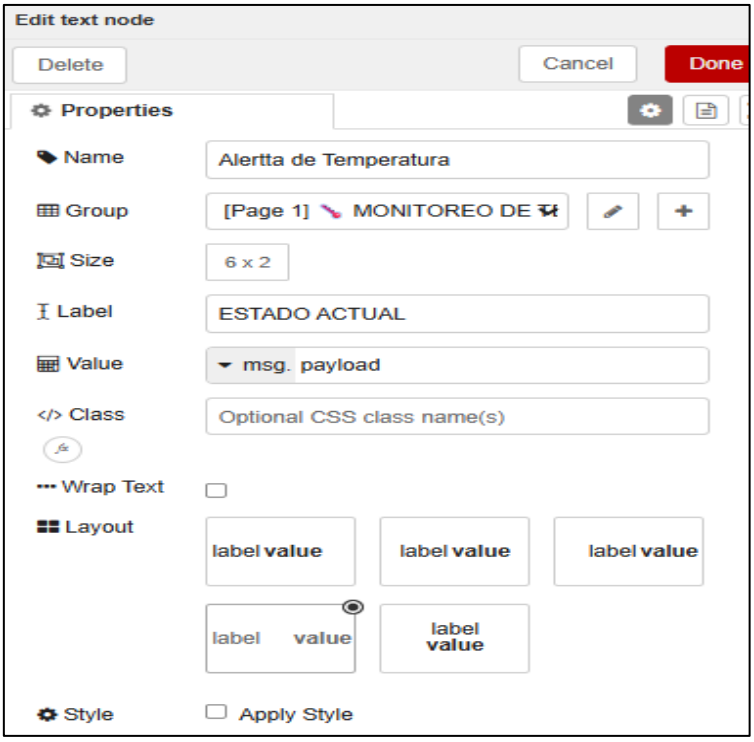


Figura 102. *Configuración de Función para Humedad.*

Figura 103. *Configuración del nodo "Historial de Humedad" (gauge).*

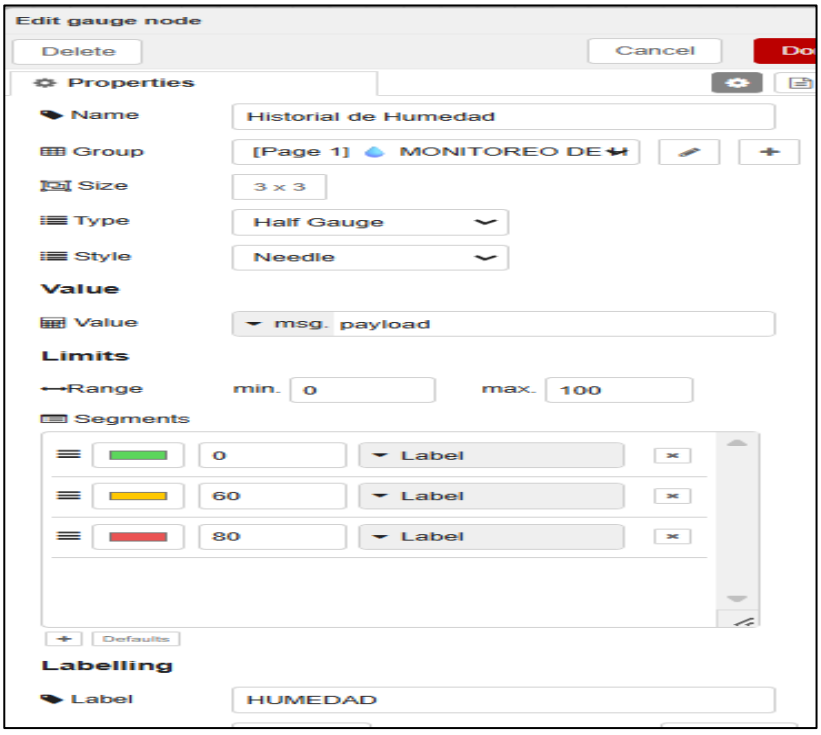


Figura 104. *Configuración del nodo "HUMEDAD" (chart).*

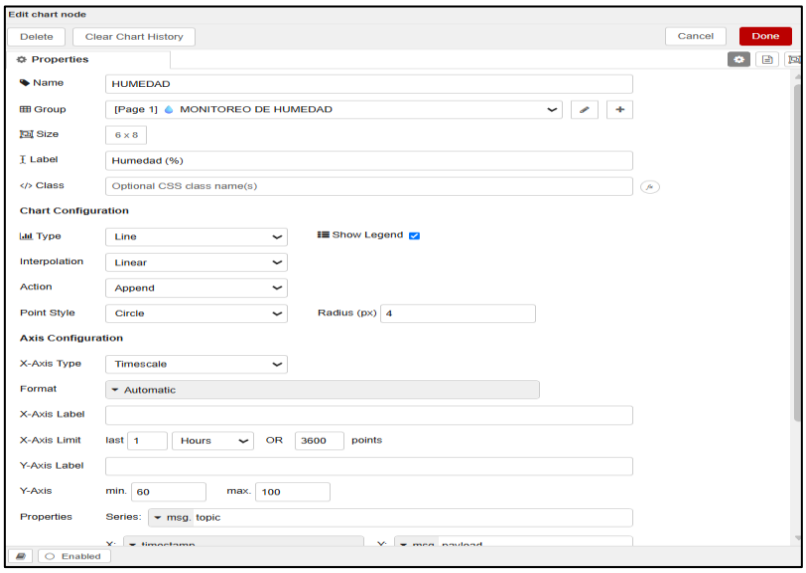


Figura 105. *Configuración del nodo "Alarma de Humedad" (text).*

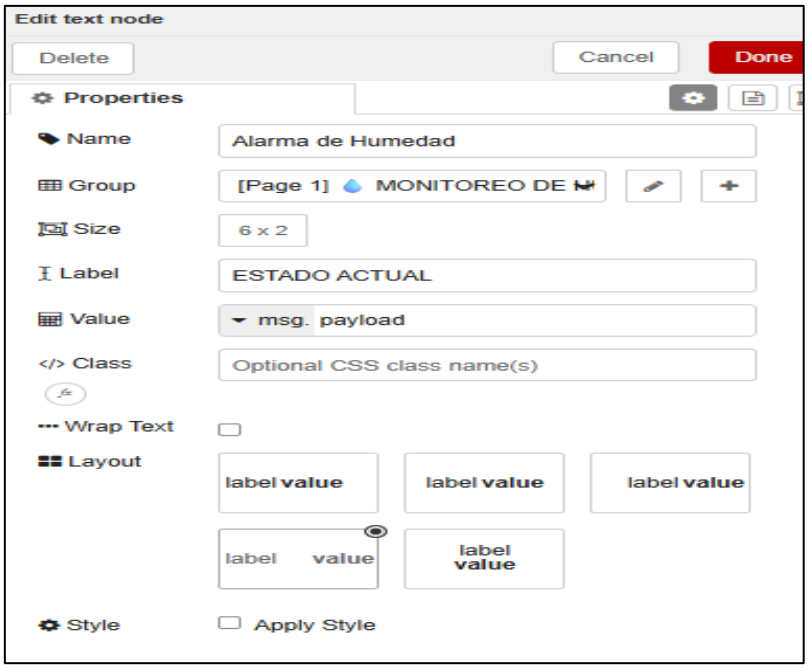


Figura 106. *Configuración de la función de SI.*

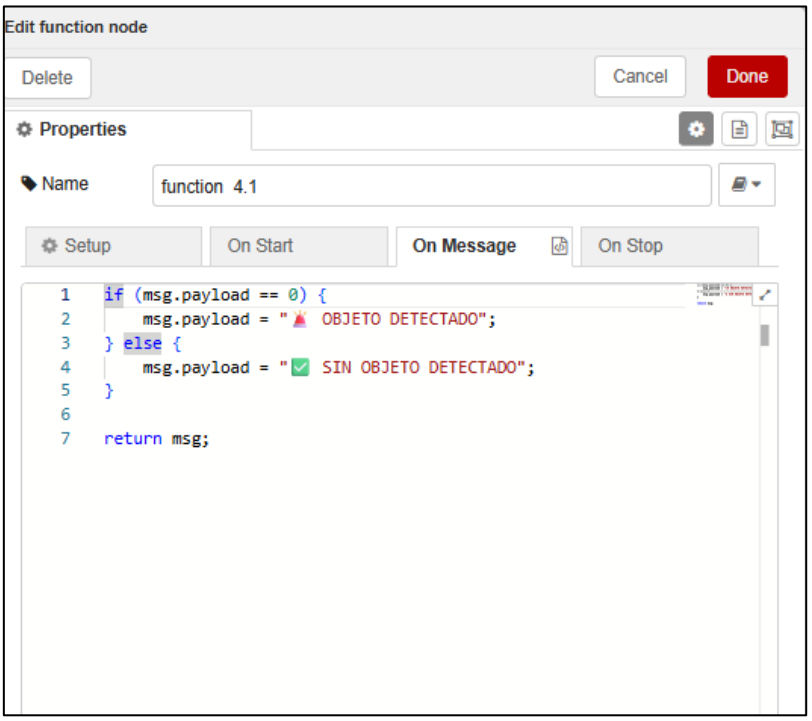


Figura 107. *Configuración del nodo "Alarma SI" (text).*

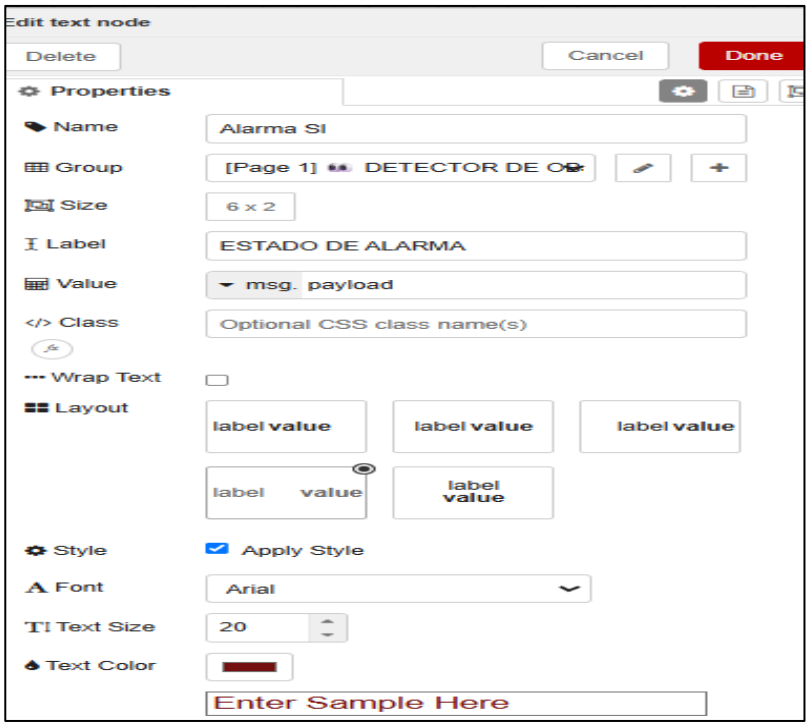


Figura 108. *Configuración del nodo sender de Telegram.*

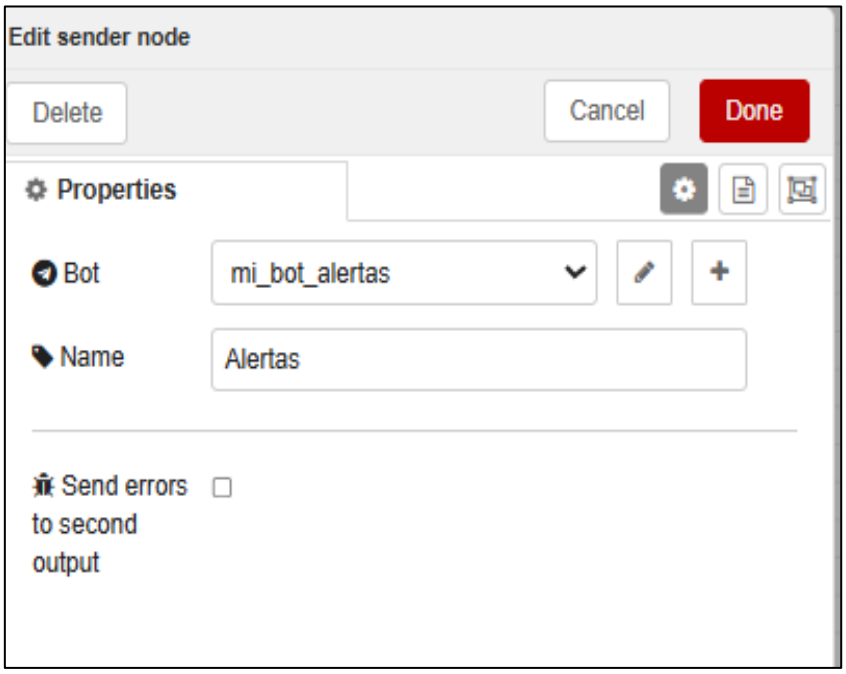
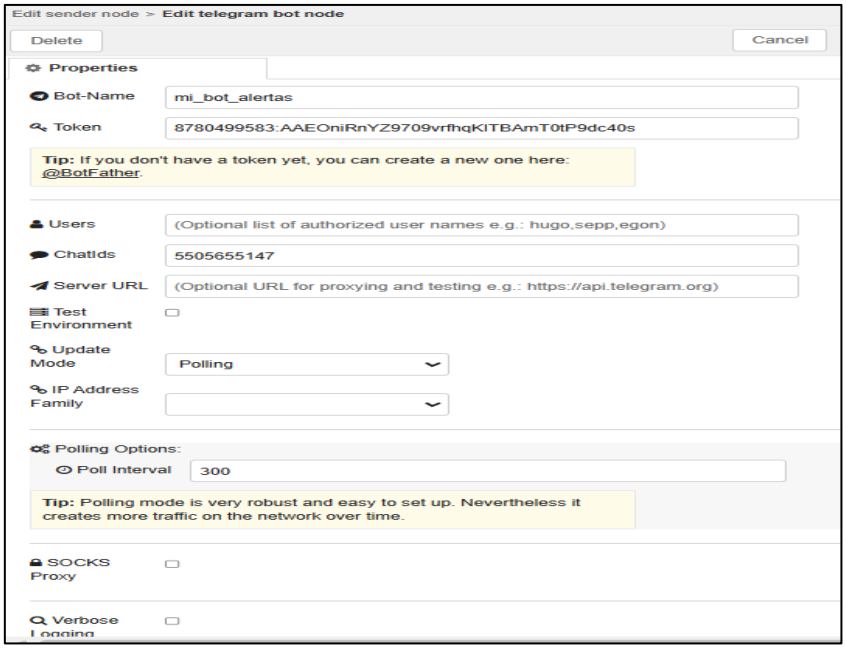


Figura 109. *Configuración de token y chatlds para telegram.*



Anexo: Código de nodo LoRa ESP32**.**

```
#include <RadioLib.h>
#include "DHT.h"
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define DHTPIN 4
#define DHTTYPE DHT11
#define IR_PIN 7
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_SDA 17
#define OLED_SCL 18
#define OLED_RST 21
#define VEXT_CTRL 36
#define LORA_NSS   8
#define LORA_DIO1  14
#define LORA_RST   12
#define LORA_BUSY  13
#define LORA_SCK   9
#define LORA_MISO  11
#define LORA_MOSI  10
DHT dht(DHTPIN, DHTTYPE);
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RST);
SX1262 radio = new Module(LORA_NSS, LORA_DIO1, LORA_RST, LORA_BUSY);
LoRaWANNode node(&radio, &US915, 2);
uint64_t joinEUI = 0x0000000000000000;
uint64_t devEUI  = 0x70B3D57ED00781C6;
uint8_t appKey[] = {
  0x4B, 0xBD, 0x45, 0xF1, 0xD0, 0xE5, 0x5D, 0x40,
  0x0B, 0xF6, 0x0C, 0x2D, 0x04, 0x9A, 0x9A, 0x33
```

```
void mostrarOLED(String l1, String l2, String l3, String l4) {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);  display.println(l1);
  display.setCursor(0, 16); display.println(l2);
  display.setCursor(0, 32); display.println(l3);
  display.setCursor(0, 48); display.println(l4);
  display.display();
}
void setup() {
  Serial.begin(115200);
  delay(2000);
  pinMode(VEXT_CTRL, OUTPUT);
  digitalWrite(VEXT_CTRL, LOW);
  delay(100);
  pinMode(OLED_RST, OUTPUT);
  digitalWrite(OLED_RST, LOW);
  delay(50);
  digitalWrite(OLED_RST, HIGH);
  delay(100);
  Wire.begin(OLED_SDA, OLED_SCL);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  dht.begin();
  pinMode(IR_PIN, INPUT);
  mostrarOLED("Nodo IoT LPWAN", "Iniciando LoRa", "SenseCAP M2", "TTN...");
  Serial.println("Iniciando LoRaWAN...");
  SPI.begin(LORA_SCK, LORA_MISO, LORA_MOSI, LORA_NSS);
  int state = radio.begin(915.0);
if (state != RADIOLIB_ERR_NONE) {
   Serial.print("Error radio: ");
```

```
   Serial.println(state);
   mostrarOLED("ERROR RADIO", String(state), "Revise pines", "LoRa");
   while (true);
  }
  radio.setDio2AsRfSwitch(true);
  Serial.println("Radio OK");
  mostrarOLED("Radio OK", "Uniendo a TTN", "Espere...", "");
  node.beginOTAA(joinEUI, devEUI, nwkKey, appKey);
  node.setDwellTime(false);
  node.setADR(false);
state = node.activateOTAA();
Serial.print("Resultado activateOTAA = ");
Serial.println(state);
if (state == RADIOLIB_ERR_NONE ||
   state == RADIOLIB_LORAWAN_NEW_SESSION) {
   Serial.println("JOIN OK - conectado a TTN");
   mostrarOLED("JOIN OK", "Conectado TTN", "Gateway OK", "");
} else {
   Serial.print("Error JOIN: ");
   Serial.println(state);
   mostrarOLED("ERROR JOIN", String(state), "Revise TTN", "Gateway");
   while (true);
}
}
void loop() {
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
  int ir = digitalRead(IR_PIN);
  if (isnan(temp) || isnan(hum)) {
Serial.println("Error DHT11");
   mostrarOLED("Error DHT11", "Revise sensor", "", "");
delay(5000);
```

```
int temperatura = temp * 10;
int humedad = hum * 10;
uint8_t payload[5];
payload[0] = highByte(temperatura);
payload[1] = lowByte(temperatura);
payload[2] = highByte(humedad);
payload[3] = lowByte(humedad);
payload[4] = ir;
mostrarOLED(
"Enviando a TTN",
"T: " + String(temp) + " C",
"H: " + String(hum) + " %",
"IR: " + String(ir)
);
Serial.print("Enviando -> Temp: ");
Serial.print(temp);
Serial.print(" Hum: ");
Serial.print(hum);
Serial.print(" IR: ");
Serial.println(ir);
int state = node.sendReceive(payload, sizeof(payload), 1);
if (state >= 0 || state == -1116 || state == -1101) {
Serial.println("Uplink enviado a TTN");
Serial.print("Codigo RadioLib: ");
Serial.println(state);
mostrarOLED("Uplink enviado", "TTN Live Data", "Temp/Hum/IR", "");
} else {
Serial.print("Error envio: ");
Serial.println(state);
mostrarOLED("Error envio", String(state), "Revisar TTN", "");
}
delay(15000);
}
```