



**UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE TELECOMUNICACIONES**

TRABAJO DE INTEGRACIÓN CURRICULAR

Previo a la obtención del título de:

INGENIERO EN TELECOMUNICACIONES

Implementación de un sistema de recolección de residuos mediante
la tecnología LPWAN e IoT para el sector de Ballenita

AUTOR:

Yagual Solano Erick Josue

DOCENTE TUTOR:

Ing. Manuel Montaña Blacio. MSc.

LA LIBERTAD - ECUADOR

Año 2025

TRIBUNAL DE SUSTENTACIÓN



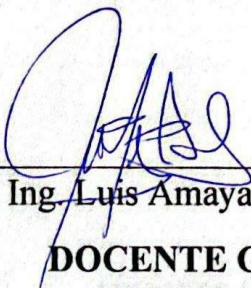
Ing. Ronald Rovira Jurado, PhD.

DIRECTOR DE LA CARRERA



Ing. Manuel Montaña Blacio, MSc.

DOCENTE TUTOR



Ing. Luis Amaya Fariño, Mgtr.

**DOCENTE GUÍA UIC/
ESPECIALISTA**



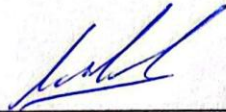
Ing. Corina Gonzabay De La A, Mgtr.

SECRETARIA

APROBACIÓN DE DOCENTE TUTOR

En mi calidad de docente tutor del trabajo de Integración Curricular denominado: **"Implementación de un sistema de recolección de residuo mediante la tecnología LPWAN e IoT para el sector de Ballenita "**, elaborado por **Erick Josue Yagual Solano** estudiantes de la Carrera de Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, previo a la obtención del título de Ingeniería en Telecomunicaciones, me permito declarar que, tras supervisar el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos. En consecuencia, lo considero apto en todos sus aspectos y listo para ser evaluado por el docente especialista

Atentamente,



Ing. Manuel Montaña Blacio. MSc.

DOCENTE TUTOR



**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 **Erick Josue Yagual Solano**, como requerimiento para la obtención del título de Ingeniero en Telecomunicaciones.

La Libertad, a los 30 días del mes de junio del año 2025

TUTOR

A handwritten signature in blue ink, appearing to read "M. Montaña Blacio", is written over a horizontal line.

Ing. Manuel Montaña Blacio, MSc.

APROBACIÓN DE DOCENTE ESPECIALISTA

En mi calidad de docente especialista del trabajo de Integración Curricular denominado: "Implementación de un sistema de recolección de residuos mediante la tecnología LPWAN e IoT para el sector de Ballenita ", elaborado por Erick Josue Yagual Solano estudiantes de la Carrera de Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, previo a la obtención del título de Ingeniería en Telecomunicaciones, me permite declarar que, tras supervisar el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos. En consecuencia, lo considero apto en todos sus aspectos y listo para la sustentación del trabajo.

Atentamente,

A handwritten signature in blue ink, appearing to read 'Luis Amaya Fariño', is written over a horizontal line. The signature is stylized and cursive.

Ing. Luis Amaya Fariño, Mgt

DOCENTE ESPECIALISTA



UPSE

**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA**

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, Erick Josué Yagual Solano

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

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

La Libertad, a los 30 días del mes de junio del año 2025

A handwritten signature in blue ink, which appears to read "Erick Josué Yagual Solano", is written over a horizontal line.

Erick Josué Yagual Solano



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

DECLARACIÓN DE RESPONSABILIDAD

Yo, Erick Josué Yagual Solano

DECLARO QUE:

El trabajo de Titulación, Implementación de un sistema de recolección de residuos mediante la tecnología LPWAN e IoT para el sector de Ballenita, previo a la obtención del título en Ingeniero 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 30 días del mes de junio del año 2025

A handwritten signature in blue ink, which appears to read "Erick Josué Yagual Solano", is written over a horizontal line.

Erick Josué Yagual Solano

AGRADECIMIENTO

En primer lugar, agradezco a Dios por darme la fuerza y la sabiduría para seguir adelante en cada etapa de mi vida.

A toda mi familia, a mi madre por estar siempre conmigo en cada uno de mis pasos como profesional ,quien ha sido mi mayor fuente de inspiración y apoyo, también a mi hermana Cristina Yagual, por ser una inspiración constante en mi vida. Tú esfuerzo, tú dedicación y todo lo que has logrado como profesional me han motivado a seguir adelante con mis propios sueños.

A la Universidad Estatal Península de Santa Elena (UPSE), por darme la oportunidad de formarme profesionalmente y darme los conocimientos y valores necesarios para afrontar el mundo laboral. Muy agradecido con los docentes, con su compromiso y dedicación, que aportaron a mi formación en el ámbito académico y personal.

Agradezco a mi tutor, el Ing. Asdrual Montaña, por su orientación, compromiso y valiosos aportes durante el desarrollo de este trabajo. Su experiencia, sus observaciones oportunas y su disposición fueron fundamentales para el desarrollo y culminación de esta tesis.

También Agradezco al especialista, Ing. Luis Amaya, por su aporte técnico durante el desarrollo de este trabajo. Su experiencia y recomendaciones permitieron mejorar aspectos clave del proyecto , que permitieron mejorar el enfoque técnico de esta tesis.

Agradezco sinceramente a mis amigos de la

universidad, fueron una parte fundamental de este camino. Gracias por su compañerismo, por los momentos compartidos, por el apoyo mutuo en los días difíciles y por las risas que hicieron más llevadero este proceso.

Erick Josué Yagual Solano

DEDICATORIA

Este presente trabajo de titulación va dedicado a mi amada madre, que siempre creyó en mí, por su amor incondicional, por sus sacrificios diarios, por enseñarme a no rendirme, por sus consejos diarios y palabras de aliento todo esto se lo debo a ella.

También dedico este logro a mis hermanas, por su compañía, confianza y palabras de aliento que me brindan cada día .

Y en especial, a la memoria de mi padre, aunque ya no esté en vida, siempre ha estado presente en mi corazón, con la esperanza de que en donde este, se sienta orgulloso de este logro, así como mi mama y mi papa sigue siendo el impulso para alcanzar mis metas.

Erick Josué Yagual Solano

ÍNDICE GENERAL

DECLARACIÓN DE RESPONSABILIDAD	¡Error! Marcador no definido.
TRIBUNAL DE SUSTENTACIÓN.....	¡Error! Marcador no definido.
CERTIFICACIÓN DE ANTIPLAGIO	¡Error! Marcador no definido.
AUTORIZACIÓN.....	¡Error! Marcador no definido.
AGRADECIMIENTO	VIII
DEDICATORIA.....	X
ÍNDICE GENERAL.....	XI
ÍNDICE DE FIGURA	XVI
ÍNDICE DE TABLAS.....	XX
ÍNDICE DE ANEXOS	XX
RESUMEN.....	XXI
ABSTRACT.....	XXII
INTRODUCCIÓN.....	1
CAPITULO I.....	2
1.1. Título	2
1.2. Identificación del problema	2
1.3. Antecedentes investigativos.....	3
1.4. Descripción del proyecto.....	5
1.5. Objetivo del proyecto	6
1.5.1. Objetivo General.....	6

1.5.2. Objetivos Específicos	6
1.6. Justificación	7
1.7. Alcance	8
CAPÍTULO II	9
2.1. Marco contextual	9
2.2. Marco Teórico	9
2.2.1. Residuos sólidos urbanos (RSU)	9
2.2.1.1. Caracterización de los residuos sólidos urbanos	10
2.2.1.2. Generación y recolección de residuos sólidos	10
2.2.2. Internet de las Cosas (IoT)	11
2.2.3. Aplicaciones del IoT en la Gestión de Residuos	11
2.2.4. Sensores IoT en la Recolección de Residuos	12
2.2.5. Arquitecturas para Sistemas IoT en la Gestión de Residuos	14
2.2.6. Protocolo de la capa de aplicación	15
2.2.6.1. Protocolo MQTT	15
2.2.6.2. Protocolo HTTP/HTTPS	16
2.2.7. Definición de LPWAN	17
2.2.8. Tecnologías LPWAN	18
2.2.8.1. LoRa	19
2.2.8.2. Sigfox	19
2.2.8.3. NB-IoT	20

2.2.8.4.	LTE-M	21
2.2.9.	Servidores LoRaWAN.....	23
2.2.10.	LoRa en la gestión de residuos.....	25
2.2.10.1.	Principales Características de LoRa.....	26
2.2.10.2.	Arquitectura de la red LoRa	26
2.2.11.	Tecnología Sigfox.....	27
2.2.11.1.	Características principales de la tecnología Sigfox.....	28
2.2.11.2.	Arquitectura de sistema Sigfox	28
2.2.12.	LoRaWAN.....	30
2.2.12.1.	Clases de nodos.....	30
2.2.12.2.	Módulo de transmisión LoRaWAN	32
2.2.13.	Algoritmos de rutas	33
2.2.13.1.	Algoritmo de Dijkstra	33
2.2.14.	Plataforma de mapeo de ruta.....	33
2.2.14.1.	Algoritmo de Bellman-Ford	33
2.2.14.2.	API de Roads	34
2.2.14.3.	Mapquest	35
CAPITULO III.....		37
3.1.	Componentes de la propuesta.....	37
3.1.1.	Componentes físicos.....	37
3.1.2.	Componentes lógicos.....	42

3.2. Diseño de la propuesta	43
3.2.1. Diseño y conexión de los componentes del sistema de monitoreo	43
3.2.2. Diseño eléctrico	45
3.3. Configuración de los dispositivos	48
3.3.1. Configuración del Gateway SenseCAP m2	48
Configuración de la red de puerta de enlace	48
3.3.2. Configuración de dispositivos LoRaWAN para la transmisión de datos.....	53
3.3.3. Configuración del dispositivo Gateway en The Things Network (TTN).....	54
3.3.4. Registro del Heltec AB-02 con TTN	57
3.3.5. Configuración y transmisión del Heltec AB-02 con TTN	59
3.3.6. Despliegue de la plataforma Node-Red para la gestión y visualización integral de datos. 61	61
3.3.7. Integración de Node-RED con Telegram para el envío de notificaciones.....	67
3.3.8. Integración y visualización de datos en el nodo Worldmap de Node-RED	72
3.4. Diagrama de flujo completo implementado en Node-Red	75
CAPÍTULO IV	77
4.1. Análisis de resultados	77
4.2. Prototipo del sistema de monitoreo de nivel de llenado	77
4.3. Pruebas de transmisión de datos a TTN	78
4.4. Resultados obtenidos en la calibración del sensor de nivel de llenado	84
4.4.1. Resultados obtenidos con el sensor ultrasónico JSN-SR04T	84

4.5. Análisis de datos de monitoreo y optimización de rutas de recolección	86
4.5.1. Detección y registro del nivel de llenado en contenedores inteligentes	86
4.5.2. Determinación del recorrido óptimo	89
4.6. Sistema de notificación de alertas mediante Telegram en la gestión de residuos	91
4.7. Visualización de datos de llenado en la consola de depuración.....	92
CONCLUSIONES.....	94
RECOMENDACIONES.....	95
BIBLIOGRAFÍA.....	96
ANEXOS.....	104

ÍNDICE DE FIGURA

Figura 1 Capas de arquitectura	14
Figura 2 Arquitectura de la red Lora	27
Figura 3 Arquitectura del SigFox	29
Figura 4 Protocolo Lora.....	30
Figura 5 Algoritmo de Dijkstra	33
Figura 6 Algoritmo de Bellman-Ford.....	33
Figura 7 API de Roads.....	35
Figura 8. Gateway Sensencap M2[41].....	38
Figura 9.CubeCell HTCC AB-02[42].....	40
Figura 10. Sensor JSN-SR04T[43]	41
Figura 11. Batería de litio 3.7v 1500mAh[44].....	42
Figura 12 Diseño y conexión de los componentes del sistema de monitoreo	44
Figura 13 Diagrama eléctrico	48
Figura 14.Esquema de conexión del gateway SenseCAP[45]	49
Figura 15 Detalle de la Etiqueta del SenseCAP M2	50
Figura 16 Inicio de Sesión en la Interfaz de Configuración del SenseCAP	50
Figura 17 Conexión del Gateway SenseCAP por red wifi	51
Figura 18 Parámetros de Conexión Inalámbrica en el SenseCAP.....	51
Figura 19 Interfaz de Escaneo de Redes Wi-Fi en el SenseCAP.....	52
Figura 20 Configuración de datos de la red Wifi.....	52
Figura 21 Vinculación del Gateway a una red Wi-Fi.....	53
Figura 22 Registro en The Things Network	53

Figura 23 Configuración del dispositivo Gateway en The Things Network	54
Figura 24 Configuración de la puerta de enlace	55
Figura 25 Configuración de la puerta de enlace	55
Figura 26 Registro el Gateway en TTN.....	56
Figura 27 Configuración Avanzada del Gateway: Etapa de Registro y Parámetros Específicos	56
Figura 28 visualización de datos en la plataforma TTN.....	57
Figura 29 Registro del módulo CubecCell HTCC AB-02.....	57
Figura 30 Registro de Dispositivos Finales en The Things Network.....	58
Figura 31 Configuración de Dispositivos Finales en The Things Network: Asignación de Identificadores y Claves.....	58
Figura 32 Interfaz de Configuración de Preferencias en Arduino IDE	59
Figura 33 Código de Inicialización de Dispositivo LoRa.....	60
Figura 34 Información de Activación y Sesión para el Dispositivo CubeCell...	61
Figura 35 Configuración de Conexión MQTT en The Things Network	62
Figura 36 Integración de MQTT en Node-RED para Procesamiento de Datos IoT	63
Figura 37 Configuración de Propiedades del Nodo MQTT en Node-RED.....	64
Figura 38 Edición de Parámetros del Nodo MQTT Broker para Conexión con The Things Network	64
Figura 39 Asignación de Usuario y Contraseña en el Nodo MQTT Broker	65
Figura 40 Configuración del Topic MQTT para la Comunicación Ascendente en TTN	66
Figura 41 Visualización de datos en Node-RED mediante MQTT	66

Figura 42 Comandos de Configuración de Bots en Telegram mediante BotFather68	
Figura 43 Configuración del Nombre y Token de Acceso para un Bot de Telegram	
.....	69
Figura 44 Instalación del paquete de Telegram en Node-RED	70
Figura 45 Configuración del nodo Telegram Sender.....	70
Figura 46 Diagrama de Flujo en Node-RED: Transmisión de Datos IoT a Telegram	
.....	71
Figura 47 Envío de Notificaciones de Alerta a Telegram	72
Figura 48 Instalación del nodo Worldmap	73
Figura 49 Integración del nodo worldmap en el diagrama de flujo.....	73
Figura 50 Personalización de parámetros de mapa en Node-RED para visualización geográfica.....	74
Figura 51 Visualización de Datos en la Plataforma de Worldmap.....	75
Figura 52 Diagrama de flujo completo implementado en Node-Red.....	76
Figura 53. Prototipo de Sistema de Nivel de Llenado Basado en LoRa.....	77
Figura 54. Transmisión de Datos por LoRaWAN desde CubeCell HTCC-AB0279	
Figura 55. Registro de Datos en TTN.....	80
Figura 56. Visualización del tráfico de paquetes LoRa en el Gateway	81
Figura 57. Recepción de paquetes en la consola de The Things Network	81
Figura 58. Monitor Serie: Registro de Datos de Nivel de Contenedor.....	82
Figura 59. Función de Decodificación de Payload para la medición del nivel de llenado.....	83
Figura 60. Monitorización del Estado de Contenedores en Tiempo Real	84
Figura 61. Resultado de simulación de un contenedor vacío de 1 m.....	85

Figura 62. Resultado de simulación de un contenedor lleno	86
Figura 63. Visualización Geográfica del Contenedor 1 con Nivel de Llenado en 0%	87
Figura 64. Visualización Geográfica del Contenedor 1 con Nivel de Llenado en 96%	87
Figura 65. Visualización Geográfica del Contenedor 2 con Nivel de Llenado en 0%	88
Figura 66. Visualización Geográfica del Contenedor 2 con Nivel de Llenado en 95%	88
Figura 67. Generación de la ruta óptima del contenedor 1	89
Figura 68. Generación de la ruta óptima del contenedor 2	90
Figura 69. Generación de la ruta óptima de ambos contenedores	91
Figura 70. Notificación sobre contenedores llenos en Telegram.....	92
Figura 71. Registro de Depuración en Node-RED	93

ÍNDICE DE TABLAS

Tabla 1 Tipos de sensores IoT	12
Tabla 2 Característica de la tecnología LPWAN.....	22
Tabla 3 Modulo LoRaWan	32
Tabla 4 Diferencias entre el Dijkstra y Bellman Ford	34
Tabla 5 Características principales del Gateway Sensecap M2.....	38
Tabla 6 Características principales del Cubecell HTCC-AB02.....	39
Tabla 7 Características principales del Sensor JSN-SR04T	41
Tabla 8 Conexiones del sensor JSN-SR04T	48
Tabla 9 Librerías Empleadas en Dispositivos CubeCell con LoRaWAN	59
Tabla 10. Consumo total de corriente del prototipo propuesto.....	78

ÍNDICE DE ANEXOS

Anexo 1 Código Arduino ID para Medición de Nivel de Llenado y Transmisión vía LoRaWAN	104
Anexo 2 Implementación del flujo en Node-RED mediante archivo .json para la visualización y monitoreo de contenedores inteligentes	108

RESUMEN

El presente trabajo presenta una solución para mejorar la gestión de los desechos sólidos mediante la implementación de un sistema de supervisión remota en tiempo real. Actualmente, la recolección de residuos se lleva a cabo de manera fija, sin tener en cuenta el estado real de los contenedores. Esto provoca un uso ineficiente de los recursos, especialmente cuando los contenedores están vacíos o con poca carga, y puede causar problemas ambientales si se desbordan. El proyecto desarrollado utiliza sensores ultrasónicos JSN-SR04T, módulos CubeCell HTCC-AB02 y un Gateway SenseCAP M2 que, todos están configurados para funcionar dentro de una red LoRaWAN en la plataforma The Things Network (TTN). La recopilación y monitoreo de datos se realizan a través de Node-RED, el mismo, que permite integrar un sistema de notificaciones Telegram que informa inmediatamente cuando los contenedores llegan a su máximo de llenado. La principal ventaja de esta solución es que se puede operar de forma independiente sin depender de la red eléctrica y a bajo precio, este sistema demuestra que las tecnologías LPWAN e IoT son viables en entornos urbanos, convirtiéndose en una solución escalable y sostenible para la gestión de residuos en comunidades similares.

Palabras clave: LoRaWAN , IoT , monitoreo, LPWAN, residuos.

ABSTRACT

This paper presents a solution to improve solid waste management by implementing a real-time remote monitoring system. Currently, waste collection is carried out in a fixed manner, without considering the actual status of the containers. This results in inefficient use of resources, especially when containers are empty or underfilled, and can cause environmental problems if they overflow. The developed project uses JSN-SR04T ultrasonic sensors, HTCC-AB02 CubeCell modules, and a SenseCAP M2 Gateway, all configured to operate within a LoRaWAN network on The Things Network (TTN) platform. Data collection and monitoring are performed through Node-RED, which allows the integration of a Telegram notification system that immediately informs when containers reach their maximum fill level. The main advantage of this solution is that it can operate independently, without relying on the power grid, and at a low cost. This system demonstrates that LPWAN and IoT technologies are viable in urban environments, becoming a scalable and sustainable solution for waste management in similar communities.

Keywords: LoRaWAN, IoT, monitoring, LPWAN, waste.

INTRODUCCIÓN

Actualmente, la gestión eficiente de los residuos sólidos urbanos es uno de los mayores desafíos que enfrentan las ciudades en crecimiento, especialmente en aquellas con un alto volumen de turismo. El crecimiento de la población y el aumento del turismo han generado una mayor necesidad de servicios de recolección de basura. Estos servicios se enfrentan a problemas como rutas poco frecuentes, contenedores llenos y la falta de información actualizada sobre su estado. De manera negativa afectan tanto al medio ambiente como la calidad de vida de los habitantes y visitantes[1].

El desarrollo de tecnologías emergentes como Internet de las cosas (IoT) y Low Power Wide Area Networks (LPWAN) han puesto a disposición nuevas oportunidades, así como la posibilidad de monitorizar los procesos de recolección de residuos, haciendo uso de contenedores provistos de sensores que son capaces de recopilar información sobre su estado (nivel de llenado). A partir de la información proveniente del estado de los contenedores, es posible generar rutas más eficientes que permitan reducir el uso de recursos o la planificación operativa de los procesos de recolección.

La presente tesis se centra en la creación y ejecución de un prototipo efectivo de IoT y LPWAN basado en un sistema de recolección de residuos. El prototipo podría ser formado con sensores ultrasónicos, módulos de comunicación LoRa, y una plataforma de tiempo real para visualización y organización de información, basada en Node-RED. A partir de ello, se describe la lógica de implementación del sistema.

Esta nueva técnica mejora la recogida de residuos en lugares turísticos reduciendo contaminación y apoyando un plan sustentable; eso muestra que la tecnología puede cambiar mucho los manejos de desechos.

CAPITULO I

1.1. Titulo

Implementación de un sistema de recolección de residuos mediante la tecnología LPWAN e IOT para el sector de ballenita.

1.2. Identificación del problema

En Ballenita, una comunidad costera con una población en crecimiento y un turismo en aumento, la gestión de residuos se ha convertido en un gran desafío. En la actualidad el sistema recolección de residuos enfrenta problemas como la ineficiencia en las rutas de recolección, la sobrecarga de los contenedores, y la falta de información precisa sobre la clasificación de residuos. Estos problemas no solo afectan a la limpieza e imagen del lugar, sino que también provocan contaminación ambiental y reducen la calidad de vida de los residentes y turista[2].

Los materiales y objetos que han cumplido el propósito para lo que fueron creados y ya no tienen valor económico para el usuario son desechados en forma de residuos. Dado que estos residuos tienden a acumularse en grandes cantidades en diversos espacios y áreas comunes como la playa, parques, mercado y en calle general. Por ello se implementa un sistema de recolección de residuos para gestionar, recolectar y eliminar dichos residuos[3].

Con el fin de gestionar los residuos sólidos de manera eficiente, es crucial para mejorar la salud pública y la calidad de vida en zonas urbanas y costeras. En Ecuador, estudios han demostrado que la acumulación de residuos ocasiona problemas de salud como la propagación de enfermedades infecciosas. Un sistema eficiente no solo mejora el entorno urbano, sino que también promueve una mayor conciencia comunitaria sobre la importancia del reciclaje y la reducción de residuos [4].

En la actualidad las rutas de recolección de residuos sólido enfrentan numerosos desafíos que conllevan a que los vehículos recolectores recorran largas distancias generando así costos adicionales en transporte, combustible, desgaste de neumáticos y mantenimiento, entre otros aspectos, además se genera la pérdida de tiempo, es perjudicial para la eficacia del servicio. También no cuentan con horarios específicos para la recolección de residuos, se realiza de forma aleatoria tanto en las rutas seguidas como en las horas del día[5].

1.3. Antecedentes investigativos

El crecimiento económico mundial está asociado con el incremento de la población especialmente en los centros urbanos lo que provoca un aumento en la generación de residuos sólidos urbanos por el consumo social, actividades productivas y componentes ambientales. Por lo tanto, las ciudades ven la necesidad de adaptar y ajustar sus sistemas de recolección de residuos en función del crecimiento de la población y capacidad para cubrir al menos la producción anual promedio de los residentes[6].

En el proyecto llamado Optimización de recolección de residuos con IoT [7], se centra en la evaluación de cómo se puede mejorar la recolección de residuos en contenedores de espacio públicos utilizando la tecnología del Internet de las cosas (IoT). Se ha desarrollado un sistema web que facilita visualizar las rutas de recogida de residuos más eficientes y facilitar la generación de informes detallados de cada contenedor, utilizando datos obtenidos a través de sensores. El enfoque del proyecto permite apreciar la factibilidad de una red de sensores instalados en contenedores de basura para optimizar la recolección. Queda demostrado también que la Internet of Things servirá en el futuro para optimizar procesos de gran escala y colaborar con la gestión de recursos en todo tipo de operaciones.

En el proyecto llamado modelo para la gestión de residuos aprovechables

incorporando tecnología IoT y servicio en la nube [8]. Proponen un método innovador para la administración de residuos utilizando tecnología IoT y servidores basados en la nube. Esta solución ofrece a la empresa supervisar en tiempo real la cantidad de residuos recolectados en sus contenedores a través de dispositivos electrónicos. Además, se encarga de generar rutas de recolección dinámicas que son asignadas a los colaboradores del sistema de recolección local. En este proyecto se logró estimar la cantidad de residuos que se producen y generar rutas de recolección eficientes desde el punto de vista de distancia y tiempo.

En [9] llamado Waste Management System Using IoT-Based Machine Learning in University, este trabajo se propone un método que logra una gestión de residuos de manera vigorosa y eficiente mediante la predicción de la probabilidad del nivel de residuos en los contenedores de basura. El sistema puede optimizar la recolección de desechos utilizando la teoría de grafos y el aprendizaje automático. Este artículo presenta un caso de investigación que se llevó a cabo en el campus real de la Universidad Ton Duc Thang (Vietnam) para evaluar el desempeño y la viabilidad de implementar el sistema. El sistema identifica la mejor ruta de gestión de la recogida de desechos, lo que ahorra tiempo.

En [10] llamado Implementation of an smart waste management system using IoT, se propone un sistema que emplea la tecnología de Internet de las Cosas (IoT) para mejorar la gestión de residuos. Este sistema integra una base de datos en SQL para cada contenedor de basura, se conecta a una aplicación móvil asociada a un contenedor inteligente. El propósito de este sistema es reducir la necesidad de intervención humana y apoyar el desarrollo de ciudades más inteligentes y eficientes. Se propone reemplazar los contenedores de basura tradicionales por estos contenedores inteligentes, podría mejorar la gestión de residuos urbanos al evitar la acumulación de basura en la vía pública.

El artículo A LoRaWAN IoT-enabled trash bin level monitoring system [11], se

resalta la necesidad de establecer un sistema de monitoreo en tiempo real capaz de proporcionar datos sobre la cantidad de basura en los contenedores tanto a municipios como a empresas de gestión de residuos. Con el propósito de optimizar la gestión de residuos sólidos municipales, el artículo expone el diseño y la validación de un sistema basado en la tecnología del Internet de las Cosas (IoT). Este sistema incorpora nodos IoT en cada contenedor necesitado de seguimiento, que registran tanto su nivel de llenado como su posición geográfica. Estos nodos recolectan información y la envían mediante una red LoRaWAN a un servidor dedicado al monitoreo de niveles de basura mediante IoT. Desde este servidor, se puede acceder y analizar la situación y localización de cada contenedor a través de una interfaz gráfica de usuario avanzada.

1.4.Descripción del proyecto

La presente tesis tiene como objetivo principal realizar un prototipo de un sistema de recolección de residuos mediante la tecnología LPWAN e IoT, la investigación se centra debido a su creciente desarrollo turístico y poblacional, enfrenta desafíos significativos relacionados con la gestión eficiente de residuos.

Aborda la necesidad de adoptar tecnologías avanzadas, como LPWAN e IoT, para mejorar la gestión de residuos y contribuir a la sostenibilidad ambiental. Además, la falta de implementación de estas tecnologías innovadoras.

El proyecto abarca el diseño y prototipo del sistema propuesto, el cual incluye la selección e instalación de sensores y el desarrollo de un algoritmo de aprendizaje de la ruta óptima. Este algoritmo utiliza como principales parámetros la ubicación predefinida en el sistema y el nivel de llenado de cada contenedor, obtenido en tiempo real. Se realizarán pruebas piloto para evaluar la eficacia y viabilidad del sistema de recolección de residuos.

Los resultados obtenidos se presentan y se analizan en la sección correspondiente, destacando la mejora significativa en la recolección de residuos y una optimización en la eficiencia de las rutas de recolección basándose en la información recibida en tiempo real, lo cual permite a los camiones de basura operar de manera eficiente. Estos resultados demostraron que la implementación de tecnologías de LPWAN e IoT en la gestión de residuos no solo es viable, sino que también servirá como modelo replicable para otras comunidades con desafíos similares, promoviendo una gestión de residuos más sostenible y tecnológicamente avanzada.

1.5. Objetivo del proyecto

1.5.1. Objetivo General

Desarrollar e implementar un sistema basado en Internet de las cosas (IoT) para mejorar la eficiencia y la gestión de recolección de residuos en Ballenita, con el fin de contribuir a la preservación del medio ambiente.

1.5.2. Objetivos Específicos

1. Realizar un estado de arte de las tecnologías IoT más adecuadas para la implementación del sistema de recolección de residuos, considerando factores como la conectividad y la eficiencia energética.
2. Crear una arquitectura de red IoT que permita definir la estructura de transmisión de datos entre los sensores.
3. Diseñar los nodos sensores IoT para la detección automática de niveles de llenado de contenedores de residuos.
4. Configurar una plataforma de gestión basada en la nube para el monitoreo remoto y la visualización de los datos recopilados por los sensores,

permitiendo una supervisión en tiempo real del estado de los contenedores de residuos.

5. Realizar las pruebas de campo necesarias para validar el correcto funcionamiento del sistema de recolección de residuos.

1.6. Justificación

La implementación de un sistema de recolección de desechos utilizando tecnologías LPWAN e IoT en el sector de Ballenita promete tener un gran impacto tanto a nivel social como ambiental, además de mejorar la eficiencia y la gestión de los recursos. Este método innovador podría ayudar a la comunidad de Ballenita a resolver una variedad de problemas relacionados con la gestión de desechos.

En el ámbito tecnológico, la adopción de tecnologías LPWAN e IoT para mejorar la recolección de residuos en Ballenita es adecuada. Estas tecnologías permiten la conexión eficiente de dispositivos inteligentes, como sensores en contenedores de basura, transmitiendo datos relevantes en tiempo real a una plataforma centralizada. Esto facilita la supervisión y optimización de los procesos operativos, generando beneficios económicos y ambientales significativos.

La implementación de un sistema de recolección de residuos utilizando la tecnología LPWAN e IoT podría resultar en una reducción significativa de los costos operativos. El uso de sensores inteligentes en los contenedores permite optimizar las rutas de recolección. Los costos asociados con la limpieza y mantenimiento de áreas públicas se ven compensados rápidamente gracias a esta eficiencia mejorada.

Este proyecto ofrece una solución que permite enviar cantidades muy pequeñas de datos a larga distancia, con total autonomía a la red eléctrica y de bajo costo, y esto es posible

gracias a los últimos avances en tecnología de comunicaciones, como la nueva tecnología de conexión inalámbrica LoRa, óptima para casos de uso que requieren conexiones de larga distancia, menor consumo y nodos de sensores inalámbricos distribuido. Además, su fácil implementación y gestión permite alimentar datos a la plataforma que finalmente genera una ruta óptima para el servicio de recolección de basura.

1.7. Alcance

Esta tesis aborda el diseño, desarrollo y prototipo de un sistema para la recolección de residuos ,utilizando tecnologías de comunicación de amplio alcance y bajo consumo de energía e Internet de las cosas. Se llevará a cabo un prototipo utilizando sensores ultrasónicos para medir el nivel de llenado de los contenedores de basura en sector de ballenita, los datos obtenidos para el intercambio de información se analizarán utilizando software de aprendizaje, algoritmos de determinación de rutas y otros componentes. El sistema de transmisión enviara un mensaje con la información del nivel de cada contenedor hacia la nube. Estos datos serán esenciales para las decisiones que tomaremos en nuestro algoritmo de optimización de rutas, que utiliza la ubicación registrada en el sistema y el nivel de llenado que se encuentra en tiempo real para cada contenedor.

CAPÍTULO II

2.1. Marco contextual

En Ballenita, una comunidad costera en expansión con un turismo en auge, la gestión de residuos se ha convertido en un desafío significativo. Actualmente, el sistema de recolección de residuos presenta problemas de ineficiencia en las rutas de recolección. Según la información disponible en la página web EMASAEP de Municipio de Santa Elena, los horarios oficiales de recolección en el centro y malecón de Ballenita son de 8:00 a 9:00 y de 16:30 a 17:30. Sin embargo, estos horarios pueden ocasionar demoras en la llegada de los vehículos de recolección debido a la necesidad de cubrir varias zonas del sector de Ballenita.

Los tiempos reales de recolección a menudo difieren de lo programado, lo que resulta en una acumulación considerable de residuos. Con base en un análisis de la cantidad de contenedores y los desafíos actuales, esta propuesta de implementación busca mejorar este aspecto crítico de la gestión de residuos[12].

2.2. Marco Teórico

2.2.1. Residuos sólidos urbanos (RSU)

Los residuos sólidos urbanos son los residuos que proviene de hogares, negocios, oficinas y servicio. Incluyen todos los residuos que no son peligroso y pueden ser gestionado como residuos comunes.

Los RSU son todos aquellos objetos y materiales que se desechan porque han cumplido su propósito y carecen de valor. La mayor parte de residuos se producen principalmente en las zonas urbanas y sus alrededores. Los residuos sólidos se pueden clasificar y separar en centros especiales para obtener materiales que puedan transformarse en nuevos productos, lo cual beneficia a la economía. Los desechos que no se reintegran a la

cadena de valor después de una caracterización y valoración se denominan desechos y son enviados a vertederos, rellenos sanitarios o incineradores para su eliminación[13].

2.2.1.1. Caracterización de los residuos sólidos urbanos

La caracterización de los desechos sólidos urbanos (RSU) es el proceso mediante el cual se analizan y clasifican los diversos tipos de desechos producidos en zonas urbanas. La caracterización de residuos permite identificar principalmente dos tipos de residuos: reciclables y no reciclables. El objetivo de la caracterización es comprender la composición y las características de los desechos para optimizar su gestión, fomentar el reciclaje y la reutilización y reducir el impacto ambiental mediante una disposición final adecuada[14].

Los RSU no reciclables serán enviados para disposición final a vertederos o rellenos sanitarios. Su composición clasifica los RSU reciclables en una de las siguientes categorías:

- Vidrio
- Papel o cartón
- Plástico
- Materiales orgánica
- Metal
- Electrónicos
- Llantas usadas

2.2.1.2. Generación y recolección de residuos sólidos

La generación de residuos puede provenir a partir de actividades domésticas, comerciales, agrícolas e industriales. Para establecer indicadores de generación, es necesario determinar la producción por habitante al día en una zona determinada. La recolección de residuos debe organizarse para ofrecer un servicio eficiente y equitativo, evitando la

generación de malos olores, polvos, desorden, contaminación ambiental y alteración del paisaje de la ciudad. Para implementar un sistema eficiente, es necesario realizar un diagnóstico que revise los horarios de recolección, el volumen de residuos generados, la distancia del punto de recolección hasta el centro de disposición y los tiempos requeridos. Además, se debe considerar cuando el camión se encuentre completamente lleno, evaluando la cantidad de residuos generados para obtener una regularidad horaria.

Esto se logra trazando hojas de ruta. En cada zona, en función de la densidad poblacional y el tamaño físico, se determina el número de viajes necesarios y el tiempo por cada ruta, estableciendo así los horarios y frecuencias de recolección[13].

2.2.2. Internet de las Cosas (IoT)

El Internet de las cosas (IoT) son la agrupación e interconexión de dispositivo y objetos electrónicos a través de una red privada o Internet. En cuanto a los dispositivos electrónicos esto pueden variar desde los sensores o procesadores hasta dispositivo mecánicos, todos pueden conectarse a Internet para permitir la interacción de máquina sin interacción humana. Los avances científicos han contribuido al desarrollo y uso generalizado de Internet de las cosas, se puede utilizar en diversos campos y posibilidades. El Internet de las cosas utiliza tecnología relacionada para recopilar datos mediante sensores eléctricos y enviarlos a la red a través de Internet para su análisis de compartimiento[13].

2.2.3. Aplicaciones del IoT en la Gestión de Residuos

El Internet de las cosas (IoT) está transformando diversas industrias, y la gestión de residuos definitivamente está sintiendo el impacto. Gracias al IoT, la gestión de residuos está adquiriendo un aspecto completamente nuevo. Las ciudades y las empresas están utilizando


tecnología sofisticada para hacer que la recolección, el reciclaje y la eliminación de residuos sean más fáciles y eficientes. Estas aplicaciones tienen como objetivo hacer que todo funcione mejor, ahorra dinero y ayuda al medio ambiente.




Mediante sensores inteligente, conectividad y análisis de datos en tiempo real, los sistemas de gestión de residuos de IoT le ayudan a monitorizar los contenedores de basura, realizar un seguimiento de flota de recogida y analizar cuantos residuos se generan. Estos contenedores inteligentes tienen sensores que pueden saber cuándo están llenos, por lo que los camiones de basura pueden planificar mejor sus rutas y ahorrar combustible, lo que es bueno para el medio ambiente. Al analizar los datos, podemos detectar patrones y tomar decisiones más inteligentes para reducir los residuos y reciclar de manera más eficaz[15].

2.2.4. Sensores IoT en la Recolección de Residuos

La incorporación de sensores IoT en recolección de residuos es importante e innovador para el manejo de desechos urbanos, facilitando una optimización eficaz y sustentable del proceso. Se muestran en la Tabla 1 los distintos tipos de sensores empleados en el manejo de desechos, abarcando desde sensores para medir niveles de temperatura, humedad y gas. Estos dispositivos son imprescindibles para controlar las condiciones de los contenedores y maximizar la eficiencia en las rutas de recolección.

Tabla 1 Tipos de sensores IoT

Tipos de sensores IoT			
Tipo de sensor	Especificaciones	Función	
Sensor ultrasónico	Medición de distancia	Monitorea el nivel de llenado y optimiza las rutas	

Sensor infrarrojo	Detección de objetos	Sistema para detectar presencia y nivel de residuos en contenedores	
Sensor de temperatura	Mide y registra la temperatura	Detecta cambio de temperatura y gestiona residuos específicos	
Sensor de humedad	Sensores capacitivos	Mide humedad para evitar acumulación y mejorar clasificación de residuos	
Sensor de gas	Sensores MQ (MQ-2, MQ-7, etc.)	Detecta gases tóxicos o inflamable	
Sensor de peso	Carga de celdas	Mide el peso del contenedor para calcular el volumen de residuos recolectados	
Sensor de movimiento	Acelerómetros o giroscopios	Detector de movimiento para seguridad de contenedores.	

2.2.5. Arquitecturas para Sistemas IoT en la Gestión de Residuos

Internet de las cosas (IoT) está compuesta por diversos componentes, estructuras de red y tecnología en la nube que funciona de acuerdo con protocolos y estándares de seguridad establecidos. Los dispositivos y los sensores están separados de la capa de la arquitectura IoT, que utilizan puertas de enlace y protocolos para monitorear la consistencia de un sistema.

Estas arquitecturas se diseñan para mejorar la eficiencia y sostenibilidad en la

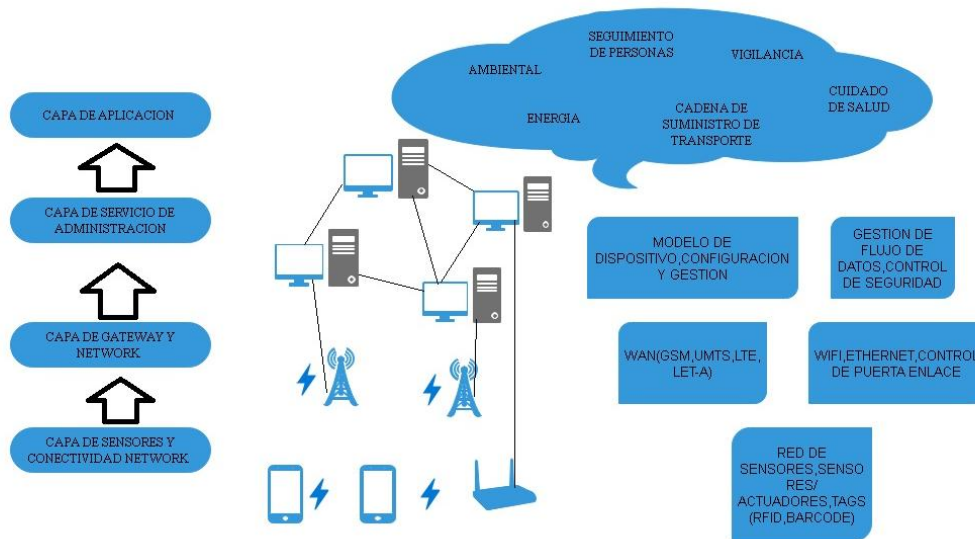


Figura 1 Capas de arquitectura

recolección y gestión de desechos mediante la integración de tecnologías como sensores de Internet de las cosas, aprendizaje automático, inteligencia artificial y visión por computadora para optimizar procesos y reducir costos en el ámbito de la gestión de desechos[16].

La Figura 1 muestra la estructura del IoT para la recolección de residuos, dónde la parte enfocada en sensores y conectividad permite la transmisión de información desde los sensores y actuadores hasta la infraestructura de procesamiento, que comprende puertas de enlace y plataforma a la nube.

La sección de Gateway y red asegura una comunicación segura y eficaz, facilitando el enlace entre la información recopilada y el sistema de almacenamiento. La sección de

administración de servicios tiene la responsabilidad de supervisar y vigilar los dispositivos y la estabilidad de la red, además de ofrecer una base de datos y un servidor web que proporciona una API REST para administrar el acceso y utilización de los datos. Finalmente, la sección de aplicaciones posibilita la visualización, administración y monitoreo de los dispositivos IoT, incluyendo desde paneles de control sencillos hasta sistemas sofisticados capaces de examinar datos de manera automatizada.

2.2.6. Protocolo de la capa de aplicación

2.2.6.1. Protocolo MQTT

MQTT es un protocolo de mensajería basado en estándares. Los sensores inteligentes, los dispositivos portátiles y otros dispositivos IoT, suelen usar redes con recursos y anchos de banda limitados para enviar y recibir datos. Dado que es fácil de implementar y comunicarse con datos IoT de manera eficiente, estos dispositivos utilizan MQTT para la transmisión de datos, que permiten la mensajería entre dispositivos en la nube[17].

Importancia del protocolo MQTT

Ligero y eficiente: MQTT es ideal para dispositivos IoT, porque requiere pocos recursos y optimiza el ancho de banda, serán aptos para microcontroladores.

Escalable: MQTT es eficiente en términos de código y energía, permitiendo la conexión con millones de dispositivos IoT.

Fiable: Ayuda a dispositivos IoT a reconectarse más rápido en redes celulares poco confiables, que tienen baja velocidad de datos y tiempos de respuesta lentos.

Seguro: ayuda a los desarrolladores a asegurar los mensajes y autenticar dispositivos y usuarios usando métodos modernos como OAuth, TLS 1.3[17].

Funcionamiento de MQTT

MQTT es un protocolo eficiente para dispositivos IoT, donde los clientes pueden enviar y recibir mensajes, al conectarse con un agente MQTT. Facilita una comunicación efectiva en redes con recursos limitados, permitiendo a los dispositivos intercambiar datos de manera optimizada[17].

Seguridad de MQTT

La comunicación MQTT utiliza el protocolo SSL para proteger los datos confidenciales transmitidos por dispositivos IoT, permitiendo autenticación, autorización e identificación mediante certificados SSL y contraseñas[17].

2.2.6.2. Protocolo HTTP/HTTPS

Un protocolo esencial para la comunicación entre clientes y servidores en la web es HTTP, que permite que los navegadores envíen solicitudes al servidor y reciban respuestas en forma de texto sin formato. Es una versión segura de HTTP que utiliza cifrado para proteger la privacidad y la integridad de los datos que se transmiten entre el navegador y el servidor del dispositivo[18].

Funcionamiento de HTTP y HTTPS

Protocolo	•Protocolo de Transferencia de Hipertexto
Seguridad	•No ofrece cifrado ni seguridad adicional
Puerto predeterminado	•80
Conexion	•No segura, datos transmitidos en texto plano
Integridad y privacidad	•Los datos pueden ser interceptados y leídos
Certificado SSL/TSL	•No requiere certificado
Utilizacion	•Adecuado para sitios web que no manejan datos sensibles

Grafica 1 Funcionamiento de HTTP

Protocolo	• Versión segura del Protocolo de Transferencia de Hipertexto
Seguridad	• Utiliza cifrado SSL/TLS para seguridad
Puerto predeterminado	• 433
Conexion	• Segura, datos transmitido de manera cifrada
Integridad y privacidad	• Los datos están cifrados, protege la privacidad
Certificado SSL/TSL	• Requiere un certificado SSL/TLS valido
Utilizacion	• Esencial para sitios web que manejan datos sensibles o transacciones financieras

Grafica 2 Funcionamiento de HTTPS

2.2.7. Definición de LPWAN

La tecnología LPWAN juega un papel importante en la construcción del Internet de las cosas. Estas plataformas de red permiten el uso de muchos sensores para recopilar y transmitir datos a largas distancia y proporcionar optimización energética a costo menor que otras opciones de conectividad.

Varias tecnologías compiten entre sí en el espacio de LPWAN, como: NB-LoT, Sigfox, Lora, etc. Tanto Bluetooth como Wifi tienen coberturas claramente cortas en comparación con las comunicaciones celulares y LPWAN, pero la tecnología 3G/4G utiliza mucha más potencia de transmisión que LORA. El alcance entre el emisor y el receptor depende del entorno en el que operan. Un rango máximo se puede lograr siempre que no haya obstáculos en el campo de visión de Fresnel. La cobertura interior depende en gran medida de tipo de material de construcción utilizado[19].

Características de LPWAN

- ✓ **Bajo consumo de energía:**

Una de la característica destacada de LPWAN es su eficiente consumo de energía. Esta tecnología está diseñada para dispositivos que funcionan con baterías y que requieren una larga vida útil, muchos protocolos LPWAN incluye modos de suspensión profunda para reducir el consumo de energía[20].

✓ **Larga distancia de transmisión:**

Se caracteriza por una amplia zona de cobertura, que puede alcanzar varios kilómetros en zonas rurales y ciudades densamente pobladas. Con esta tecnología se puede establecer comunicaciones incluso con señales débiles donde proporciona una transmisión eficiente y confiable en todo tipo de entornos[20].

✓ **Tasa de Transferencia de Datos Reducida:**

Esto es adecuado para aplicaciones que necesitan enviar pequeñas cantidades de datos. Esta función esta optimizada específicamente para Internet de las cosas y también es ideal para sensores y dispositivos que envíen datos de forma intermitente[21].

✓ **Dispositivo de bajo costo e implementación:**

En términos de costo, los módulos de comunicación LPWAN son más económicos que otras tecnologías inalámbricas en cuanto los costó de implementación de redes como Sigfox y Lora son relativamente bajos, ya que requieren menos infraestructura que a las redes generalizadas[20].

2.2.8. Tecnologías LPWAN

No todas las LPWAN son iguales, hay varias tecnologías disponibles que trabajan tanto en frecuencia con y sin licencia. Con la tecnología adecuada, las redes LPWAN podrán establecer conexiones de forma fácil y eficiente.

De acuerdo con una investigación llevado a cabo por ALFA IoT[22], se identificaron cuatro tecnologías más comunes utilizadas en el campo de IoT y es necesario describir cada

uno de ellos en la sección siguiente.

2.2.8.1. LoRa

La tecnología lora (Long Range) fue desarrollada específicamente para aplicaciones de Internet de las cosas (IoT) y permite comunicaciones inalámbricas de largo alcance. Al igual que Sigfox, Lora se centra en dispositivos que necesitan transmitir pequeñas cantidades de datos a larga distancia con bajo consumo de energía. Las redes con estas características se denominan LPWAN (Low Power Wide Area Network). LoRa es la tecnología más adecuada para proyecto IoT por su eficiencia, costo reducido y gran alcance.

Dentro del amplio rango de tecnología inalámbricas, tales como Wifi, Bluetooth, LTE, Sigfox y Zigbee, LoRa se destaca por emplear una forma específica de modulación de radiofrecuencia conocida como Chirp Spread Spectrum (CSS). Semtech ha patentado esta tecnología, gracias a su resistencia a interferencia y capacidad para mantener comunicaciones a largas distancia, es ampliamente utilizada en aplicaciones militares y espaciales.

La tecnología LoRa fue desarrollada inicialmente por Semtech, sin embargo, en la actualidad es gestionada y promovida por LoRa Alliance que se dedica al desarrollo y promoción a esta tecnología, facilitando así la creación de redes IoT eficiente y escalables[22].

2.2.8.2. Sigfox

Sigfox es una tecnología LPWAN basada en modulación UNB desarrollada por la empresa francesa Sigfox para proporcionar conectividad de Internet de las cosas (IoT) utilizando espectro sin licencia. La red actúa como una red LPWAN, que permite transmitir mensajes a baja velocidad y bajo costo, reduciendo el consumo de energía de los dispositivos desconectados. Sigfox trabaja para complementar los sistemas conectados para garantizar el

servicio y la confiabilidad y para automatizar procesos a través de relaciones con los desarrollos de IoT. Es particularmente eficiente para comunicaciones de enlace ascendente desde el dispositivo terminal a la estación base, pero menos eficiente para comunicaciones de enlace descendente. Las redes Sigfox están actualmente disponibles en más de 25 países y pueden transmitir mensajes a distancias de 30 a 50 kilómetros en áreas rurales y de 3 a 10 kilómetros en áreas urbanas utilizando mensajes de enlace ascendente livianos (12 bytes sin incluir el encabezado de carga útil) en el rango de 2 a 50 kilómetros[23].

2.2.8.3. NB-IoT

NB-IoT, estandarizado por el 3GPP (3rd Generation Partnership Project), es una tecnología de redes de área amplia de baja potencia (LPWAN) diseñada para optimizar el consumo energético de los dispositivos conectados. Esta tecnología se distingue por su capacidad de penetración, siendo especialmente atractiva en áreas donde las tecnologías celulares convencionales enfrentan dificultades de cobertura.

NB-IoT puede operar junto con GSM (The Global System for Mobile Communications) y LTE (Long Term Evolution) en bandas de frecuencia con licencia, como 700, 800 y 900 MHz, utilizando un ancho de banda de frecuencia de 200 KHz. Las velocidades de datos están limitadas a 20-250 Kbps para el enlace ascendente y hasta 200 Kbps para el enlace descendente. Cada mensaje puede llevar una carga útil máxima de 1600 bytes, mientras que la eficiencia energética permite que la batería de un dispositivo tenga una vida útil de hasta 10 años transmitiendo en promedio 200 bytes diarios.

La tecnología NB-IoT ha sido probada en una amplia variedad de aplicaciones prácticas como medición, estacionamiento y agricultura inteligente, implementadas en regiones de Asia, América y Europa. Según la GSMA (Asociación GSM), NB-IoT cuenta

con 92 redes desplegadas globalmente. Un aspecto distintivo de NB-IoT es su capacidad para operar en redes celulares públicas utilizando espectro de radio con licencia, que facilita su despliegue y explotación comercial al aprovechar la infraestructura existente de los operadores móviles[24].

2.2.8.4. LTE-M

El LTE-M, que se deriva del estándar 4G LTE del proyecto de asociación de tercera generación (3GPP), es una máquina diseñada para la evolución a largo plazo. Aprovecha todas las ventajas que ofrece LTE, como una arquitectura de protocolo de Internet plano eficiente, movilidad mejorada, mecanismos prioritarios de manejo y seguridad robusta. Además, cuenta con una base globalmente establecida. Cuenta con sólidas conexiones a las redes 2G y 3G heredadas de la 3GPP, una transición sin problemas hacia la poderosa solución del 5G. LTE-M es una opción eficaz para abarcar diversas aplicaciones que requieren diferentes anchos de banda y latencias. Aunque comparte similitudes importantes con el Internet de las cosas de banda estrecha, posee beneficios adicionales en cuanto a su rango de ancho de banda, latencia y capacidad para movilizarse. Algunas desventajas son que los dispositivos finales pueden ser complicados y costosos de implementar. LTE-M es la tecnología de red de área amplia (LPWAN) de baja potencia más adecuada para las aplicaciones de IoT que necesitan velocidades de datos más altas, latencias bajas, movilidad total y capacidad para transmitir voz en situaciones típicas donde hay buena cobertura[24].

La Tabla 2 se expone una comparación de las tecnologías LPWAN, donde son las tecnologías más representativas las que se destacan. A partir del análisis realizado en la tabla, se puede apreciar las principales características técnicas de cada una de las tecnologías más representativas, información muy útil a considerar en el momento de plantear o diseñar un

plan o proyecto relacionado con el Internet de las Cosas (IoT).

Tabla 2 Característica de la tecnología LPWAN

Aspecto técnico	LORA	SIGFOX	NB-IoT	LTE-M
Modulación	CSS	BPSK	QPSK	QPSK
Cobertura y alcance	1-5 km(urbano) 10-40km(rural)	3-10km (urbano) 10-50 km (rural)	1-10 km (dependiendo del entorno)	1-10 km (dependiendo del entorno)
Banda de frecuencia	Banda ISM (sin licencia) en Europa 868MHz	Banda ISM (sin licencia) en Europa 868MHz	LTE con licencia	LTE con licencia y 2G
Ancho de banda	125kHz, 250kHz 500kHz	192 kHz	1.4 MHz	200 kHz
Velocidad de datos	50 kbps	100 bps	250 kbps	1 Mbps
Máximo de mensaje/día	Ilimitado	140(UL) y 4 (DL)	Ilimitado	Ilimitado
Máxima de carga útil y longitud	243 bytes	12 bytes 8 bytes	1600 bytes	27.2 bytes 62.2 bytes
Bidireccional	Half-duplex	Half-duplex	Half-duplex	Full-duplex
Latencia	Media y alta	Alta	Baja y media	Baja
Seguridad	Cifrado de extremo a extremo	Cifrado de extremo a extremo	Cifrado y autenticación 3GPP	Cifrado y autenticación 3GPP
Duración de batería	15 años	20 años	10 años	10 años
Costo	Muy bajo	Muy bajo	Moderado	Moderado
Inmunidad e	Muy alto	Muy alto	Bajo	Medio

interferencia				
Handover	Los terminales no están conectados a una única estación base			
Consumo de batería	Muy bajo	Muy bajo	Bajo	Bajo a moderado
Típicas aplicaciones	Medidores inteligentes, Monitoreo ambiental, Cuidades inteligente	Cuidades inteligente, Rastreo de activo, sensores	Medidores inteligentes, Cuidades inteligente, Monitoreo de infraestructura	Vehículos conectados, Monitoreo de salud, logística

2.2.9. Servidores LoRaWAN

El servidor maneja la capa de aplicación LoRaWAN, entre estas, su propia función la descifra y descodifica de datos de uplinks, cola de downlinks y cifrado y codificación de downlinks. En otras palabras, es el componente responsable de que se maneje toda la información de los paquetes hacer válida de datos y hacerla plenamente accesible a las final aplicaciones. Este suele actuar como servidor con protocolos habituales del ámbito del Internet de las Cosas. como MQTT y HTTP.

LoRaWAN proporciona múltiples soluciones para desplegar redes IoT de amplia cobertura y bajo consumo, abarcando diferentes usos del caso y redes. Entre las más destacadas tenemos ChirpStack, una opción open-source y altamente configurable y modular; The Things Network (TTN) simplifica las redes comunitarias con una interfaz de aplicación muy simple y fácil; Loriot asegura disponibilidad y servicio empresarial a las empresas para las aplicaciones comerciales, industriales más dependeran; Actility ThingPark se enfatiza a su gran escalabilidad en alta implementación de despliegue en despliegue a gran

escala[25].

The Things Network (TTN)

El servidor TTN nos permite conectar directamente y enviar información hacia plataformas externas, esta funcionalidad es muy útil para crear visualizaciones y registro de los datos sentidos por los dispositivos finales conectados a este servidor. Además, TTN también ha creado el servidor The Things Stack (TTS), que es un servidor robusto y flexible de red LoRaWAN que atiende a los despliegues exigentes de LoRaWAN[26].

ChirpStack

ChirpStack es una solución en código abierto, modular y altamente configurable que permite flexibilidad en la implementación de redes LoRaWAN. Contiene tres componentes fundamentales, el Network Server, el Application Server y Join Server,mando de la concentración total de red desde conectar los dispositivos hasta procesar datos. Es ideal para quienes están en busca de una plataforma personalizable y flexible para cumplir con variadas necesidades[27].

Loriot

Loriot es una plataforma robusta de gestión de redes LoRaWAN, así como para todas aquellas aplicaciones comerciales e industriales que asegura una gran disponibilidad, escalabilidad y seguridadSu arquitectura flexible admite despliegues en la nube o locales con soporte para una amplia gama de dispositivos y Gateway. Loriot se distingue por sus tools de monitoreo, administración de dispositivos y analítica de datos, proporcionando una óptima y confiable operación[28].

Actility ThingPark

Actility ThingPark es una plataforma comercial específica para gestionar de forma eficiente redes LoRaWAN, unge soluciones desarrolladas para masivo despliegues y casos

de uso avanzados en Internet of the Things (IoT). Su arquitectura modular incluye módulos de Network Server, Application Server y un conjunto de herramientas de monitoreo avanzado para la administración a nivel superior de las redes híbridas, públicas o privadas con alta escalabilidad y no disponible. Además, su compatibilidad con estándares internacionales y su capacidad de la interoperabilidad con a un amplio rango de equipos y servicios de la nube simplifica el tiempo de implementación de las soluciones IoT complejas, maximizando el desempeño, y asegurando la seguridad de la comunicación[29].

2.2.10. LoRa en la gestión de residuos

Es un módulo de tecnología de conectividad inalámbrica que se utiliza en sistema IoT. El módulo LoRa esta patentado y fabricado por semtech corporation. las ventajas de la tecnología LoRa son una amplia cobertura de área con bajo consumo de energía, miles de nodos se pueden conectar en una red, lo que hace adecuada para el Internet de las cosas (IoT)[30].

Estas comunicaciones estas destinada principalmente a dispositivos IoT que necesitan transmitir pequeños datos. La cantidad de información es muy pequeña, lo que hace que no requieran altas velocidades de transmisión, y la prioridad es reducir el consumo energético para alargar la vida útil de la batería de estos dispositivos.

La implementación de redes de nodos utilizando tecnología LoRa es viable debido a su bajo costo, rápido despliegue, amplia cobertura y eficiente consumo de energía. Estos nodos serán los encargados de recopilar y enviar datos relacionados con diversas variables como el nivel de llenado, humedad, temperatura y gas etc.

La característica única de LoRa incluye el consumo de energía ultra bajo, gran alcance de cobertura y penetración en interiores, uso de espectro sin licencia, geolocalización esto

permite a las redes localizar sus dispositivos, despliegue de redes pública y privadas utilizando el mismo hardware y software que integra seguridad total en el protocolo LoRaWAN y actualizaciones de firmware en línea.

Entre los casos de uso que identificamos: servicio público inteligente, como el monitoreo de nivel del agua, supervisión y gestión de desechos en contenedores, monitoreo de los niveles de radiactividad, seguimiento y gestión de flotas de camiones y en la agrícola es el monitoreo de crecimiento de las plantas[31].

2.2.10.1. Principales Características de LoRa

Conectividad de largo alcance: Cobertura de 15 kilómetros en zonas rurales y 5 kilómetros en ciudades, y tiene buena penetración en edificios.

Bajo consumo de energía: La duración de la batería es de 15 años y reduce los costos de mantenimiento

Transmisión segura: Cifrado AES de 128 bits para proteger los datos transmitidos.

Geolocalización precisa: Puede localizar señales con una precisión de decenas de metros sin utilizar GPS, lo que resulta útil para el seguimiento de activos.

Móvil: Permite la comunicación constante con dispositivos en movimiento sin perjudicar el consumo de energía.

Alta capacidad: LoRa es ideal para los operadores de redes públicas en mercados de gran escala porque admite millones de mensajes por estación base.

2.2.10.2. Arquitectura de la red LoRa

Dispositivo o nodos finales: se trata de componentes que permiten que los dispositivos (sensores o actuadores) se conecten a la red LoRa y recopilen información del entorno después se envíen a la puerta de enlace[32].

Puerta de enlace (Gateways): conocida como estación base en una red LoRa que recopilan información enviada por los nodos finales y la envía a un servidor de red[32].

Servidor de red: gestiona el flujo de datos entre la puerta de enlace y aplicaciones[32].

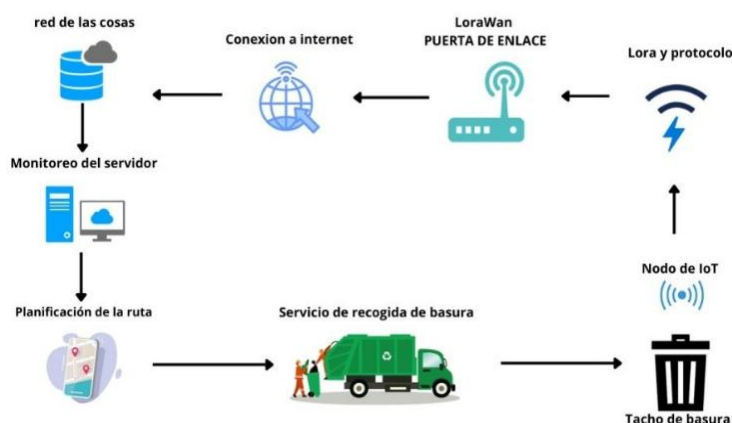


Figura 2 Arquitectura de la red LoRa

2.2.11. Tecnología Sigfox

Sigfox es una red LPWAN dirigida al IoT que transmite datos de tamaño limitado con un alcance de larga distancia y bajo consumo, implementando la tecnología UNB (Ultra Narrow Band) en las bandas ISM, utilizando 192 kHz de las bandas disponibles para el envío de mensajes. Los dispositivos finales se conectan a las estaciones base mediante modulación DBPSK (Differential Binary Phase-Shift Keying) en una UNB con un ancho de canal de 100 Hz en la banda de frecuencia ISM. Al utilizar UNB, se emplea un ancho de banda eficiente, que resulta en un nivel bajo de ruido. Los sistemas de comunicación UNB establecen comunicaciones inalámbricas de baja potencia ocupando un rango estrecho de frecuencias, con alta sensibilidad de recepción, baja potencia de consumo, un diseño de antena económico y un rendimiento máximo de 100-600 bps. Los datos que se pueden enviar tienen un tamaño de 12 bytes y se pueden enviar hasta 140 mensajes por día en el enlace ascendente y 4

mensajes por día en el enlace descendente de 8 bytes[33].

2.2.11.1. Características principales de la tecnología Sigfox

- Alcance de larga distancia: coberturas de 3-8 kilómetros en áreas urbanas y rurales 30-50 kilómetros.
- Bajo consumo de energía: diseñada para dispositivos IoT con una larga duración de batería.
- Frecuencia ISM: Opera bandas ISM sin necesidad de licencia específica.
- Tecnología UNB: Tiene alta eficiencia espectral y nivel bajo de ruido.
- Capacidad de mensaje limitada: Enlace ascendente 140 mensaje/día y descendente 4 mensajes/día.
- Modulación DBPSK y GFSK: Para enlace ascendente (DBPSK) y descendente (GFSK), optimización robusta y eficiente.
- Seguridad: Incluye cifrado de datos y autenticación para garantizar comunicaciones seguras.
- Red bidireccional: Soporta comunicación bidireccional con capacidad limitada en el enlace descendente[34].

2.2.11.2. Arquitectura de sistema Sigfox

La arquitectura de red IoT se define como un modelo que incluye diferentes tecnologías como redes de sensores inalámbricos, redes móviles y redes de actuadores, donde los objetos inteligentes se ubican y conectan a una única dirección. Las redes Sigfox utilizan una infraestructura de estaciones base, antenas y equipos propietarios dispuestos en una topología en estrella con un funcionamiento detallado y simplificado: muchas cosas se

comunican y envían datos a una estación base que escucha constantemente el espectro, recibe información y envía datos.

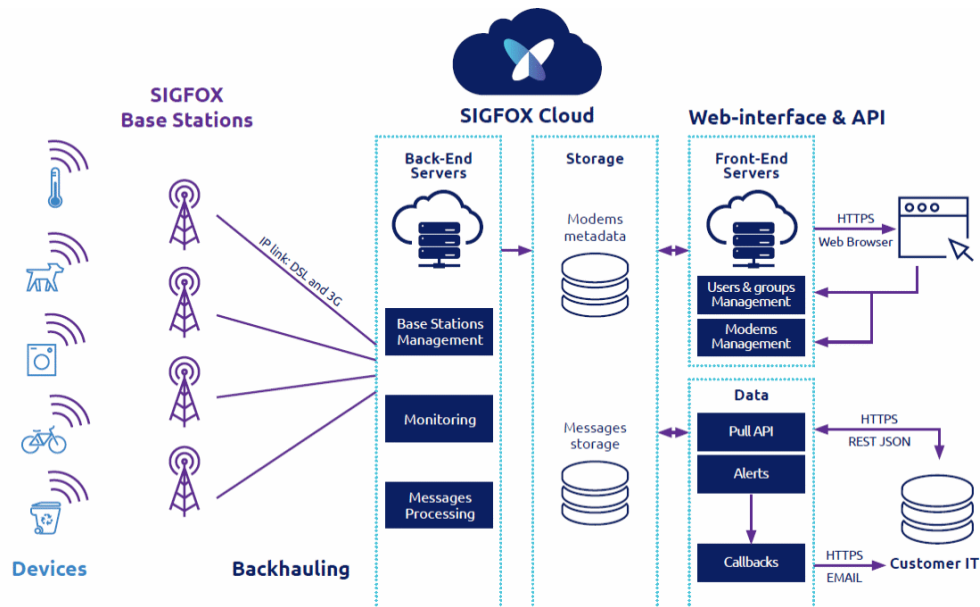


Figura 3 Arquitectura del SigFox

Nota. En esta imagen describe que la redes IoT como SIGFOX facilitan la comunicación de dispositivos y gestionan datos. Incluyen estaciones base, servidores de back-end e interfaces para interactuar con los datos del sistema. Tomado de block-administration[35].

La tecnología UNB permite a Sigfox reducir las colisiones potenciales que causan interferencias en la transmisión, ofrece una conexión más fuerte y resistente para mantener altos niveles de QoS y permitir una gran escalabilidad con múltiples objetos interconectados.

La conexión entre la estación base y la nube de Sigfox se establece parcialmente mediante tecnologías como Ethernet, radioenlaces por satélite o redes móviles 3G/4G/5G. Sigfox Cloud se gestiona como un proveedor de servidor para el uso y almacenamiento de los datos de los clientes, proporcionando acceso directo a sus datos, funciones y

mensajes[36].

2.2.12. LoRaWAN

En un protocolo de comunicación con una amplia área de cobertura (LPWAN) y bajo potencia que permite que los dispositivos con baterías se conecten a Internet. Su enfoque es atender las necesidades fundamentales del Internet de las cosas (IoT), como la comunicación bidireccional, la seguridad integral, la capacidad de manejar la movilidad de dispositivo y a capacidad de localización, son su principal objetivo[31].

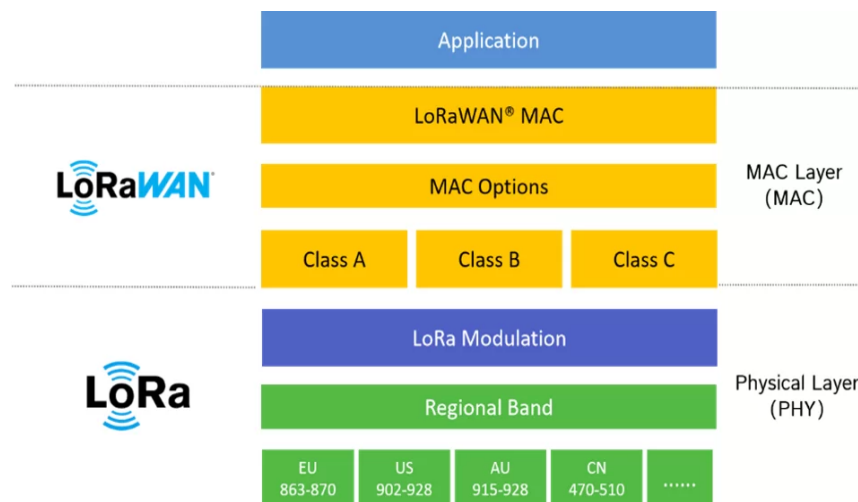


Figura 4 Protocolo LoRa

Nota. es una representación de la arquitectura del protocolo LoRaWAN que tiene tres capas que son: Aplicación, MAC y Física, optimizando para las redes de baja potencia con funciones específicas para la comunicación eficiente. Tomado de vencoel[37].

2.2.12.1. Clases de nodos

LoRaWAN tiene tres tipos de dispositivos: clase A, B y C. Todos los dispositivos deben tener clase A y los otros dos son como versiones especiales. Todos los dispositivos pueden comunicarse entre sí en ambas direcciones, lo que significa que pueden enviar y recibir mensaje.

Cuando se trata de actualizar el firmware inalámbricas, el dispositivo debe volver a una clase inferior, como la clase B o clase C.

Cuando se trata de enviar mensaje de enlace ascendente, los dispositivos finales no pueden enviar mensaje de enlace descendente de manera simultánea[31].

Clase A: En una red LoRaWAN, todos los dispositivos deben ser compatibles con la tecnología clase A. Este dispositivo puede enviar mensaje cuando quiera, abrir dos ventanas de recepción y recibir mensaje de otro dispositivo en la red. Esta clase es super eficiente en lo que respecta al uso de energía y es perfecta para situaciones en las que solo necesitan enviar datos. El tiempo de la transmisión se basa en el dispositivo final y la recepción puede recibir datos después de una transmisión exitosa[38].

Clase B: Los dispositivos de esta clase extienden las capacidades de Clase A ,al abrir periódicamente ventanas de recepción conocidas como "ranuras de ping", destinadas a recibir mensajes de enlace descendente. Estas ventanas de recepción se crean sin necesidad de una transmisión previa, lo que mejora la capacidad de recibir datos al mantener sincronizado el dispositivo final con el Gateway. Esta sincronización adicional también resulta en un aumento del consumo eléctrico, el dispositivo debe mantener una mayor actividad para gestionar las ventanas de recepción[38].

Clase C: Esta clase de dispositivos finales permanece en modo de recepción y sólo se interrumpe cuando se ejecuta una transmisión, que resulta en un consumo mayor debido a la mejor latencia entre el dispositivo final y el Gateway. Los dispositivos de esta clase extienden las capacidades de Clase A, al mantener abiertas las ventanas de recepción a menos que estén transmitiendo un enlace ascendente. Esto implica que los dispositivos de Clase C pueden recibir mensajes de enlace descendente en casi cualquier momento, que resulta en una latencia muy baja para los enlaces descendentes[38].

2.2.12.2. Módulo de transmisión LoRaWAN

En sistemas de comunicación, los módulos de transmisión son indispensables para permitir la transferencia de datos entre dispositivos a través de tecnologías como Bluetooth, Wi-Fi o GSM. Su uso es común en aplicaciones de telecomunicaciones, IoT y control remoto para garantizar una conexión eficiente y la transmisión de información instantánea. En la Tabla 3 se muestran los distintos tipos de modulo LoRa empleados en el manejo de desechos.

Tabla 3 Modulo LoRaWAN

Característica	RAK811 Lora	CubeCell HTCC-AB02	TTGO LoRa32 ESP32
Protocolo	LoRaWAN	LoRaWAN	LoRa/LoRaWAN
Frecuencia	433MHz, 868MHz, 915MHz	433MHz, 868MHz, 915MHz	433MHz, 868MHz, 915MHz
Microcontroladores integrados	No	STM32L0 ,SX1262 LoRa chip	ESP32 (dual-core Xtensa LX6
Memoria flash	Ninguna	128KB Flash	4MB Flash
Memoria Ram	Ninguna	20KB SRAM	520KB SRAM
Alimentación	3.3V	3.3V	3.3V
Consumo de energía	Bajo de energía	Ultra bajo consumo (modo deep sleep)	Variable bajo en modo de bajo consumo
Capacidades adicionales a	GPS	ADC, RTC, batería recargable integrada	Wi-Fi, Bluetooth, pantalla OLED
Interfaz	UART, SPI	ADC, RTC, batería recargable integrada	UART, SPI, I2C

2.2.13. Algoritmos de rutas

2.2.13.1. Algoritmo de Dijkstra

El algoritmo de Dijkstra es un modelo de búsqueda que identifica la ruta más eficiente entre un punto de partida y cualquier otro punto dentro de una red. El algoritmo opera mediante iteraciones y emplea etiquetas temporales y permanente para identificar la ruta más eficiente[39].

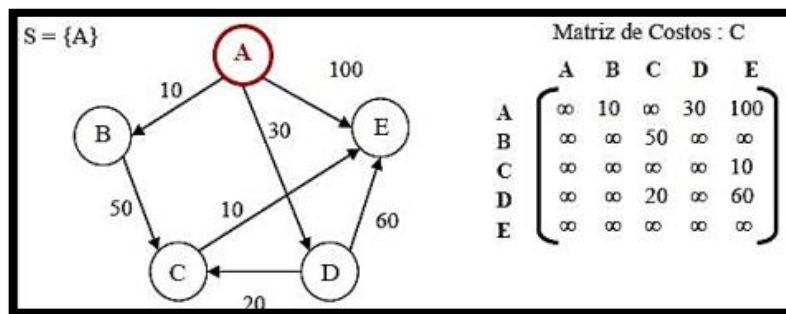


Figura 5 Algoritmo de Dijkstra

2.2.14. Plataforma de mapeo de ruta

2.2.14.1. Algoritmo de Bellman-Ford

El algoritmo calcula las rutas más cortas de forma ascendente, al igual que otros problemas de programación dinámica. Primero, calcula las distancias más cortas que tienen un borde máximo en la ruta. Luego calcula los caminos más cortos con dos aristas como máximo, y así sucesivamente[40].

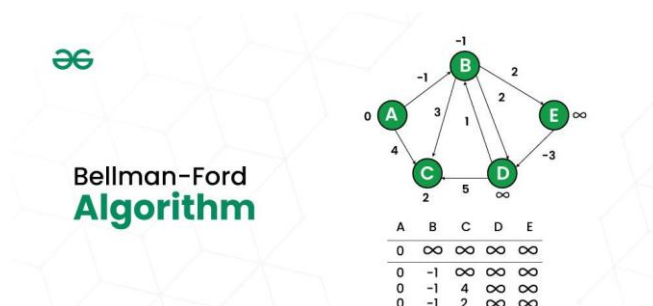


Figura 6 Algoritmo de Bellman-Ford

Diferencias entre Bellman Ford y el de Dijkstra:

Ambos calculan la distancia más corta de cada vértice de un gráfico desde un vértice de fuente única. Sin embargo, en la Tabla 4 podemos observar las diferencias importantes entre ellos.

Tabla 4 Diferencias entre el Dijkstra y Bellman Ford

Algoritmo de Dijkstra	Algoritmo de Bellman Ford
El algoritmo de Dijkstra puede fallar con bordes de peso negativo y no funcionará con ciclos de peso negativo.	El algoritmo Bellman-Ford maneja bordes negativos y detecta ciclos negativos.
El resultado incluye vértices con información completa de la red.	El resultado incluye vértices con información sobre sus conexiones a otros vértices.
El algoritmo de Bellman Ford tiene más gastos generales que el algoritmo de Dijkstra.	El algoritmo de Dijkstra tiene menos gastos generales que el algoritmo de Bellman Ford.
El algoritmo de Dijkstra tiene más escalabilidad	El algoritmo de Bellman Ford tiene menos escalabilidad

2.2.14.2. API de Roads

Roads API es un servicio que acepta solicitudes HTTPS con coordenadas de latitud y longitud para uno o más puntos en un mapa. Usa estos puntos para encontrar tramos de ruta cercanos y muestra un ID de lugar junto con metadatos, como el tramo de ruta más cercano y los límites de velocidad. Con la API de Roads, puedes obtener límites de velocidad, horas de llegada estimadas y otros metadatos de las rutas, mejorando tus apps de conducción con

funciones como conectar coordenadas de latitud y longitud con rutas, encontrar la ruta más cercana para un conjunto de coordenadas y obtener información sobre los límites de velocidad para los tramos de ruta[41].

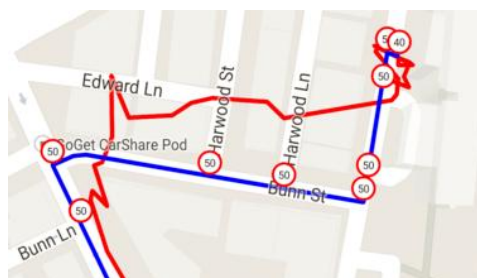


Figura 7 API de Roads

Funcionalidades de la API de Roads

Ajuste de carreteras: Este servicio utiliza hasta 100 coordenadas de GPS proporcionadas para mostrar las rutas más probables y puede agregar puntos adicionales para crear una ruta más fluida.

Rutas cercanas: El servicio proporciona segmentos de carretera para coordenadas GPS individuales, procesando hasta 100 puntos y devolviendo el segmento más cercano para cada uno, sin requerir que los puntos formen una ruta continua[41].

2.2.14.3. Mapquest

Proporciona indicaciones de conducción muy detalladas a cualquier lugar basadas en el mapa y la información de las ubicaciones establecidas en ese mapa. Ayuda a orientarse rápidamente y muestra las condiciones del tráfico en la ruta y el tiempo estimado para llegar al lugar que se busca, permitiendo encontrar las rutas más convenientes[42].

Ventajas de mapquest:

- Toma la ruta o la distancia más corta.
- Informa a los usuarios sobre las leyes de tránsito y los límites de velocidad en muchos lugares.

CAPITULO III

3.1. Componentes de la propuesta

3.1.1. Componentes físicos

Para el desarrollo y puesta en marcha del sistema de recolección de residuos basado en LPWAN e IoT implica el uso de varios componentes físico que se necesitan para operar de manera efectiva. Estos incluyen sensor de nivel de llenado, dispositivos de comunicación LPWAN utilizados para enviar datos de esta forma los equipos utilizados involucran unidades de procesamiento de información, así como las fuentes de alimentación necesarias para garantizar la autonomía del sistema, los cuales trabajan de manera coordinada para llevar a cabo la vigilancia y control del estado de los contenedores de residuos de forma sincrónica en tiempo real, mejorando las rutas de recolección y operatividad del servicio de limpieza urbana. Dentro de lo mencionado se hizo uso del siguiente dispositivo:

✓ Gateway SenseCAP M2

En la Figura 8 se observa el Gateway SenseCAP M2 es un dispositivo de puerta de enlace LoRaWAN® multiplataforma. Fue producido para habilitar la comunicación de bajo consumo y de larga distancia en una red de Internet de las Cosas. La puerta de enlace está permitida por los juegos globales de franjas de LoRaWAN® 865MHz-923MHz, lo que la hace apropiada para una variedad de aplicaciones dentro de la tecnología de edificios inteligentes, monitoreo ambiental y agricultura digital.

El Gateway es clave en la configuración de un sistema de recolección de residuos IoT ya que ofrece la conectividad y transmisión de datos a larga distancia de los sensores ubicados a los contenedores a las plataformas de monitoreo. Mediante tecnologías LPWAN, asegura una comunicación sostenible, de bajo consumo energético y vasta cobertura. En la Tabla 5

se encuentran las características principales del Gateway seleccionad a para la propuesta, proporcionando una visión general de sus funcionalidades y beneficios[43].

Tabla 5 Características principales del Gateway SenseCAP M2

Característica Técnica	
Modelo	Gateway SenseCAP M2
Chip de puerta de enlace LoRa	SX1302 / SX1303
Antena	LoRa: Antena externa de 3dBi
Wifi	Soporta 2.4 GHz 802.11 b/g/n
Ethernet	RJ45 (10M / 100 M)
Protocolo de comunicación	LoRaWAN (Clases A y C)
Voltaje de entrada	12V DC – 2A
Servidores de red compatible	The Things Network (TTN), ChirpStack, Helium, AWS IoT Core, Actility, Loriot



Figura 8 Gateway SenseCAP M2[43]

✓ **Modulo Heltec CubeCell AB-02**

En la Figura 9 Se observa la placa CubeCell HTCC-AB02, este módulo es desarrollado por Heltec, específicamente para las aplicaciones LoRa/LoRaWAN que necesitan de comunicación inalámbrica a largo alcance y de bajo consumo de energía. Cuenta con la base de los modelos ASR605x, que combinan el microcontrolador (MCU) y el TX/RX

chip SX1262, lo que garantiza una transmisión eficiente de datos. Además, su diseño está proyectado para que los programadores puedan integrarlo fácilmente con baterías o paneles solares de manera externa, lo cual lo vuelve una opción ideal para proyectos que requieren un diseño compacto debido a su tamaño reducido y su alta funcionalidad.

El módulo Heltec CubeCell AB-02 es fundamental en el sistema de recogida de residuos mediante LPWAN e IoT, ya que registra y procesa los datos de los sensores de nivel de contenedores, transmitiendo estos datos a largas distancias a través de LoRaWAN. Su bajo consumo de energía garantiza un seguimiento autónomo prolongado y su capacidad de procesamiento en el dispositivo, mejora la red eficientemente reduciendo la cantidad de datos que se envían, así la gestión remota y la toma de decisiones en plataformas como The Things Network (TTN). En la Tabla 6 se presentan las características principales del módulo CubeCell HTCC-AB02[44].

Tabla 6 Características principales del Cubecell HTCC-AB02

Característica Técnicas	
Modelo	Cubecell HTCC-AB02
Puerta de enlace	Compatible con SenseCAP M2 y otros Gateway LoRaWAN.
Transceptor LoRa	SX1262
Frecuencia	470 MHz / 868 MHz / 915 MHz (según la región)
Temperatura de operación	-20 a 80 °C
Batería	3.7 v de litio
Interfaz de programación	USB-C y UART
Energía solar	5.5-7 v (panel solar)



Figura 9 CubeCell HTCC AB-02[44]

✓ **Sensor Ultrasónico JSN-SR04T**

Es un sensor ultrasónico que es una versión mejorada del HC-SR04 y presenta la calidad de impermeabilidad de hasta un rango de 500 cm de medición lo que hace que este sensor pueda ser utilizado en agua sin ningún tipo de miedo, siempre y cuando la profundidad no sea muy alta. Este sensor incluye un cable conectado al módulo con una longitud de 2,5 m y un rango de tensión de 3-5 voltios para el procesamiento de la señal. La forma de trabajo de este sensor se fundamenta en la ley de la reflexión, que utiliza las ondas sonoras emitidas y dependiendo de que los objetos devuelvan la señal que el sensor, reciba de vuelta.

En la Figura 10 se muestra el sensor JSN-SR04T que es utilizado para medir el nivel de los contenedores de residuos. Este sensor funciona mediante la emisión de ondas ultrasónicas que, al reflejarse en la superficie de los desechos, permiten calcular la distancia entre el sensor y el nivel del material acumulado. Estos datos luego son enviados a dispositivo IoT, como el módulo Heltec CubeCell AB-02, que transmite la información a través de LoRaWAN hacia una plataforma centralizada para el monitoreo, lo que permite la supervisión y optimización de las rutas de recogidas de manera remota. En la Tabla 7 se presenta las características principales del sensor[45].

Tabla 7 Características principales del Sensor JSN-SR04T

Característica Técnicas	
Modelo	JSN-SR04T
Voltaje de operación	5v
Rango de detección	25 a 450 cm
Precisión	3mm
Tiempo de respuesta	< 100ms
Frecuencia de operación	40kHz
Angulo de detección	< 50°
Tipo de sensor	Ultrasónico, resistente al agua
Temperatura de trabajo	-10°C hasta 70°C



Figura 10 Sensor JSN-SR04T[45]

✓ **Batería de litio 3.7v 1500mAh**

La batería de polímero de litio de 3,7 V y 500 mAh es una fuente de alimentación recargable que se utiliza a menudo en el diseño de dispositivos electrónicos compactos. Se utiliza especialmente en dispositivos como auriculares inalámbricos, donde se necesita una solución energética ligera, compacta y eficiente. En el desarrollo de sistemas electrónicos para el monitoreo constante, es fundamental contar con fuentes de energía que sean tanto confiables como eficientes. Esto asegura que los dispositivos puedan operar de manera continua durante largos períodos sin interrupciones. Por lo tanto, se optó por utilizar esta batería como la principal fuente de energía para el dispositivo que se encarga de monitorear el nivel de llenado de los contenedores. En la Figura 11 se muestra el aspecto físico de la

batería utilizada[46].



Figura 11 Bateria de litio 3.7v 1500mAh[46]

3.1.2. Componentes lógicos

➤ Arduino IDE

El Arduino IDE es una herramienta clave en el diseño y la programación de sistemas embebidos, permitiéndonos programar microcontroladores de una forma sencilla y eficiente. Esta interfaz de desarrollo integrada, escrita en lenguaje similar a C/C++, permite al usuario la codificación, compilación y carga de código a placas Arduino en un entorno fácil de usar. Dada su versatilidad, el Arduino IDE facilita la conexión a sensores, actuadores y módulos de comunicación para la creación de sistemas automatizados de diversas complejidades, facilitando la implementación de soluciones orientados a la robótica, domótica, monitorización ambiental, agropecuaria, entre otro sector. A su vez, la posibilidad de compilar y depurar el código en tiempo real aumenta eficiencia del sistema y reduce la posibilidad de fallos, muy importante en tareas altamente tecnológicas.

➤ Node-red

Node-RED es una herramienta de programación para enlazar dispositivos de hardware, servicios web y API. Node-RED es de código abierto y fue desarrollado por IBM

Emerging Technology. Básicamente es una herramienta de programación visual destinada para Internet de las Cosas, pero puede ser usada en otras aplicaciones para construir de manera rápida flujos de distintos servicios. Está soportado por Node.js (una plataforma de JavaScript del lado del servidor). Node-RED permite a los usuarios integrar servicios web y gadgets reemplazando tareas de codificación comunes, lo que debería ser posible con una interfaz visual de arrastrar y soltar.

Muchos componentes de Node-RED se conectan entre sí para formar un flujo en el propio editor de Node-RED. Los flujos de Node-RED están almacenados JSON, se puede exportar e importar fácilmente para compartir con otros. Node-RED puede ejecutarse en un navegador local en <http://localhost:1880>, también proporciona un módulo de interfaz de usuario que proporciona un conjunto de nodos en Node-RED para crear rápidamente un panel de datos en tiempo real. La interfaz de usuario está disponible en el navegador local en <http://localhost:1880/ui>.

3.2. Diseño de la propuesta

3.2.1. Diseño y conexión de los componentes del sistema de monitoreo

En la Figura 12 se muestra el diseño del sistema, se integra el sensor especializado que permite la medición de parámetros fundamentales, como el nivel de llenado de los contenedores. Los datos se captan y procesan mediante un microcontrolador central. La arquitectura del sistema está estructurada sobre un modelo de varias capas, en las cuales cada una tiene un propósito particular y diseñada para aprovechar al máximo las capacidades del Internet de las Cosas (IoT). El modelo se conforma de cuatro capas: capa de percepción, capa

de red, capa de middleware y capa de aplicación.

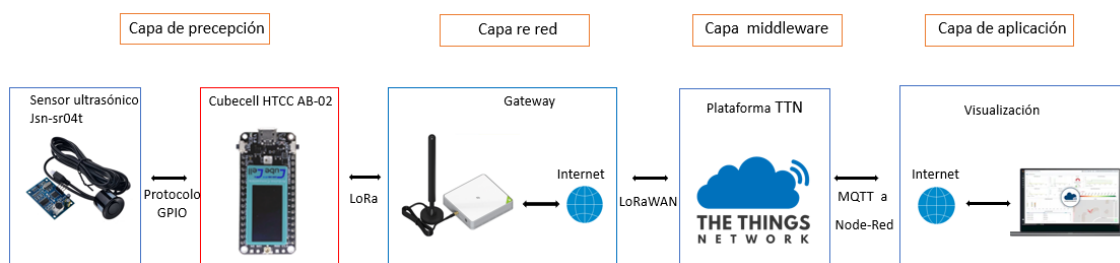


Figura 12 Diseño y conexión de los componentes del sistema de monitoreo

En la capa de percepción aparece el primer nivel de la arquitectura del sistema, responsable de la recolección de datos. En esta capa, el sensor ultrasónico JSN-SR04T, conocido por su función para medir distancias mediante ondas ultrasónicas donde se encarga de capturar información del entorno, como la distancia a un objeto o el nivel de un líquido en un contenedor. Este sensor está conectado a Cubecell HTCC AB-02, un microcontrolador compacto especializado en IoT, no sólo lee el sensor, sino que también procesa los datos que recibe. El HTCC AB-02, con bajo consumo energético y capacidades de conectividad inalámbrica, transforma los datos en un formato adecuado para su transmisión, preparando que la información pueda propagarse de manera eficiente a la siguiente etapa del sistema.

En la capa de red, la comunicación entre el dispositivo físico y la nube se logra mediante la comunicación entre el Cubecell y un Gateway. El Cubecell cuenta con una conexión inalámbrica, probablemente basado en el protocolo LoRa, perfecto para transmisiones de largas distancias y con bajo consumo energético, características que el Cubecell AB-02 soporta por naturaleza. El Gateway un dispositivo que actúa como intermediario, recibe los datos enviados por Cubecell y los remite a través de Internet a una plataforma en la nube. Este proceso asegura que los datos capturados en el entorno físico puedan llegar a servidores remotos, superando las limitaciones de distancia y conectividad.

Por lo tanto, la capa de red juega un papel fundamental al asegurar una comunicación eficaz y fiable entre los dispositivos de IoT y las infraestructuras digitales, facilitando que los datos estén accesibles para su posterior procesamiento.

Las capas middleware y de aplicación completan el ciclo de esta arquitectura. En la capa middleware, la plataforma The Things Network (TTN) actúa como un sistema central que administra la información obtenida a través del Gateway. TTN, es muy utilizado en proyectos de IoT, se ocupa de guardar, ordenar y manejar la información, garantizando que esté preparada para ser utilizada en aplicaciones prácticas. Mediante el uso del protocolo MQTT, un estándar de comunicación ligero y eficaz para IoT, TTN transmite los datos procesados a la capa de aplicación. En esta fase final, la información se muestra al usuario final a través de una interfaz de visualización, en la cual es una aplicación en web.

3.2.2. Diseño eléctrico

Se llevó a cabo el diseño electrónico del prototipo utilizando el programa Fritzing, herramienta que ofrece funcionalidades para ilustrar gráficamente conexiones y circuitos electrónicos, como se aprecia en la Figura 13. Debido a su utilidad para distribuir con claridad los distintos elementos del sistema, este software es una opción común en el desarrollo de proyectos electrónicos. Este software facilita el proceso de ensamblaje físico del circuito, al permitir una planificación ordenada que minimiza errores y optimiza el tiempo de construcción. A continuación, se describe el procedimiento realizado para la elaboración y distribución de los distintos componentes del prototipo.

Selección de los componentes:

Para la implementación se eligieron componentes que facilitaran una operación eficiente y bajo consumo de energía. Como unidad principal de control se utilizó el módulo

CubeCell HTCC-AB02, debido a su integración con tecnología LoRa y su compatibilidad con varios sensores. Se optó por el sensor ultrasónico JSN-SR04T como dispositivo de medición, es adecuado para entornos húmedos debido a su diseño impermeable. Se utilizó una batería de polímero de litio como fuente de energía, con capacidad para mantener el circuito operativo durante un tiempo razonable sin conexión continua a una fuente externa.

Integración de la fuente de alimentación:

La alimentación del sistema se hizo mediante una batería de 3.7V conectada directamente al módulo CubeCell. Este dispositivo cuenta con sistema interno de gestión de carga que supervisa el proceso de alimentación de forma segura y eficiente, protegiendo la batería de sobrecargas y garantizando su larga duración.

El módulo CubeCell HTCC-AB02 incluye un conector USB de tipo C, que tiene diversas funciones, entre las que destacan:

Recarga de batería: Permite conectarse a una fuente de alimentación externa para cargar la batería de forma directa. Este proceso se controla automáticamente mediante un circuito integrado de control de carga, lo que evita riesgos relacionados con la sobrecarga.

Interfaz de programación y configuración: Ofrece una conexión directa desde un ordenador, donde se puede cargar firmware, editar el código del programa y ajustar las configuraciones del sistema sin tener que utilizar equipos de desarrollo adicional.

Conexión del sensor JSN-SR04T:

En cuanto el sensor ultrasónico JSN-SR04T se conecta al módulo CubeCell por cuatro líneas principales. La línea de alimentación VCC del sensor, está conectada al pin de salida de voltaje del CubeCell, puede ser de 3.3v o 5v dependiendo de la configuración y la línea GND se conecta al pin de tierra común del módulo.

En cuanto su función, el sensor tiene dos pines principales que permiten medir la distancia a un objeto mediante ondas ultrasónicas que son:

Trigger: Es el pin que se encarga de activar la medición. El sensor recibe un pulso del microcontrolador, emite una onda ultrasónica hacia su entorno.

Echo: Es el pin que envía una señal de salida al microcontrolador. esta señal es el tiempo que le toma a la onda ultrasónica alcanzar un objeto y regresar, lo que permite medir la distancia de forma precisa.

Estas conexiones permiten una comunicación eficiente entre el sensor y el microcontrolador, garantizando una lectura precisa y confiable de los datos.

Validación del diseño:

Una vez finalizado el diseño esquemático, se procedió al proceso de verificación con el entorno gráfico del software utilizado para comprobar la lógica de las conexiones y verificar la ausencia de errores eléctricos. Finalmente, con base en el diseño se continuó con el ensamblaje físico del prototipo, siguiendo estrictamente el plano diseñado. Concluida su construcción se realizaron diversas pruebas funcionales que permitieron la valoración del comportamiento del sistema. Los resultados obtenidos mostraron un correcto funcionamiento tanto del suministro de energía al microcontrolador como en la comunicación con el sensor, lo que se logró verificar la funcionalidad y confiabilidad del diseño electrónico diseñado.

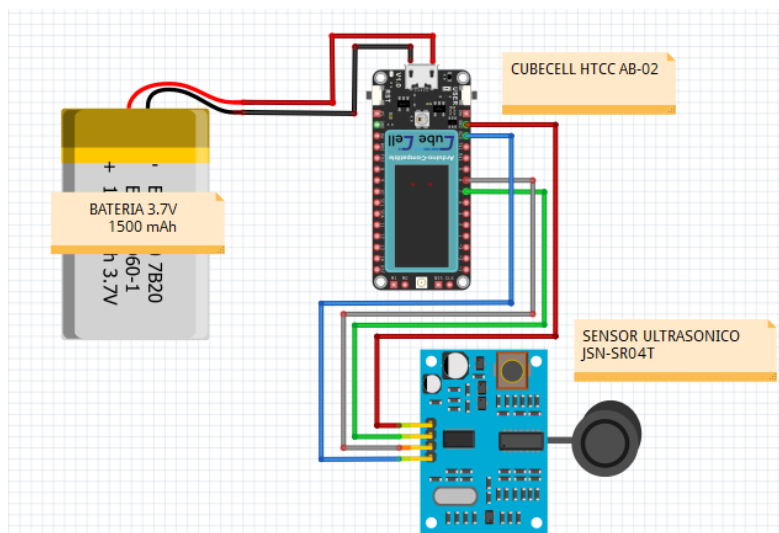


Figura 13 Diagrama eléctrico

La Tabla 8 presenta la asignación de pines correspondiente del sensor JSN-SR04T. Estos datos son fundamentales para garantizar una interpretación adecuada y precisa de las conexiones necesarias, lo que a su vez asegura el rendimiento óptimo del dispositivo.

Tabla 8 Conexiones del sensor JSN-SR04T

Sensor JSN-SR04T		Placa CubeCell HTCC-AB02
VCC		3.3V
GND		GND
TRIG		7(GPIO7)
ECHO		6(GPIO6)

3.3. Configuración de los dispositivos

3.3.1. Configuración del Gateway SenseCAP m2

Configuración de la red de puerta de enlace

Se conectó la antena LoRa al puerto correspondiente del Gateway, asegurando de que quedara correctamente ajustada. Luego se conectó el cable de corriente al puerto USB-C del

dispositivo para encenderlo. Una vez cargado el dispositivo, el indicador de encendido led se pone en color rojo, tal como se puede observar en la Figura 14.

Pasado unos 15 segundos, el led superior empezó a parpadear en color verde, lo que significa que la puerta de enlace está en fase de arranque. Existe dos maneras de conectarse a Internet.

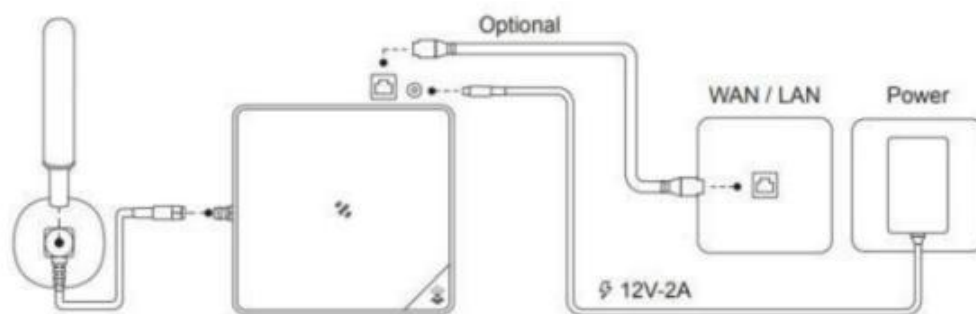


Figura 14 Esquema de conexión del Gateway SenseCAP[47]

➤ **Conexión por cable Ethernet**

Para configurar la conexión del Gateway a Internet sobre una red cableada, se conectó un cable Ethernet en el puerto correspondiente del dispositivo. Una vez conectada esta opción se enciende el LED situado en la parte superior del Gateway en color verde fijo, indicando que el puente de enlace se encontraba correctamente configurado en la red. Es importantísimo, que tanto el Gateway como el pc donde estará configurado, estén conectados al mismo enrutador o a la misma red local, ya que, al estar conectados, así como habrá comunicación directa entre los dos equipos. Luego, se accedió a la página de configuración del enrutador para verificar la dirección IP asignada al dispositivo. Esta dirección IP está

requerida para acceder a la interfaz de configuración de Gateway mediante un navegador Web. Finalmente se obtuvieron las credenciales de acceso nombre usuario y contraseña las cuales están impresas en el rótulo pegado al cuerpo del dispositivo, como se muestra en la Figura15.



Figura 15 Detalle de la etiqueta del SenseCAP M2

Una vez obtenida la IP del equipo, ingresamos la dirección 192.168.168.1 en la barra de navegación del navegador web permitiéndonos acceder a la consola local del Gateway. En esta interfaz, se pedían las claves del acceso por lo tanto se ingresó el username y password al dispositivo. Finalmente se dio en el botón de inicio de sesión para acceder a la configuración del sistema, como se muestra en la Figura 16

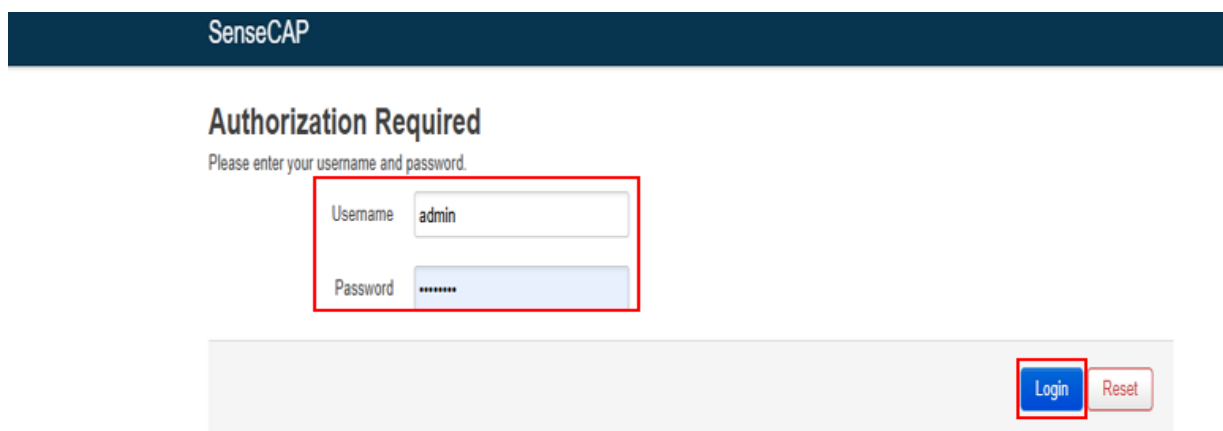


Figura 16 Inicio de sesión en la interfaz de configuración del SenseCAP

➤ **Conexión por red WIFI**

Una vez iniciado sesión nos dirigimos a Network y damos click a Wireless, tal como se visualiza en la Figura 17.

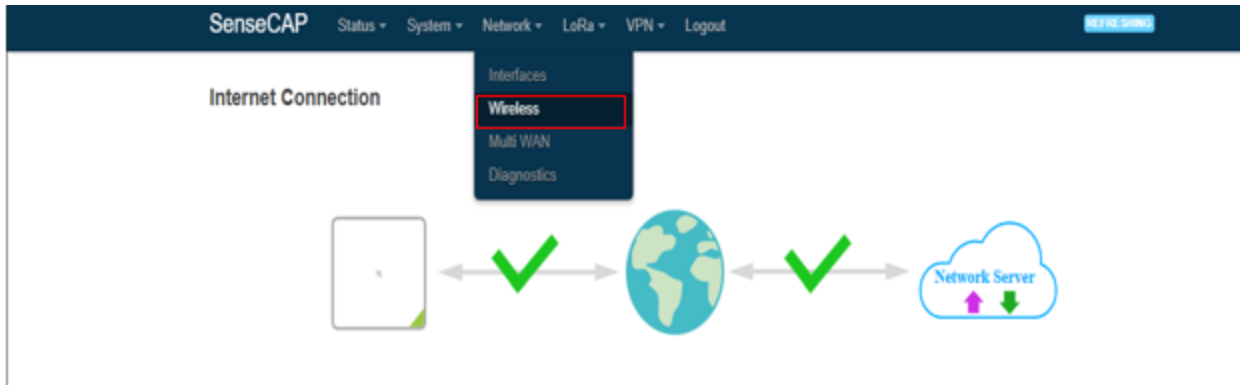


Figura 17 Conexión del Gateway SenseCAP por red wifi

De acuerdo con la Figura 18, una vez dado click en Wireless damos en Scan para escanear el wifi

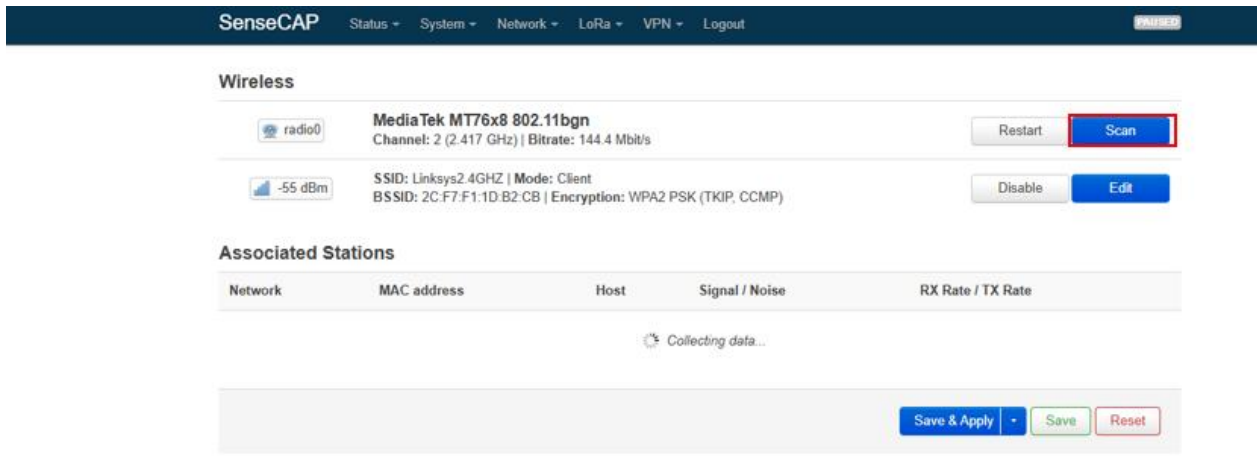


Figura 18 Parámetros de conexión inalámbrica en el SenseCAP

Selecciona tu wifi para unirte a la red ,como se indica en la Figura 19.

Join Network: Wireless Scan

Signal	SSID	Channel	Mode	BSSID	Encryption	
-55 dBm	Linksys2.4GHZ	2	802.11b	44:9B:C1:F1:61:AC	WPA2-PSK [AES]	Join Network
-78 dBm	Rodríguez yiga5 2.4	4	802.11b	44:9B:C1:F1:61:AC	WPA2-PSK [AES]	Join Network
-81 dBm	CMCC_share	10	802.11b	44:9B:C1:F1:61:AC	WPA2-PSK [AES]	Join Network
-82 dBm	JUANKA2020	10	802.11b	44:9B:C1:F1:61:AC	WPA2-PSK [AES]	Join Network
-83 dBm	DESCONOCIDO	8	802.11b	44:9B:C1:F1:61:AC	WPA2-PSK [AES]	Join Network
-86 dBm	JAZMIN2G	4	802.11b	44:9B:C1:F1:61:AC	WPA2-PSK [AES]	Join Network
-91 dBm	AGROCALIDAD24	11	802.11b	44:9B:C1:F1:61:AC	WPA2-PSK [AES]	Join Network
-92 dBm	NETLIFE-stlmzamoram1	2	802.11b	44:9B:C1:F1:61:AC	WPA2-PSK [AES]	Join Network

Figura 19 Interfaz de escaneo de redes Wi-Fi en el SenseCAP

Después de introducir en la consola de configuración los datos de la red Wifi, se optó por la opción guardar y aplicar con el fin de confirmar los cambios realizados. Como se observa en la Figura 20, una vez que se completó este proceso, el indicador LED superior del Gateway parpadeó y se quedó iluminado a verde fijo, indicando que el Gateway se había conectado correctamente a la red inalámbrica, tal como se demuestra en la Figura 21.

Joining Network: "Linksys2.4GHZ"

Replace wireless configuration

Name of the new network

WPA passphrase
Specify the secret encryption key here.

Lock to BSSID
Instead of joining any network with a matching SSID, only connect to the BSSID `44:9B:C1:F1:61:AC`.

Figura 20 Configuración de datos de la red Wifi



Figura 21 Vinculación del Gateway a una red Wi-Fi

3.3.2. Configuración de dispositivos LoRaWAN para la transmisión de datos Registro en The Things Network (TTN)

Para ingresar a The Things Network se procede abrir un navegador e ingresar a la página oficial <https://www.thethingsnetwork.org/>. Una vez teniendo la página cargada damos click en iniciar sesión y crear cuenta, Al ingresar se accede al modo consola. En la Figura 22 se muestra el proceso de registro en TTN.



Figura 22 Registro en The Things Network

3.3.3. Configuración del dispositivo Gateway en The Things Network (TTN)

Conforme a las indicaciones del proveedor del Gateway, el primer paso que se deberá realizar es conectar el equipo al Internet. Para ello se estableció los parámetros de red requeridos para su conexión inalámbrica a una red privada mediante Wifi. La verificación del acceso exitoso se puede apreciar como se muestra en la Figura 23.



Figura 23 Configuración del dispositivo Gateway en The Things Network

➤ Configuración de la puerta de enlace

Para configurar la comunicación dentro de la red LoRa hay que configurar una serie de parámetros fundamentales, como se ve en la Figura 24.

Paso 1: Configuración de Red LoRa

Paso 2: Establecer el modo en reenvío de paquetes

Paso 3: Durante la configuración de la red el sistema encontrará automáticamente el EUI del Gateway conectado. También se define el protocolo de reenvío de paquetes mediante el puerto 1700, protocolo estándar para las redes LoRaWAN de comunicación bidireccional entre Gateway y servidores. Si bien, este protocolo, establecido a través de UDP y creado por Semtech, ha sido implantado por la LoRa Alliance por su sencillez y compatibilidad para poder incorporar y dirigir dispositivos en redes IoT.

Paso 4: Configuración de plan de canales

Paso 5: La región US902-928 es una banda de frecuencias de espectro ISM (Industrial, Scientific, and Medical), utilizada en los Estados Unidos entre dispositivos inalámbricos sin licencia. Este rango que va entre 902 y 928 MHz es excelente para usar en aplicaciones IoT, ya que muchos módulos de hardware están diseñados para trabajar en estos espectros, mientras que el plan de frecuencia FSB1 (channel 0 ~ channel 7) ofrece un uso eficiente del espectro dividiéndolo en subbandas específicas, reduciendo colisiones e interferencias entre dispositivos, garantizando compatibilidad, flexibilidad y conformidad a la norma. Tal como se refleja en la Figura 25.



Figura 24 Configuración de la puerta de enlace



Figura 25 Configuración de la puerta de enlace

Una vez configurado el Gateway para que tenga acceso a Internet y los parámetros para conectividad con LoRa, procedemos a registrar el Gateway en TTN, como se muestra en la Figura 26. Para eso se ingresa a la consola de TTN. Luego ir al apartado **Gateway** y se

seleccionó la opción **Register Gateway**. Se define el **EUI** de la puerta de enlace que está en la etiqueta del dispositivo. Posteriormente, se diseñó un **ID** específico compuesto por letras minúsculas, números y guiones. También se le dio nombre para su identificación Finalmente seleccionar el **plan de frecuencia** apropiado, según la versión del dispositivo, para asegurar así su compatibilidad con el entorno de operación, tal como se muestra en la Figura 27. Una vez realizado este procedimiento se pudo visualizar en tiempo real el estado operativo del Gateway a través de la consola de TTN, permitiendo verificar su estado de conexión y recibimiento de paquetes de datos, de acuerdo con lo ilustrado en la Figura 28.

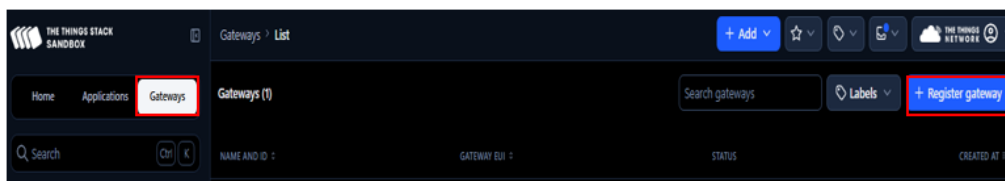


Figura 26 Registro el Gateway en TTN

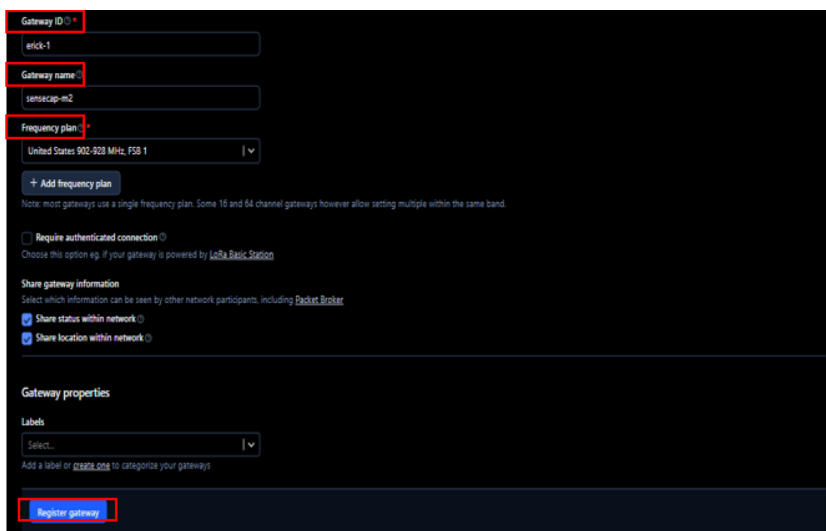


Figura 27 Configuración avanzada del Gateway: Etapa de registro y parámetros específicos

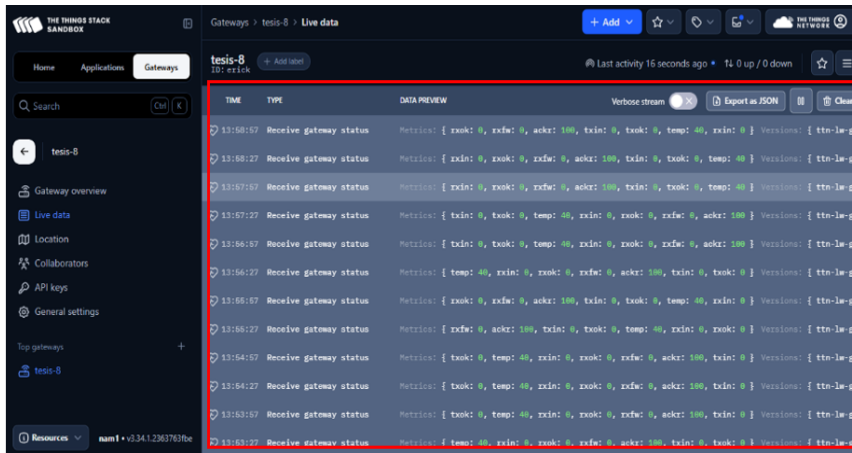


Figura 28 visualización de datos en la plataforma TTN

3.3.4. Registro del Heltec AB-02 con TTN

Para registrar el módulo **HTCC AB-02** es necesario entrar a la consola de administración. Luego, debemos ingresar al apartado de **Aplicación** y seleccionar la opción **Añadir aplicación**. Al hacerlo se abrirá una ventana nueva en la que se deben rellenar los campos correspondientes: **Application ID**, **Application Name** y una pequeña **Descripción**. Por último, se elige la opción "**Create Application**" para cerrar el proceso, tal como se muestra en la Figura 29.



Figura 29 Registro del módulo CubecCell HTCC AB-02

Una vez creada la aplicación, se procede a registrar el dispositivo para el cual se quiere utilizar. Se debe acceder al apartado **End Devices** (Dispositivos Finales) en la consola

de administración. En esta sección se elige la opción **Register End Device** (Registrar Dispositivo), como se muestra en Figura 30.

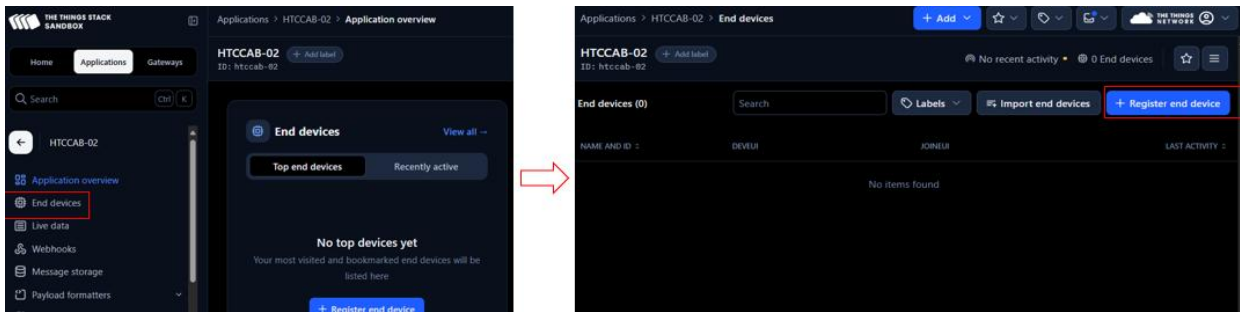


Figura 30 Registro de dispositivos finales en The Things Network

Al seleccionar la opción nuevo dispositivo, se abre un formulario donde se deben ingresar datos como el tipo del hardware (Brand ID y Model ID), la región (Profile) y los identificadores únicos (DevEUI, AppSKey, NwkSKey, etc.). Esta información es crucial para asegurar la compatibilidad del dispositivo con la red LoRaWAN, para facilitar una autenticación segura y garantizar su funcionamiento adecuado, como se visualiza en la Figura 31.

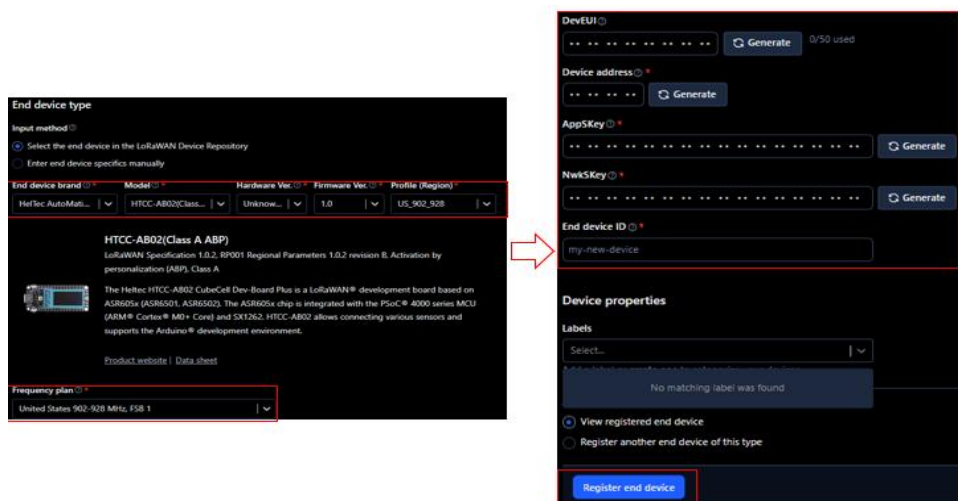


Figura 31 Configuración de dispositivos finales en The Things Network: Asignación de identificadores y claves

3.3.5. Configuración y transmisión del Heltec AB-02 con TTN

En primer lugar, se debe configurar la herramienta de desarrollo con la plataforma Arduino IDE. Para ello es necesario insertar la placa CubeCell Board a través de gestor de tarjetas. Antes se necesita agregar la URL del repositorio oficial Heltec al apartado de preferencias dentro del IDE, lo cual permite la descarga e instalación de las herramientas propias para la programación del módulo CubeCell HTCC-AB02, como se observa en la Figura 32.

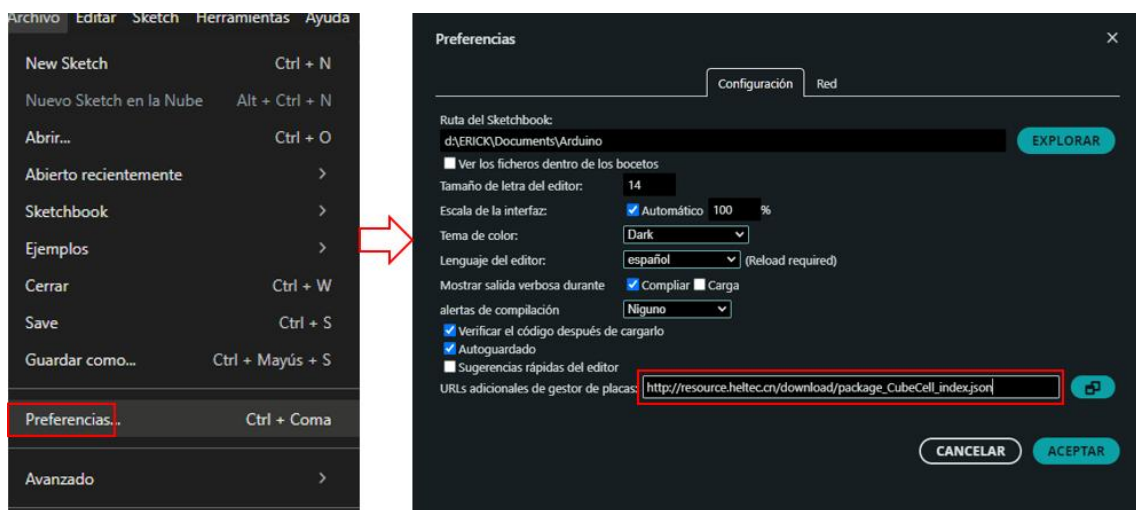


Figura 32 Interfaz de configuración de preferencias en Arduino IDE

A continuación, se instalan las librerías necesarias a través del Gestor de Librerías del Arduino IDE. Estas ofrecen funciones importantes, como la comunicación LoRa y la lectura de sensores. Las librerías necesarias se muestran en la Tabla 9 a continuación.

Tabla 9 Librerías empleadas en dispositivos CubeCell con LoRaWAN

Librería	Descripción	Función principal
CubeCell Development Framework	Herramientas diseñadas especialmente para dispositivos CubeCell.	Proporciona soporte completo para el hardware y funciones básicas.

LualtekCubeCell	Librería adaptada para sensores y módulos CubeCell en aplicaciones IoT.	Lectura de sensores y envío de datos por LoRa.
MCCI Arduino LoRaWAN Library	Implementación confiable del protocolo LoRaWAN compatible con TTN.	Permite la conexión con redes LoRaWAN como The Things Network (TTN).
IBM LMIC Framework	protocolo LoRaWAN adaptada para Arduino.	Maneja la comunicación LoRa a bajo nivel.

➤ **Transmisión de datos con el módulo CubeCell HTCC-AB02 en The Thing Network.**

Para enviar los datos del monitor serial del Arduino IDE a The Things Network (TTN), el módulo CubeCell HTCC-AB02 debe ser configurado mediante el método ABP (Activation By Personalization), te permite hacer la conexión sin la necesidad de hacer un proceso de autenticación automático con la red, tal y como se representa en la Figura 33. En el sketch se configuran manualmente los parámetros DevAddr, NwkSKey y AppSKey, como se refleja en la Figura 34. Una vez cargado el programa, el dispositivo envía los datos al Gateway, que transmite a los servidores de TTN a través de Internet. El proceso completo de configuración y carga del código se encuentra en Anexo 1.

```

/*autenticacion por ABP*/
uint8_t nwkSKey[] = {0x5A, 0x30, 0x1B, 0x99, 0x80, 0xB5, 0x28, 0xA5, 0x18, 0xB8, 0x71, 0x6D, 0xF2, 0x7F, 0x9F, 0xC0};
uint8_t appSKey[] = { 0x65, 0xFC, 0x1D, 0x70, 0xC8, 0x59, 0xDF, 0xE6, 0xDE, 0xF7, 0xE1, 0x24, 0x2F, 0x06, 0x2C, 0x1C };
uint32_t devAddr = (uint32_t)0x260c222b;

```

Figura 33 Código de inicialización de dispositivo LoRa

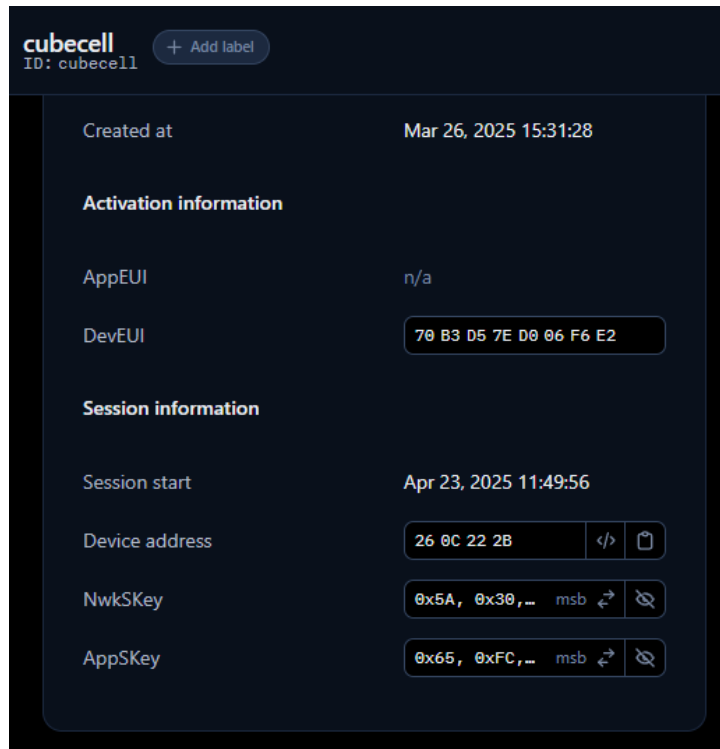


Figura 34 Información de activación y sesión para el dispositivo CubeCell

3.3.6. Despliegue de la plataforma Node-Red para la gestión y visualización integral de datos.

Una vez que se reciben los datos en el servidor IoT, se requiere un método con el cual se comunique a las aplicaciones que se usan de los datos, TTN soporta dos protocolos de comunicación en su capa de transporte: MQTT y AMQP. Sin embargo, TTN se configura por defecto con un broker MQTT basado en una implementación de Eclipse Mosquitto. Este enfoque simplifica enormemente el flujo de extracción de datos, ya que el broker ya está configurado y solo se necesitaría implementar un cliente para utilizar los datos que recibe la aplicación. Para configurar un cliente bróker se realizaron los siguientes pasos:

1. Se realizó la descarga e instalación de mosquitto desde la página oficial <https://mosquitto.org/download/>
2. El segundo paso consiste en identificar las credenciales de acceso y la dirección IP

del servidor MQTT proporcionado por TTN para aplicaciones IoT. Para conectarse a dicho servidor, es necesario generar una nueva clave API que actúe como contraseña de autenticación, además de disponer de la dirección del servidor `nam1.cloud.thethings.network:1883` y el nombre de usuario `cubecell12@ttn`, como se muestra en la Figura 35.

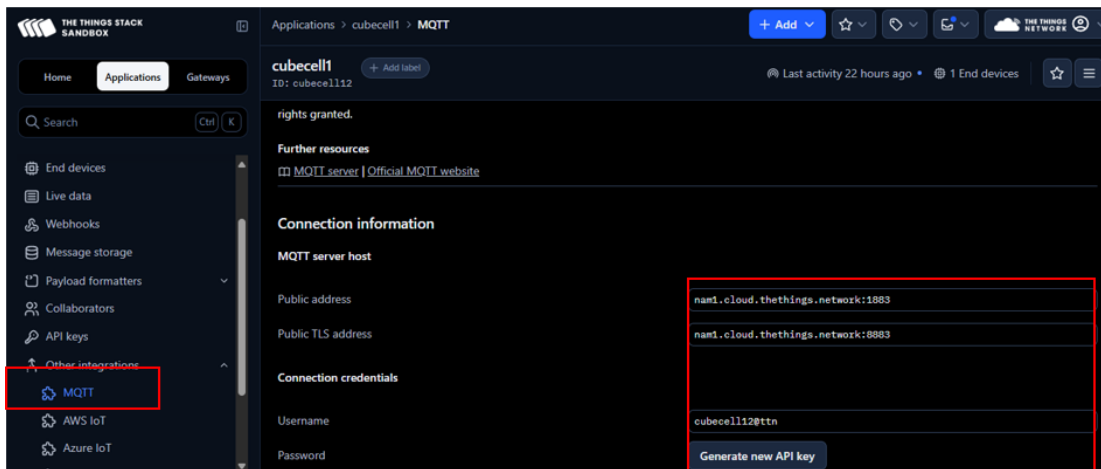


Figura 35 Configuración de conexión MQTT en The Things Network

3. En Node-RED se realizó la configuración del cliente MQTT. Se utilizó el nodo MQTT, que se conecta con el servidor en la nube y permite obtener los datos necesarios. Se debe arrastrar el nodo MQTT, ubicado en la categoría input, hacia el flujo de trabajo, tal como se muestra en la Figura 36.

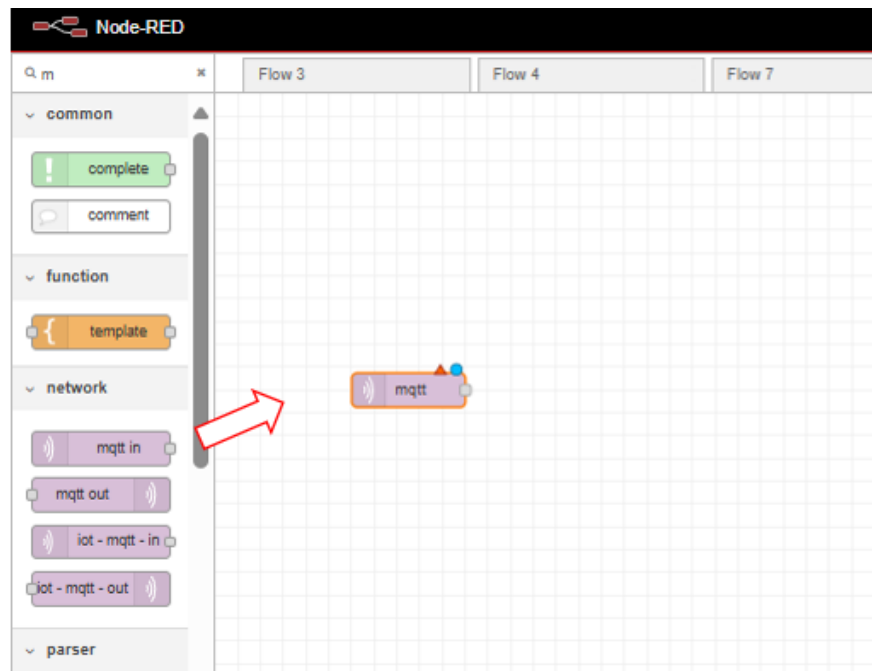


Figura 36 Integración de MQTT en Node-RED para procesamiento de datos IoT

4. Una vez añadido el Node MQTT al entorno de trabajo, se lleva a cabo la configuración del mismo. Node-Red lo simplifica así que solo hay que especificar el servidor MQTT (broker) y el topic asociado. Este nodo, vista como receptor, se encarga a suscribir al topic con el objetivo de recibir los mensajes publicados. Al hacer doble clic sobre el nodo se abre el panel de configuración, donde se debe seleccionar el servidor MQTT presionando el icono del lápiz junto a la opción server, tal como se aprecia en la Figura 37.

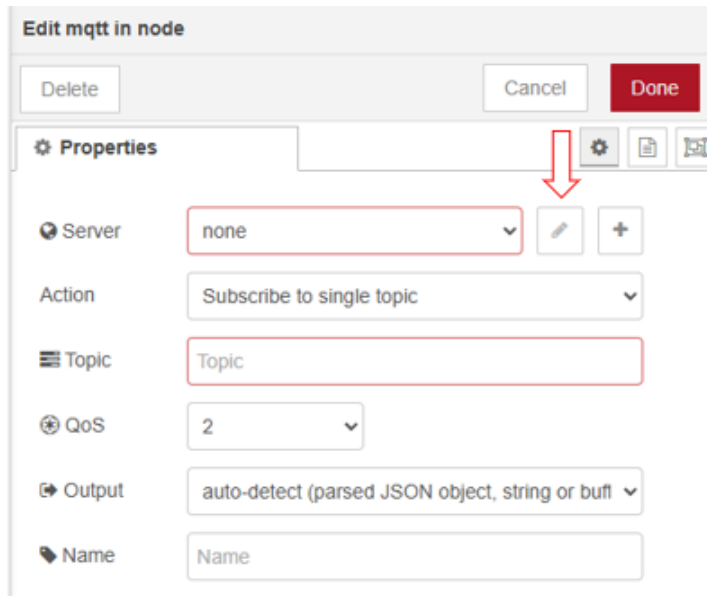


Figura 37 Configuración de propiedades del Nodo MQTT en Node-RED

5. Después se abre un nuevo panel de configuración donde se debe proporcionar un nombre para el bróker MQTT, en este caso sería Cubecell1. En el campo 'Server', se debe acceder con la dirección del servidor `nam1.cloud.thethings.network` y el puerto utilizado, que por defecto es el 1883, como se puede ver en la Figura 38.

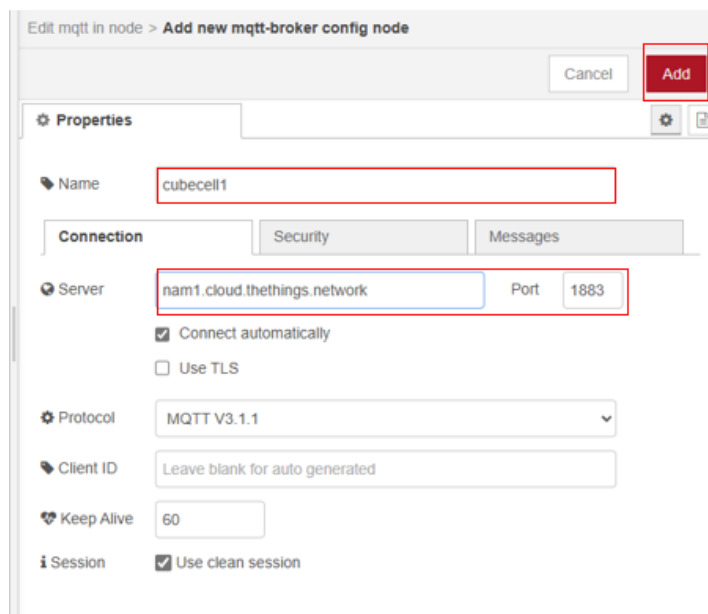
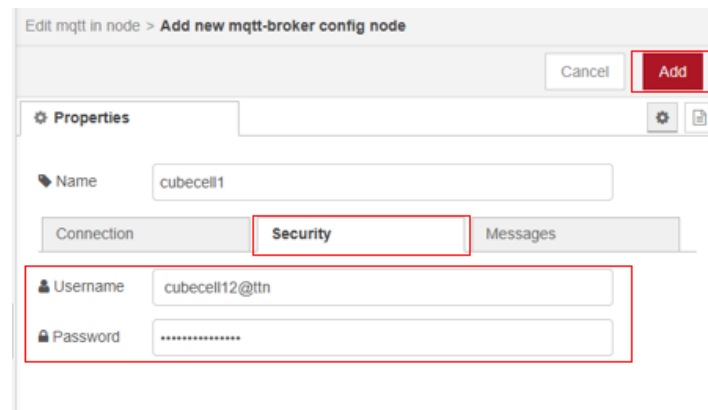


Figura 38 Edición de parámetros del Nodo MQTT Broker para conexión con The Things Network

- Ahora se debe acceder al campo de seguridad donde debemos rellenar los campos el nombre del usuario y la contraseña esto se encuentra en la Figura 39. No olvidar hacer clic en el botón 'Add' para confirmar los cambios.



The screenshot shows a web interface for configuring an MQTT broker. The title bar reads "Edit mqtt in node > Add new mqtt-broker config node". At the top right, there are "Cancel" and "Add" buttons. Below the title bar is a "Properties" section with a gear icon and a refresh icon. The "Name" field contains "cubecell1". Below this are three tabs: "Connection", "Security", and "Messages". The "Security" tab is selected and highlighted with a red box. Under the "Security" tab, there are two fields: "Username" with the value "cubecell12@ttn" and "Password" with a masked value "*****". Both the "Security" tab and the "Username" and "Password" fields are highlighted with red boxes.

Figura 39 Asignación de usuario y contraseña en el Nodo MQTT Broker

- Una vez finalizado este paso, el entorno de configuración del nodo MQTT está automáticamente actualizado y muestra el broker MQTT que se configuró anteriormente. Finalmente se debe introducir el campo 'Topic' considerando la versión de LoRa, el ID de aplicación y el ID de dispositivo. En este caso, el tópico configurado es: v3/cubecell12@ttn/devices/cubecell/up.", como se muestra en la Figura 40.

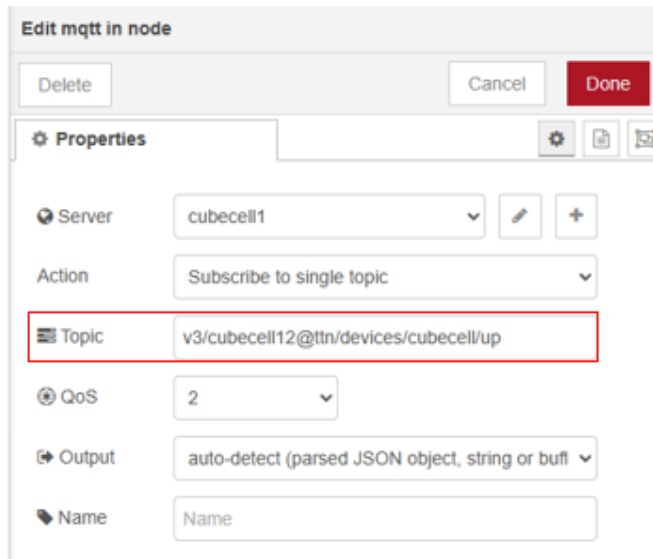


Figura 40 Configuración del Topic MQTT para la comunicación ascendente en TTN

- Una vez completada la configuración se procede a verificar el sistema. Se incorporó un nodo DEBUG con el propósito de confirmar la recepción de los mensajes. La información recibida incluye diversos datos; en este caso específico, se trabaja principalmente con el objeto **decoded_payload**, del cual se extraen los valores individuales, tal como se muestra en la Figura 41.



Figura 41 Visualización de datos en Node-RED mediante MQTT

3.3.7. Integración de Node-RED con Telegram para el envío de notificaciones

Para facilitar que el sistema alerte eventos de manera autónoma para el usuario final se hizo una integración de Node-RED con Telegram. Este proceso se ejecutó utilizando un bot de Telegram y su configuración en el entorno de Node-RED. El desarrollo que se realiza se muestra a continuación:

1. Creación del bot en Telegram

El primer paso fue crear un bot a través del sitio BotFather de Telegram. Para ello, se buscó el usuario @BotFather dentro de la aplicación. Una vez abierto el chat, se ejecutó el comando /start. Se desplegó un conjunto de opciones, como se muestra en la figura 42 correspondiente. Posteriormente, se seleccionó la opción /newbot para iniciar el proceso de creación. Durante este procedimiento, se solicitó asignar un nombre y un identificador único para el bot. Al finalizar, BotFather generó un token de autenticación, el cual sería utilizado más adelante para establecer la conexión entre el bot y la plataforma Node-RED ,tal como se observa en la Figura 43.



Figura 42 Comandos de configuración de Bots en Telegram mediante BotFather



Figura 43 Configuración del nombre y token de acceso para un Bot de Telegram

2. Instalación del paquete de Telegram en Node-RED

En el entorno Node-RED se tuvo que instalar el repositorio Node-Red-contrib-telegrambot que trae los nodos necesarios para la integración. Esta instalación se llevó a cabo desde el menú de configuración de Node-RED, accediendo a la pestaña Manage palette y buscando dicha paleta para su instalación, como se muestra en Figura 44.

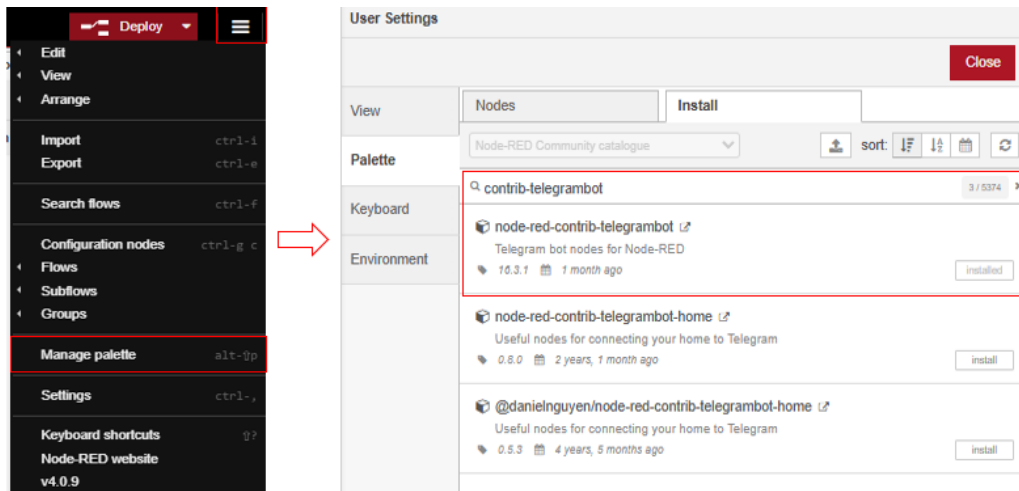


Figura 44 Instalación del paquete de Telegram en Node-RED

3. Configuración del nodo Telegram Sender

Después de la instalación del paquete, se agregó al flujo un nodo del tipo telegram sender. Se incluyó el token del bot y el chat ID que se habían obtenido previamente en su configuración. Gracias a esta configuración se logró conectar Node-RED con Telegram y así posibilitar el envío de mensajes, tal como se ilustra en la Figura 45.

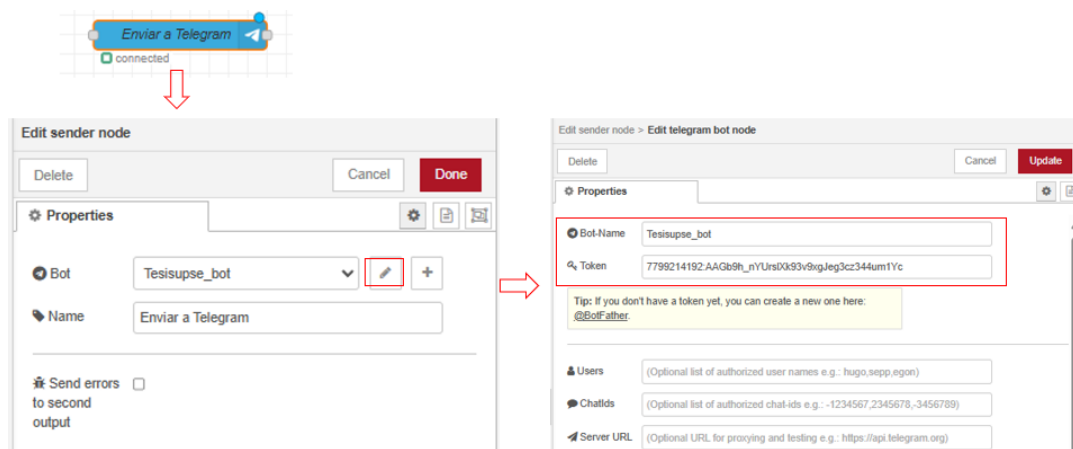


Figura 45 Configuración del nodo Telegram Sender

4. Diseño del flujo de notificación

Se implementó un flujo en la plataforma Node-RED con el propósito de enviar notificaciones automáticas cuando se detectara el nivel de llenado del contenedor. Estos

mensajes eran generados en formato de texto y dirigidos al nodo telegram sender, el cual se encargaba de enviarlos al usuario final mediante el bot previamente configurado en la aplicación Telegram, tal como se refleja en la Figura 46.

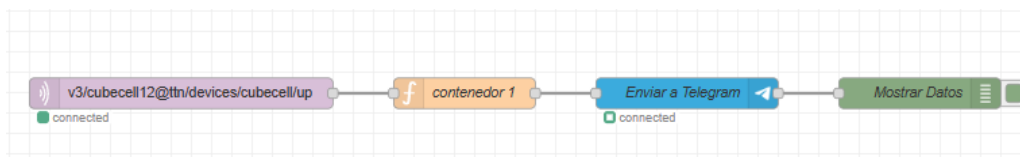


Figura 46 Diagrama de flujo en Node-RED: Transmisión de datos IoT a Telegram

5. Prueba y verificación

Por último, se llevaron a cabo pruebas funcionales mediante el envío de mensajes de prueba desde Node-RED al bot de Telegram. Se corroboró que las notificaciones llegaran adecuadamente al usuario destinatario, asegurando de este modo la eficacia del sistema, tal como se presenta en la Figura 47.

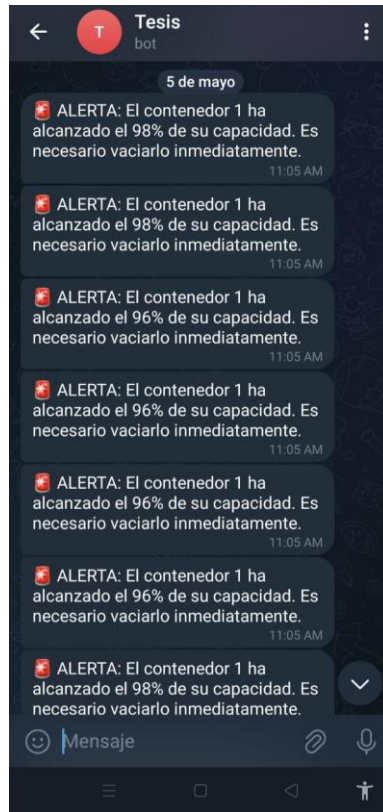


Figura 47 Envío de notificaciones de alerta a Telegram

3.3.8. Integración y visualización de datos en el nodo Worldmap de Node-RED

Para visualizarla gráficamente la posición del contenedor y mostrar los datos en tiempo real obtenidos por el sistema, se incorporó éste el Nodo Worldmap dentro de la plataforma Node-RED. Se describe a continuación el procedimiento utilizado para su integración:

1) Instalación del nodo Worldmap

Se ingresó al menú de gestión de paletas de Node-RED y se instaló el paquete node-red-contrib-web-worldmap. Este paquete incorpora los nodos necesarios para la visualización geográfica sobre mapas interactivos, tal y como se aprecia en la Figura 48.

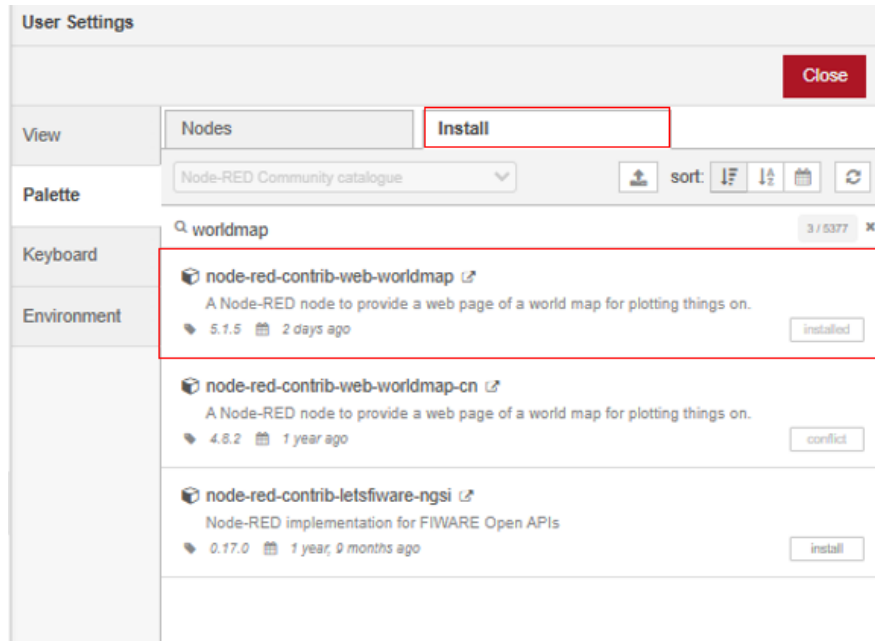


Figura 48 Instalación del nodo Worldmap

2) Incorporación del nodo al flujo

Con la instalación finalizada, se agregó el nodo worldmap al flujo de trabajo. Este nodo permite una interfaz web para mostrar marcadores en coordenadas geográficas, como se muestra en Figura 49.

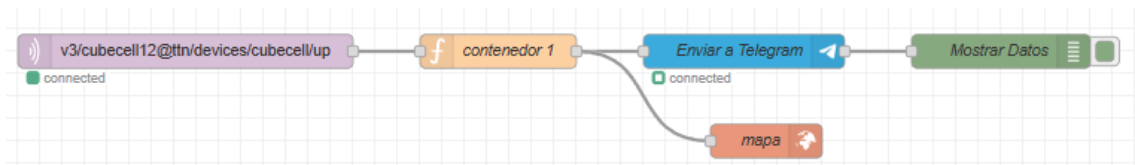


Figura 49 Integración del nodo worldmap en el diagrama de flujo

3) Configuración básica del nodo Worldmap

En la configuración del nodo se estableció la dirección para acceder al worldmap, en

este caso definida como **/mapa**, junto con el nivel de zoom y la posición inicial que debe mostrar el mapa al cargarse en el navegador, como se evidencia en la Figura 50.

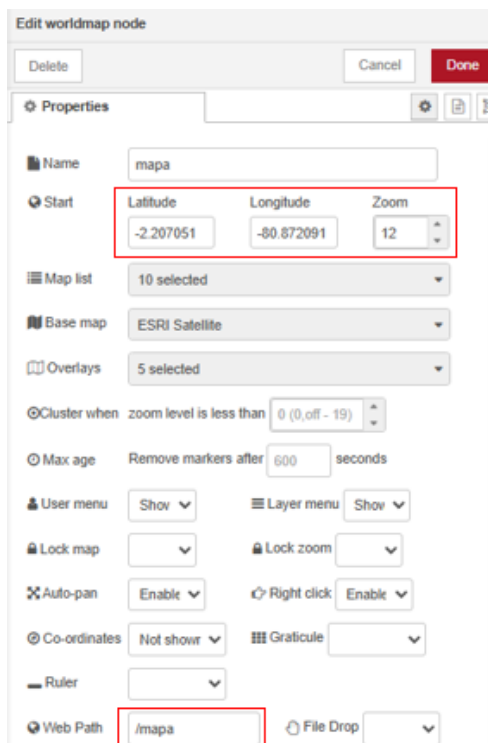


Figura 50 Personalización de parámetros de mapa en Node-RED para visualización geográfica

4) Visualización en tiempo real

Finalmente, para acceder al mapa, desde una página web fue ingresado la dirección correspondiente al servidor Node-RED, a través de la ruta configurada: <http://localhost:1880/worldmap>. A través de esta interfaz se podía observar en tiempo real la ubicación del contenedor, así como su estado de operación.

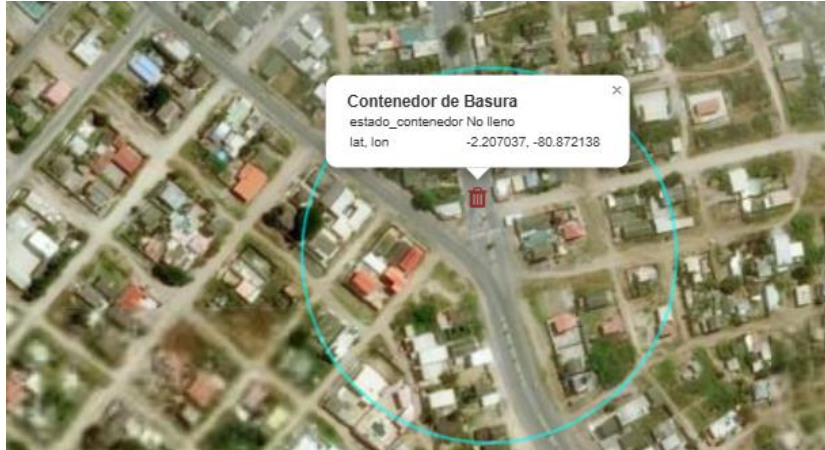


Figura 51 Visualización de datos en la plataforma de Worldmap

3.4. Diagrama de flujo completo implementado en Node-Red

El flujo creado en Node-RED, como se muestra en la Figura 52. tiene como objetivo monitorear el nivel de llenado de dos contenedores, utilizando datos enviados por un dispositivo CubeCell AB-02, el cual se encuentra conectado a la red LoRaWAN a través de The Things Network (TTN). El sistema recibe esta información mediante el protocolo MQTT y procesa los datos de cada contenedor de forma individual, lo que permite verificar el estado específico de cada uno. Una vez analizada la información, los datos se muestran en la consola para facilitar su revisión y también se representan en un mapa, lo que contribuye a una supervisión geográfica más intuitiva. Además, en caso de que se detecte que el nivel de llenado supera un umbral predefinido, el sistema genera automáticamente una notificación que es enviada al usuario mediante un bot de Telegram. De esta manera, se posibilita una gestión remota, eficiente y en tiempo real del estado de los contenedores. En el Anexo 2 se incluye el archivo. json correspondiente al diagrama de flujo completo implementado en Node-RED.

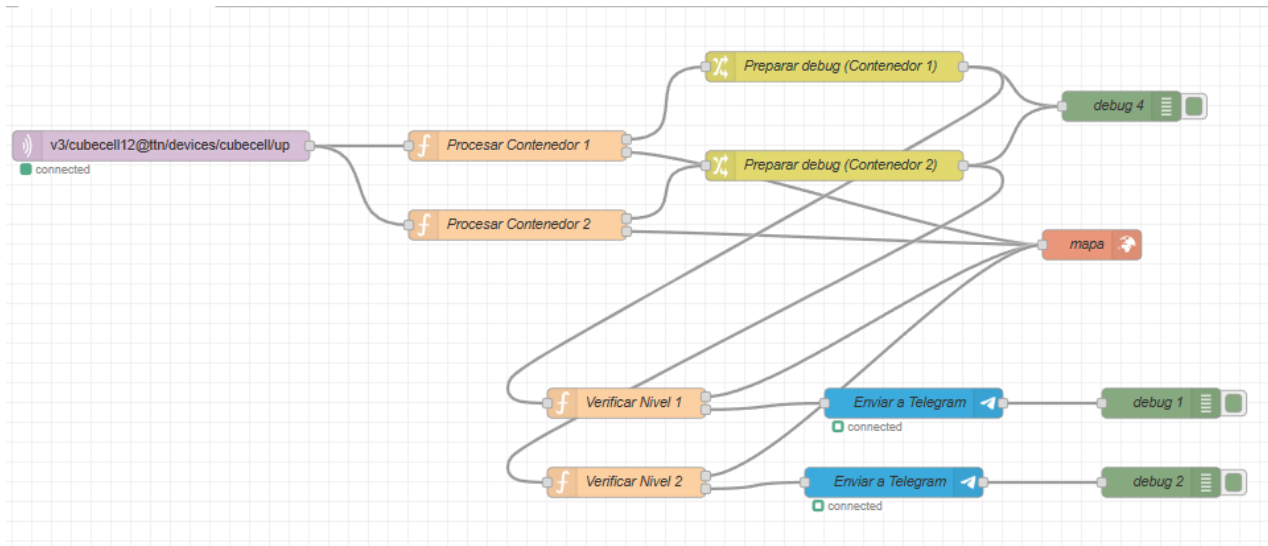


Figura 52 Diagrama de flujo completo implementado en Node-Red

CAPÍTULO IV

4.1. Análisis de resultados

4.2. Prototipo del sistema de monitoreo de nivel de llenado

El resultado es un prototipo funcional que integra sensores especializados para detectar y medir los niveles de residuos en los contenedores de basura. Como se muestra en la Figura 53, el sistema proporciona información en tiempo real sobre el estado de llenado de los contenedores, promoviendo una gestión más eficiente de los residuos y optimizando las rutas de recogida.

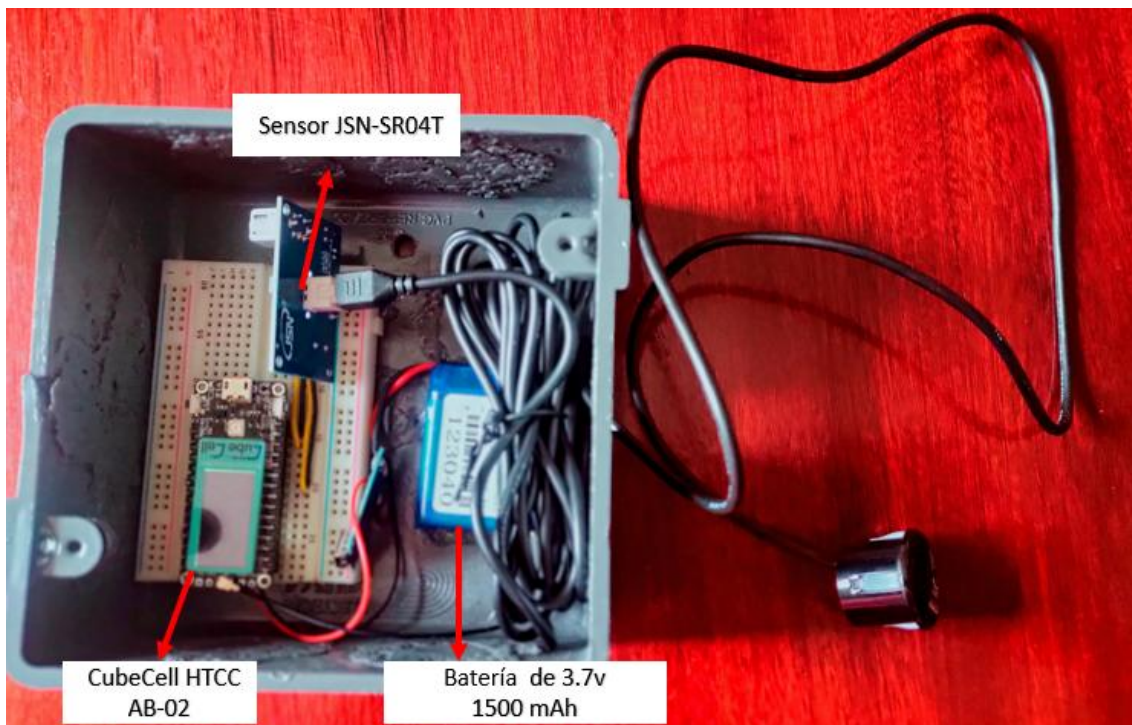


Figura 53 Prototipo de sistema de nivel de llenado basado en LoRa

Cálculo de consumo energético

Con el fin de garantizar la máxima autonomía está previsto apreciar el consumo energético en cada uno de los elementos y sus varios estados de funcionamiento. Los resultados obtenidos aparecen en la Tabla 10.

Tabla 10. Consumo total de corriente del prototipo propuesto

Componentes	Consumo
CubeCell HTCC AB-02	15 mA
JSN-SR04T	20 mA
Total estimado	35 mA

Para determinar la autonomía total del sistema en caso de operar únicamente en modo de transmisión continua, se utiliza una batería de 3.7 V y 1500 mAh. La autonomía se obtiene mediante:

$$Autonomia (h) = \frac{Capacidad\ de\ la\ bateria\ (mAh)}{Consumo\ del\ dispositivo\ (mA)}$$

$$Autonomia (h) = \frac{1500 (mAh)}{35 (mA)}$$

$$Autonomia (h) = 42.85 (h)$$

Por lo tanto, si el sistema transmite datos continuamente, la duración de la duración batería es de aproximadamente 42.85 horas. Este cálculo representa el consumo máximo de energía posible y ayuda a determinar la duración mínima de la batería. Para garantizar una carga segura y proteger la batería de daños, se utiliza un módulo TP4056, para supervisar la carga y prevenir problemas como sobrecargas o cortocircuitos.

4.3. Pruebas de transmisión de datos a TTN

Una vez que se ha realizado la configuración de los dispositivos en el servidor, tanto del Gateway como de los nodos finales, y se han definido los parámetros de activación mediante el mecanismo ABP es posible visualizar el flujo de datos proveniente del nodo hacia el servidor.

Prueba 1:

La función `prepareTxFrame()` configura el payload para transmisión LoRaWAN, definiendo un tamaño de 4 bytes mediante `appDataSize = 4` y asignando valores secuenciales (0x00, 0x01, 0x02, 0x03) al buffer `appData`; como se muestra en la Figura 54, en aplicaciones reales, estos valores serían reemplazados por lecturas de sensores optimizando así el consumo energético y el tiempo de transmisión del dispositivo IoT.



```
81     appDataSize = 4;
82     appData[0] = 0x00;
83     appData[1] = 0x01;
84     appData[2] = 0x02;
85     appData[3] = 0x03;
86 }
87
88
```

Salida Monitor Serie X

Mensaje (Intro para mandar el mensaje de 'CubeCell-Board Plus (HTCC-AB02)' a 'COM3')

```
unconfirmed uplink sending ...
unconfirmed uplink sending ...
unconfirmed uplink sending ...
unconfirmed uplink sending ...
unconfirmed uplink sending ...
```

Figura 54 Transmisión de datos por LoRaWAN desde CubeCell HTCC-AB02

La interfaz de monitorización LoRaWAN muestra los mensajes de comunicación donde los datos "00 01 02 03" enviados desde el dispositivo con la dirección "26 0C 22 2B" coinciden tal como se visualiza en la Figura 55, con los valores configurados dentro de la función `prepareTxFrame()`.

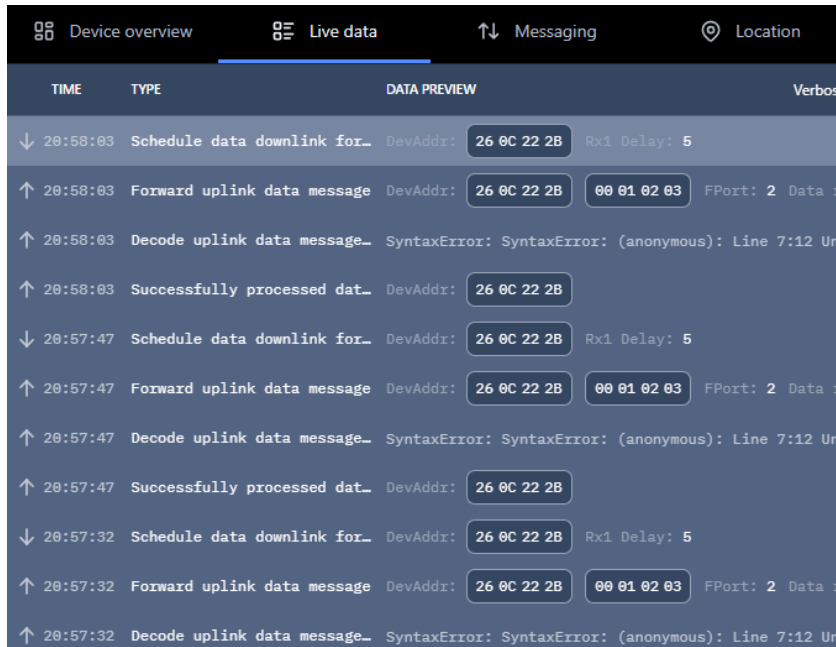


Figura 55 Registro de datos en TTN

Verificación de transmisión y recepción de paquetes LoRaWAN

Durante el proceso de validación, los paquetes enviados desde la puerta de enlace a la red LoRaWAN se verificaron mediante la plataforma The Things Network (TTN). La Figura 56 muestra el registro de la puerta de enlace, que registra un total de 977 paquetes, de los cuales 522 se recibieron (RX) y 455 se enviaron (TX) al servidor de red. Esto indica una conectividad suficiente y una conexión operativa.

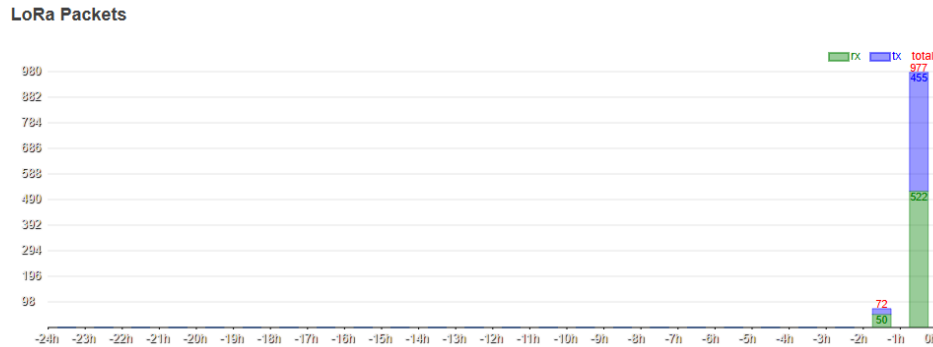


Figura 56 Visualización del tráfico de paquetes LoRa en el Gateway

La Figura 57 muestra la interfaz TTN, que refleja la recepción continua de mensajes del nodo sensor. Los datos recibidos indican el estado del contenedor con la etiqueta "Contenedor lleno" y una tasa de llenado del 98 %. Parámetros técnicos como RSSI, SNR y velocidad de datos confirman la correcta calidad de la señal durante la transmisión.

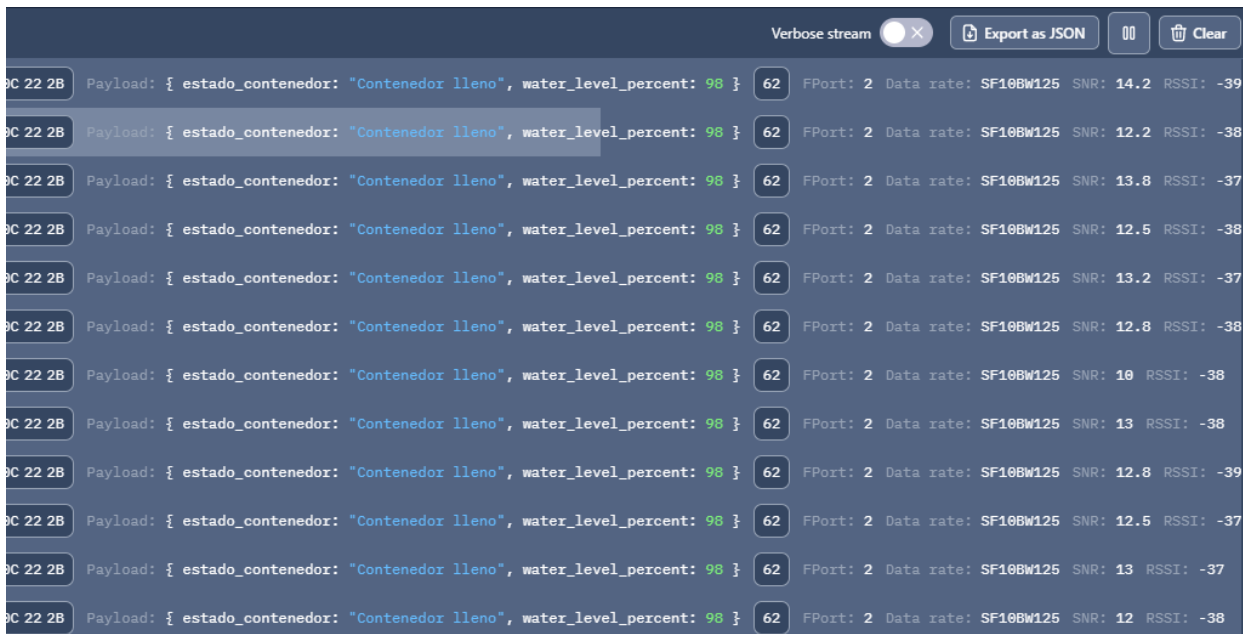
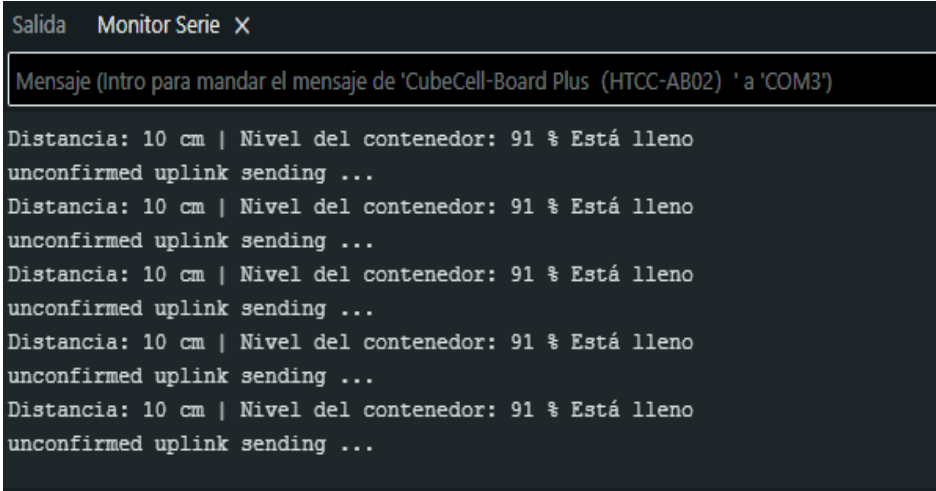


Figura 57 Recepción de paquetes en la consola de The Things Network

Prueba 2: Transmisión de datos de nivel de llenado desde nodos LoRaWAN al servidor

Para comprobar el correcto funcionamiento del nodo y su capacidad para generar datos dentro de los rangos previstos, se ha verificado el comportamiento del dispositivo a través del monitor serial del entorno de desarrollo Arduino, permitiendo observar en tiempo real la transmisión de los datos de nivel de llenado y su envío mediante uplink . La Figura 58, muestra una serie de lecturas continuas con una distancia constante de 10 cm, correspondiente a un nivel del contenedor del 91 %, lo que indica que se encuentra lleno.

The image shows a screenshot of the Arduino Serial Monitor window. The title bar reads "Salida Monitor Serie X". Below the title bar, there is a text input field containing the message "Mensaje (Intro para mandar el mensaje de 'CubeCell-Board Plus (HTCC-AB02)' a 'COM3')". The main area of the monitor displays a series of five identical lines of text, each representing a data transmission. Each line consists of two parts: a status message and a data message. The status message is "Distancia: 10 cm | Nivel del contenedor: 91 % Está lleno" and the data message is "unconfirmed uplink sending ...".

```
Salida Monitor Serie X
Mensaje (Intro para mandar el mensaje de 'CubeCell-Board Plus (HTCC-AB02)' a 'COM3')
Distancia: 10 cm | Nivel del contenedor: 91 % Está lleno
unconfirmed uplink sending ...
Distancia: 10 cm | Nivel del contenedor: 91 % Está lleno
unconfirmed uplink sending ...
Distancia: 10 cm | Nivel del contenedor: 91 % Está lleno
unconfirmed uplink sending ...
Distancia: 10 cm | Nivel del contenedor: 91 % Está lleno
unconfirmed uplink sending ...
Distancia: 10 cm | Nivel del contenedor: 91 % Está lleno
unconfirmed uplink sending ...
```

Figura 58 Monitor Serie: Registro de datos de nivel de contenedor

Cuando los datos llegan al servidor, los bytes están codificados en hexadecimal y no son legibles, por lo que para ser útiles necesitan ser decodificados. Las funciones que nos proporciona TTN nos facilitan la creación de funciones JavaScript habituales para decodificar los mensajes provenientes de los dispositivos vinculados a la aplicación. Estas funciones se denominan "formatos de pago". Tal como se visualiza en la Figura 59, para decodificar el conjunto de bytes que estamos generando en los nodos, la función de decodificación se presenta de la siguiente forma.



```
1 // Payload formatter for distance in cm
2 function decodeUplink(input) {
3   if (input.bytes.length !== 2) {
4     return {
5       errors: ["Invalid payload length"]
6     };
7   }
8
9   const distance = (input.bytes[0] << 8) | input.bytes[1];
10
11   return {
12     data: {
13       distance_cm: distance,
14       nivel_contenedor: distance <= 60 ? "Medio lleno" : (distance <= 90 ? "Lleno" : "Vacio")
15     }
16   };
17 }
18
```

Figura 59 Función de decodificación de Payload para la medición del nivel de llenado

Una vez decodificados los datos, se puede comprobar que la información recogida por el monitor serie se coincide con los datos en tiempo real suministrados por el servidor TTN. En la Figura 60, se puede observar que el payload decodificado muestra el estado del contenedor como "lleno" y un nivel de agua del 91%, lo cual confirma la correcta transmisión e interpretación de los datos por parte de la red. Este procedimiento de decodificación asegura que los valores, inicialmente enviados en forma de arreglo de bytes en formato hexadecimal, se transformen en valores legibles y que estén en relación con las mediciones efectuadas por

los nodos.

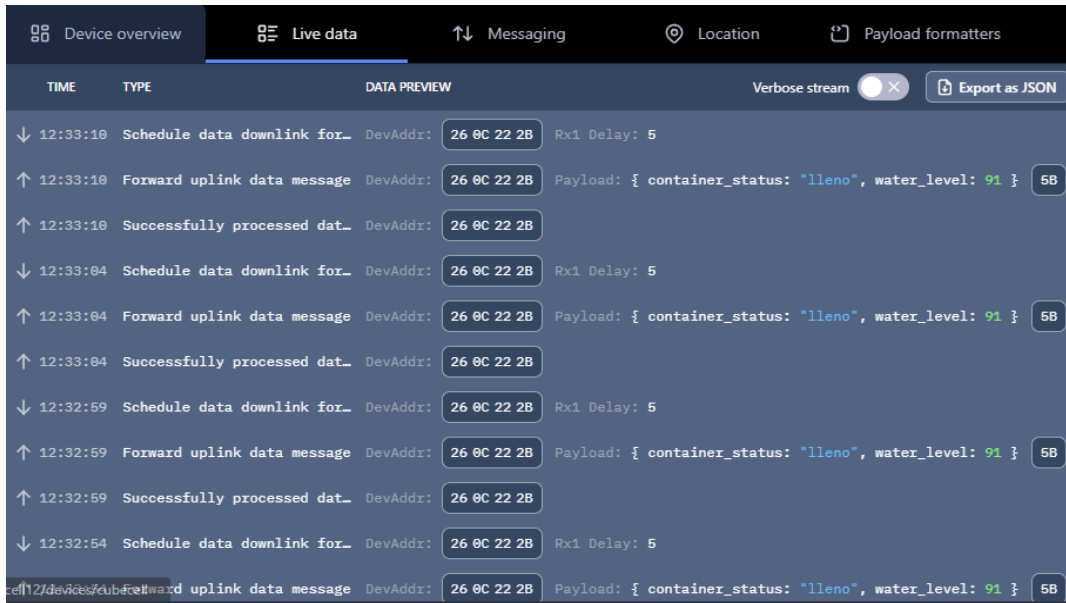


Figura 60 Monitorización del estado de contenedores en tiempo real

4.4. Resultados obtenidos en la calibración del sensor de nivel de llenado

4.4.1. Resultados obtenidos con el sensor ultrasónico JSN-SR04T

Se comprobó que el sensor ultrasónico se encuentra calibrado para una distancia de 1 metro, lo que corresponde al 0 % de capacidad, es decir, un contenedor completamente vacío. En la Figura 61, se muestra la aplicación del sensor ultrasónico JSN-SR04T, donde se evidencia la medición precisa de la distancia.



Figura 61 Resultado de simulación de un contenedor vacío de 1 m

Cuando el nivel de residuos en el contenedor alcanza o supera el 90 % de su capacidad, se considera que está lleno. Esto corresponde a una distancia medida por el sensor de aproximadamente 0,8 metros desde el sensor hasta la superficie de los residuos, considerando que un contenedor vacío presenta una distancia de 1 metros, según se muestra en la Figura 62.



Figura 62 Resultado de simulaci3n de un contenedor lleno

4.5. An谩lisis de datos de monitoreo y optimizaci3n de rutas de recolecci3n

4.5.1. Detecci3n y registro del nivel de llenado en contenedores inteligentes

Se puede observar que el contenedor 1, ubicado en las coordenadas -2.203461, -80.872671, tiene dos estados: cuando est谩 vaci3 (0 % de capacidad), su icono se muestra en rojo en Node-RED a trav茅s del Worldmap, como se ilustra en la Figura 63; mientras que, cuando est谩 lleno, el icono se muestra en verde, indicando que el contenedor ha alcanzado su capacidad total, como puede apreciarse en la Figura 64.

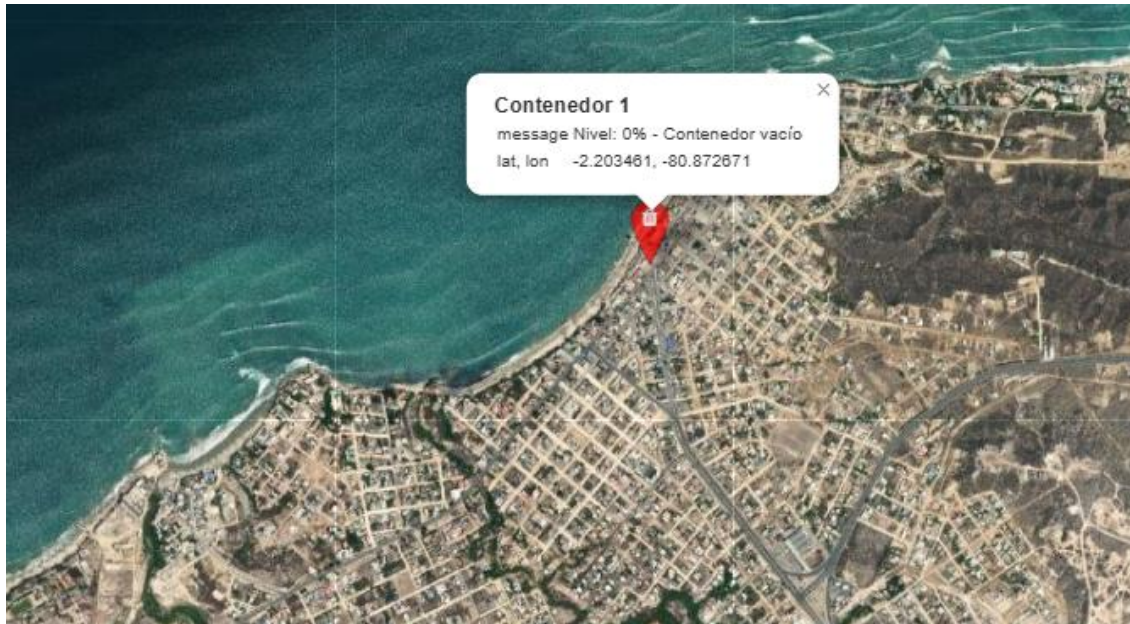


Figura 63 Visualización geográfica del contenedor 1 con nivel de llenado en 0%

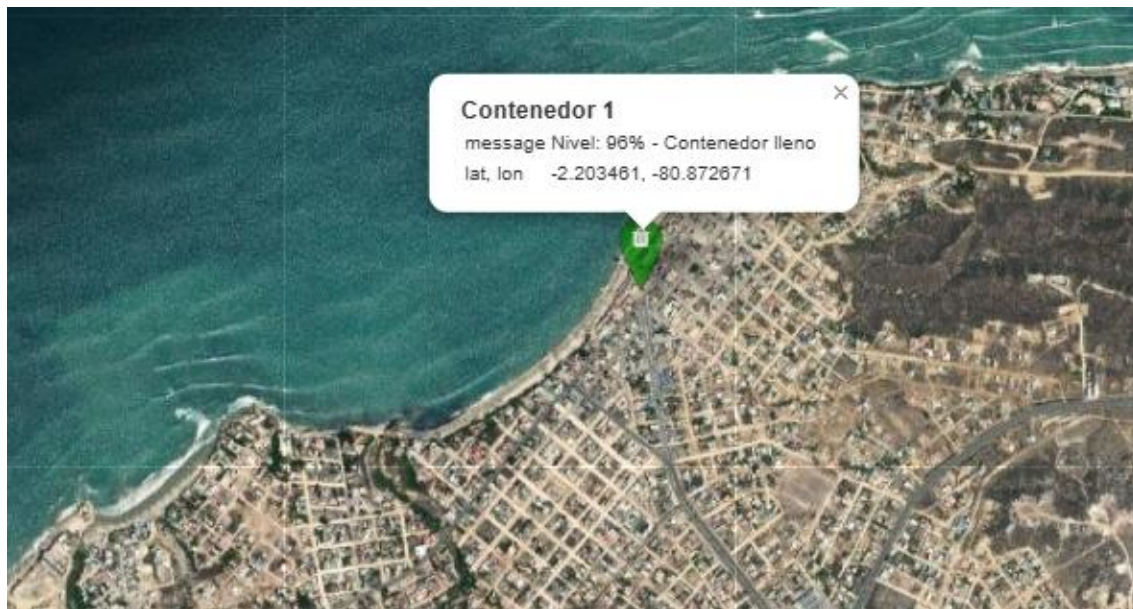


Figura 64 Visualización geográfica del contenedor 1 con nivel de llenado en 96%

De igual forma, el contenedor 2 con coordenadas -2.201910, -80.871411 utiliza el mismo sistema de monitoreo para indicar su nivel de llenado. En la Figura 65 se visualiza el estado del contenedor con un nivel de 0%, es decir vacío. En cambio, en la Figura 66, se muestra el contenedor con un nivel de 95%, lleno. lo que demuestra el correcto

funcionamiento del sistema en diferentes estados del contenedor.

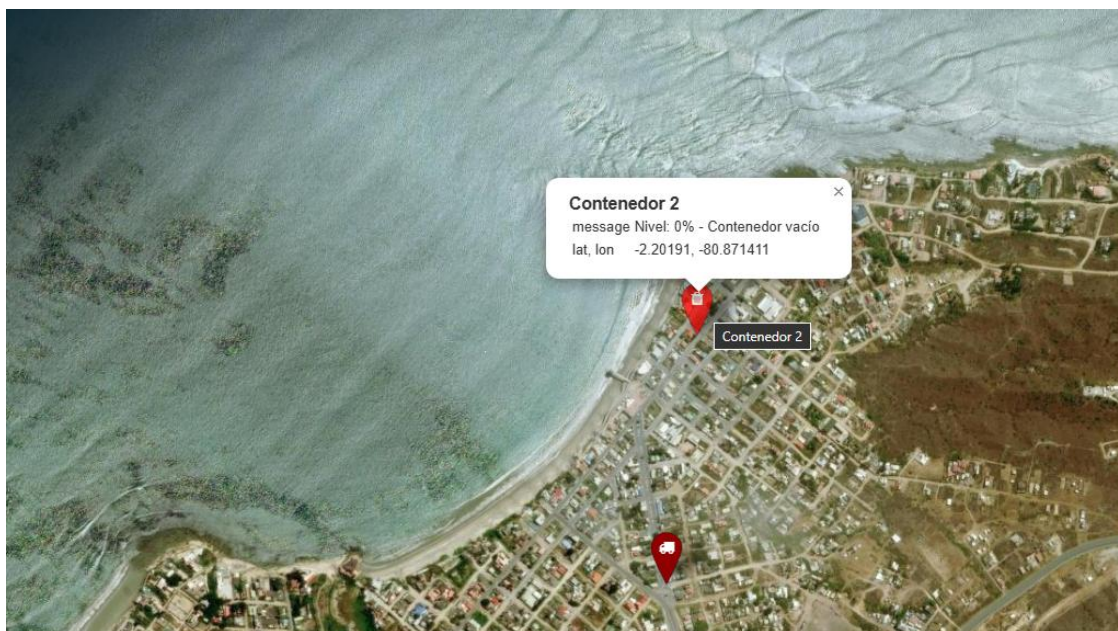


Figura 65 Visualización geográfica del contenedor 2 con nivel de llenado en 0%

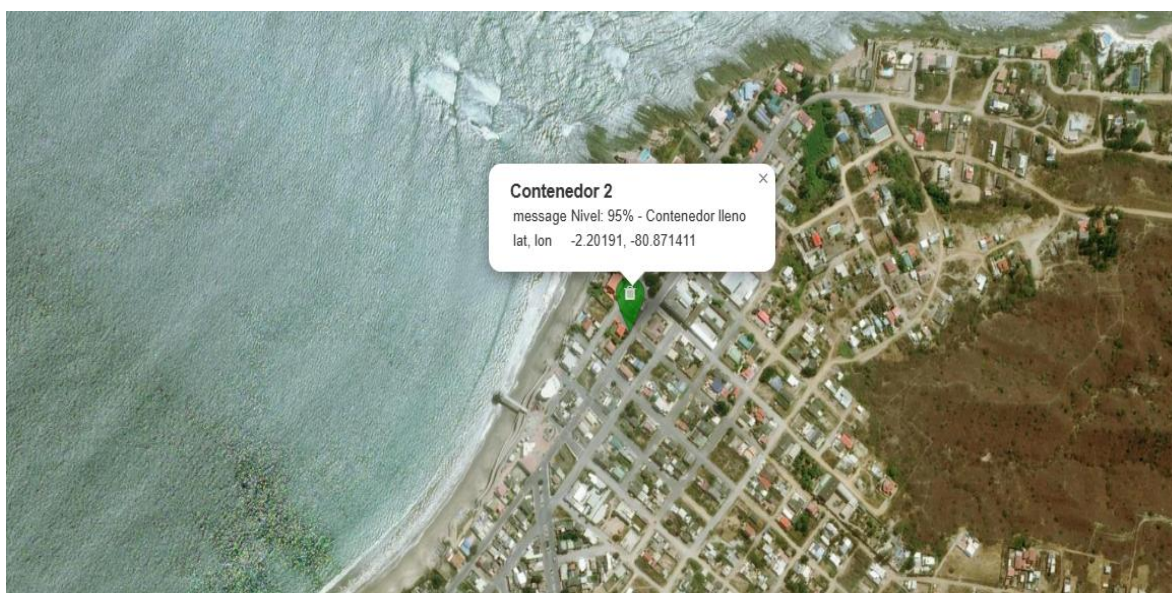


Figura 66 Visualización geográfica del contenedor 2 con nivel de llenado en 95%

4.5.2. Determinación del recorrido óptimo

La generación de rutas se basa en el nivel de llenado de los contenedores, considerando únicamente aquellos que supera el 90 % de su capacidad . El algoritmo de enrutamiento parte de la ubicación inicial del camión y calcula una ruta que incluye únicamente los contenedores llenos, marcados con un icono verde. Si solo el contenedor 1 está lleno, el sistema genera una ruta , como se muestra en la Figura 67. Si solo el contenedor 2 está lleno, genera una ruta diferente, como se muestra en la Figura 68.

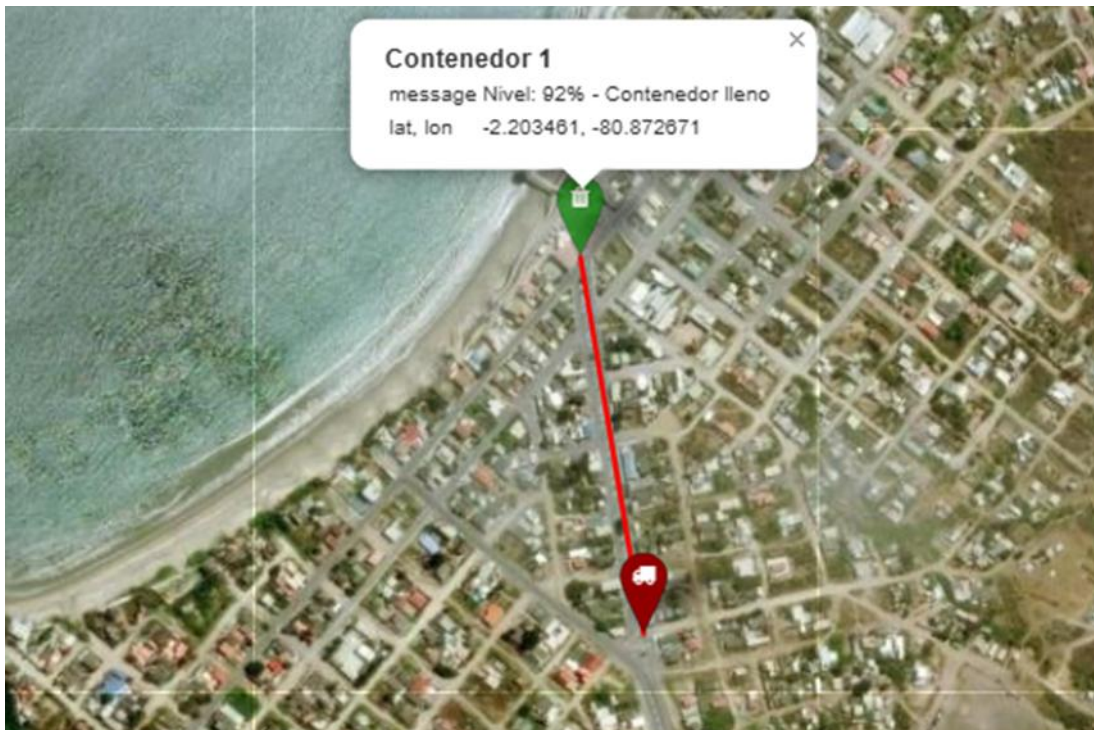


Figura 67 Generación de la ruta óptima del contenedor 1



Figura 68 Generación de la ruta óptima del contenedor 2

Finalmente, si ambos contenedores superan el nivel de llenado al umbral establecido, el algoritmo genera una ruta directa que cubre ambos puntos de recolección, como se muestra en la Figura 69. Una vez recolectados los residuos, la ruta generada previamente se elimina automáticamente, lo que significa que el camión recolector ha vaciado el contenedor. De esta forma, el sistema garantiza que los vehículos sigan rutas optimizadas, centrándose únicamente en los contenedores que deben vaciarse, evitando rutas innecesarias y mejorando considerablemente la eficiencia operativa.



Figura 69 Generación de la ruta óptima de ambos contenedores

4.6. Sistema de notificación de alertas mediante Telegram en la gestión de residuos

El sistema de monitoreo de llenado de contenedores incluye un bot de alerta en la plataforma Telegram que notifica automáticamente cuando un contenedor supera el 90 % de su capacidad y necesita vaciarse. Como se muestra en la Figura 70, estas notificaciones son en tiempo real y generan las URL con dirección a la ubicación geográfica de los contenedores a través de Google Maps, junto con sus respectivas coordenadas. Permiten generar rutas optimizadas para la recolección, lo que permite responder con rapidez, evitar desbordamientos y tomar decisiones operativas para el personal.

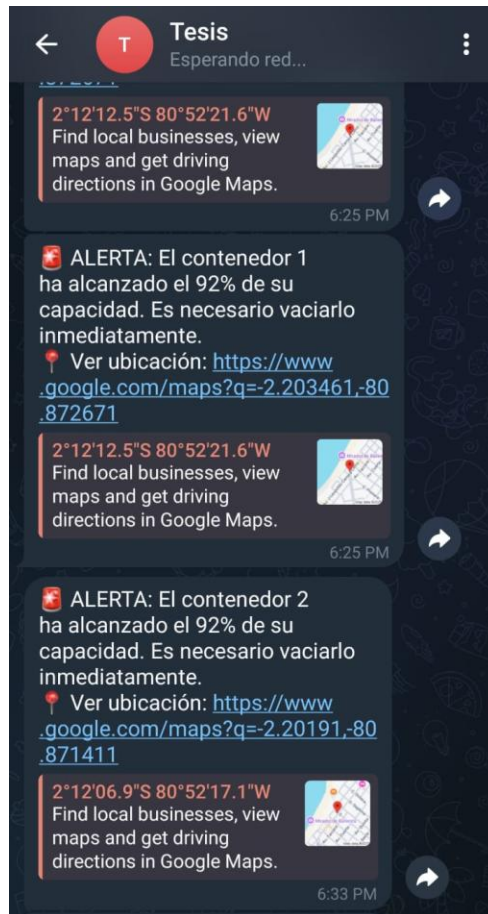


Figura 70 Notificación sobre contenedores llenos en Telegram

4.7. Visualización de datos de llenado en la consola de depuración

La interfaz de depuración de Node-RED se utiliza para visualizar los datos recibidos por los sensores de nivel instalados en contenedores inteligentes. Como se muestra en la Figura 71 en la parte superior se encuentran dos objetos JSON relacionados a los contenedores 1 y 2, con un nivel de llenado de 96 %. Además, se muestran los mensajes generados y enviados al bot de Telegram, incluyen datos estructurados como el tipo de mensaje, el ID del chat . Con esta visualización se puede validar en tiempo real el correcto funcionamiento del sistema, garantizando que los datos recogidos por los sensores sean procesados y transmitidos de manera correcta.

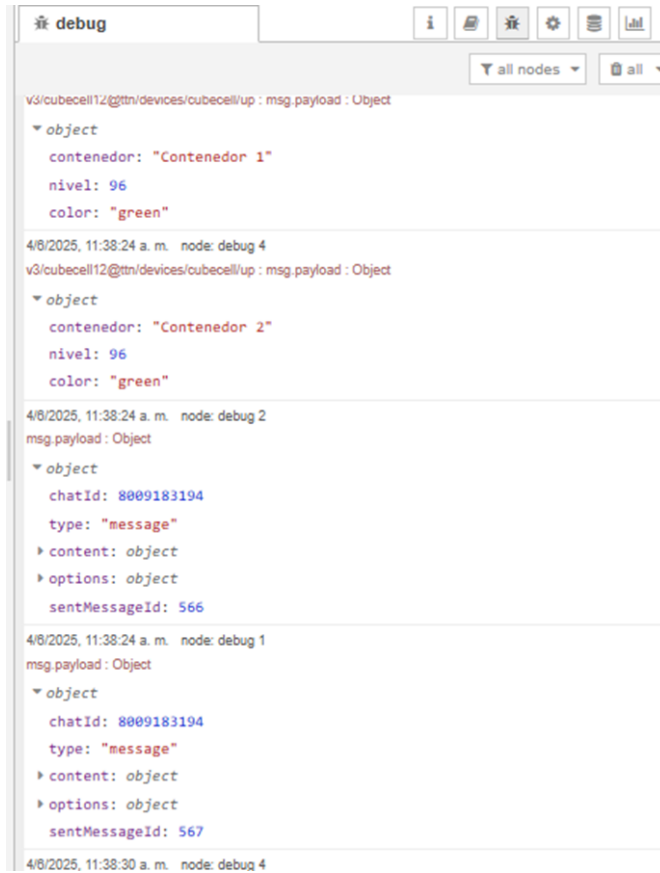


Figura 71 Registro de depuración en Node-RED

CONCLUSIONES

- Se realizó un prototipo de recolección de residuos mediante tecnologías LPWAN e IoT . Mediante sensores ultrasónicos y módulos de transmisión LoRa, se monitoreó remotamente el nivel de llenado de los contenedores .
- Se comprobó que la tecnología LoRa ofrece un buen alcance y rendimiento, permitiendo la transmisión eficiente de datos desde los nodos hacia el Gateway y la plataforma TTN.
- El sistema desarrollado permite la automatización de procesos y la monitorización remota de los niveles de los contenedores mediante plataformas como The Things Network (TTN) y Node-RED. Esta funcionalidad mejora la toma de decisiones y permite una gestión basada en datos en tiempo real.
- La incorporación del nodo Worldmap en Node-RED, se presenta de forma gráfica la ubicación y el estado de los contenedores en tiempo real, optimizando el monitoreo del sistema y mejorando la toma de decisiones con la gestión de residuos.
- La plataforma Node-RED y el servicio de mensajería Telegram permitió el monitoreo automatizado de contenedores y notificaciones instantáneas, facilitando la supervisión del sistema y simplificando la toma de decisiones para el personal relevante.

RECOMENDACIONES

- Aplicar el sistema desarrollado a otros sectores turístico, con el fin de extender los beneficios del monitoreo remoto y en tiempo real a más puntos de recolección, fortaleciendo así la gestión integral de residuos sólidos.
- Es recomendable incorporar más sensores, como de temperatura, humedad o gases, para todas aquellas variables que se deseen monitorear y lograr así una gestión más integral y preventiva de los residuos almacenados en los contenedores.
- Desarrollar una aplicación móvil que muestre el estado de cada contenedor, emita alertas y genere reportes con el fin de facilitar la relación entre el sistema y los usuarios del servicio.
- Incorporar fuentes de energía renovable, como paneles solares, a fin de prolongar la autonomía de los dispositivos y garantizar un funcionamiento continuo sin necesidad de intervención frecuente.
- Establecer un plan de mantenimiento preventivo para los dispositivos utilizados, incluyendo revisiones periódicas y calibraciones de los sensores, a fin de garantizar la precisión de los datos y la operatividad continua del sistema.

BIBLIOGRAFÍA

- [1] Á. Y. Karin Villarroel, “Gestión de residuos urbanos para ciudades más limpias: hacia una economía circular en LAC,” ciudades sostenibles. Accessed: Jun. 23, 2025. [Online]. Available: <https://blogs.iadb.org/ciudades-sostenibles/es/gestion-de-residuos-urbanos-para-ciudades-mas-limpias-hacia-una-economia-circular-en-lac/>
- [2] J. Gutberlet, “Household waste and health risks affecting waste pickers and the environment in low- and middle-income countries,” *Int J Occup Environ Health*, vol. 23, no. 4, pp. 299–310, May 2018, Accessed: Apr. 29, 2025. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC6147112/>
- [3] A. Sáez and J. A. U. G, “Manejo de residuos sólidos en América Latina y el Caribe,” *Revista Omnia*, vol. 20, no. Información Científica, pp. 122–124, May 2019, Accessed: Apr. 29, 2025. [Online]. Available: <http://cimogsys.esPOCH.edu.ec/direccion-publicaciones/public/docs/books/2019-09-18-221533-61%20Libro%20Memorias%20SECTEI%202018-comprimido.pdf>
- [4] MARÍA BETSHABÉ VERA SANTOS, “GESTIÓN DE RESIDUOS SÓLIDOS URBANOS Y SU INCIDENCIA AMBIENTAL EN LOS CANTONES VALENCIA, MOCACHE Y CALUMA.,” UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO, QUEVEDO, 2020. Accessed: Apr. 29, 2025. [Online]. Available: <https://repositorio.uteq.edu.ec/handle/43000/6281>
- [5] García Quendí and Yomaira Liceth, “Diseño de una ruta de recolección para optimizar el servicio de recolección de residuos sólidos por contenedores del GAD Municipal de Tulcán con base en mecanismos de resolución de problemas VRP,” FACULTAD DE COMERCIO INTERNACIONAL, INTEGRACIÓN, ADMINISTRACIÓN Y ECONOMÍA EMPRESARIAL, Tulcán,

2022. Accessed: Apr. 29, 2025. [Online]. Available:
<http://repositorio.upec.edu.ec/handle/123456789/1471>

[6] ALCOCER QUINTEROS Patricio Rubén, CEVALLOS MUNOZ Omar, and KNUDSEN GONZALEZ José, “Mejoramiento de la gestión integral de los residuos sólidos urbanos en el cantón de Quevedo, Ecuador.,” *Revista Universidad y Sociedad*, vol. 11, no. 5, pp. 362–367, Dec. 2019, Accessed: Apr. 29, 2025. [Online]. Available:
http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S2218-36202019000500362&lng=es&nrm=iso&tlng=en

[7] A. Ricatti, “Optimización de recolección de residuos con IoT,” *Universidad Siglo 21*, Buenos Aires, 2017. Accessed: Jun. 23, 2024. [Online]. Available:
<https://repositorio.21.edu.ar/handle/ues21/14049>

[8] López Puyo Francisco and N. D. Pastran Zamora, “Modelo para la gestión de residuos aprovechables incorporando tecnología IoT y servicios en la nube,” *Universidad Distrital Francisco Jose de ca*, Bogota, 2021. Accessed: Apr. 29, 2025. [Online]. Available: <http://hdl.handle.net/11349/28307>

[9] Nathalie Mitton, “Waste Management System Using IoT-Based Machine Learning in University,” vol. 2020, pp. 1–13, Feb. 2020, Accessed: Apr. 29, 2025. [Online]. Available: <https://doi.org/10.1155/2020/6138637>

[10] P Haribabu, Sankit R Kassa, J Nagaraju, R Karthik, N Shirisha, and M Anila, “Implementation of an Smart Waste Management system using IoT ,” *Proceedings of the International Conference on Intelligent Sustainable Systems*, Hyderabad, pp. 1–2, 2017. Accessed: Jun. 23, 2024. [Online]. Available:
<https://ieeexplore.ieee.org/document/8389367>

[11] S.R. Jino Ramson, S. Vishnu, and A. Alfred Kirubaraj, “A LoRaWAN IoT

- enabled Trash Bin Level Monitoring System,” pp. 1–10, 2021. Accessed: Apr. 29, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/9427265>
- [12] Emasa EP, “Recoleccion de desechos ,” Emasa EP. Accessed: Jun. 23, 2025. [Online]. Available: <https://emasaep.gob.ec/recoleccion/>
- [13] Obando Suarez Jean Paul, “sistema de recoleccion de desechos optimizado, mediante IoT en el sector de olivio de la ciudad de ibarra provincia de imbabura,” Universidad Tecnica del Norte, Ibarra, 2022. Accessed: Apr. 29, 2025. [Online]. Available: <http://repositorio.utn.edu.ec/handle/123456789/13118>
- [14] A. Muzlera, S. Antonini, and E. Pettigiani, “Caracterización de residuos sólidos urbanos domiciliarios en Unquillo, Córdoba,” Universidad nacional de Córdoba, 2013. Accessed: Apr. 29, 2025. [Online]. Available: <http://hdl.handle.net/11086/546744>
- [15] Abdullah Addas, Muhammad Nasir Khan, and Fawad Naseer, “Gestión de residuos 2.0: aprovechar el internet de las cosas para una solución de ciudad inteligente eficiente y ecológica,” *journals.plos*, Jul. 2024, Accessed: Jun. 23, 2025. [Online]. Available: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0307608>
- [16] Guinea Mobile, “Arquitectura IoT,” Guinea.
- [17] AWS, “¿Qué es MQTT?,” Amazon. Accessed: Jun. 23, 2024. [Online]. Available: <https://aws.amazon.com/es/what-is/mqtt/>
- [18] AWS, “¿Cuál es la diferencia entre HTTP y HTTPS?,” Amazon. Accessed: Jun. 23, 2024. [Online]. Available: <https://aws.amazon.com/es/compare/the-difference-between-https-and-http/>
- [19] Jin Lin, “An Internet of Energy Things Based on Wireless LPWAN,” *ScienceDirect*, vol. 3, no. 4, pp. 460–466, Aug. 2017, Accessed: Apr. 29, 2025.

[Online]. Available: <https://doi.org/10.1016/J.ENG.2017.04.011>

[20] Eric Yañez G, “LPWAN: ¿Qué son y para qué se utilizan?,” LinkedIn.

Accessed: Apr. 29, 2025. [Online]. Available:

<https://www.linkedin.com/pulse/lpwan-qu%C3%A9-son-y-para-se-utilizan-eric-ya%C3%B1ez-g/>

[21] Alexander La rosa, “LPWAN como base de comunicaciones para IoT,”

pandorafms. Accessed: Apr. 29, 2025. [Online]. Available:

<https://pandorafms.com/blog/es/que-es-lpwan/>

[22] AlfaIoT, “LPWAN: ¿Qué son las redes de área amplia de baja potencia?”

Accessed: Apr. 29, 2025. [Online]. Available: <https://alfaiot.com/iot/lpwan/>

[23] Alejandra Mabel Pinto Erazo, “DESARROLLO DE UNA METODOLOGÍA

DE ANÁLISIS DE TECNOLOGÍAS DE RADIO IoT, PARA ESCENARIOS

URBANO Y RURAL, POR MEDIO DE MODELOS MATEMÁTICOS,”

UNIVERSIDAD TÉCNICA DEL NORTE, Ibarra, 2020. Accessed: Apr. 29, 2025.

[Online]. Available: <http://repositorio.utn.edu.ec/handle/123456789/10809>

[24] EDWIN FABRICIO AVILA CUEVA and MIGUEL ÁNGEL PARRA

ORDÓÑEZ, “DESARROLLO DE UN PROTOTIPO DE RED LPWAN

CON TECNOLOGÍA LoRa PARA LA DETECCIÓN DE INTRUSOS EN LAS

VIVIENDAS DE UNA ZONA RESIDENCIAL,” ESCUELA POLITÉCNICA

NACIONAL, QUITO, 2020. Accessed: Apr. 29, 2025. [Online]. Available:

<http://bibdigital.epn.edu.ec/handle/15000/21345>

[25] Daniel Rodríguez Díaz, “Red privada LoRaWAN para el ámbito de

aplicaciones en la Industria 4.,” Universidad Politecnica de Catalunya, España, 2021.

Accessed: Apr. 29, 2025. [Online]. Available: <http://hdl.handle.net/2117/359214>

- [26] AlfaIoT, “The Things Network (TTN),” AlfaIoT. Accessed: Apr. 29, 2025. [Online]. Available: <https://alfaiot.com/iot/the-things-network-ttn-una-red-iot-global-para-la-sociedad/>
- [27] ChirpStack, “ChirpStack open-source LoRaWAN,” ChirpStack. Accessed: Apr. 29, 2025. [Online]. Available: <https://alfaiot.com/iot/chirpstack/>
- [28] Loriot, “Loriot,” Loriot. Accessed: Apr. 29, 2025. [Online]. Available: <https://loriot.io/>
- [29] thingpark, “thingpark,” thingpark. Accessed: Apr. 29, 2025. [Online]. Available: <https://www.thingpark.com/>
- [30] Liya Yuni Astutik and Rahadian Yusuf, “Performance Study of LoRa IoT Technology,” Jan. 2023, Accessed: Apr. 29, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10011110>
- [31] G. W. R. Hurtado Arguedas, “Diseño de un sistema de monitoreo remoto basado en tecnología IoT para la optimización de las rutas de recolección de basura de los contenedores de los principales parques de San Isidro basado en el nuevo estándar lorawan ITU-T Y.4480,” UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS, lima, 2023. Accessed: Apr. 29, 2025. [Online]. Available: <http://hdl.handle.net/10757/672552>
- [32] Ignacio Ordoñez Monfort, “Estudio de la arquitectura y el nivel de desarrollo de la red LoraWan y de los dispositivos Lora,” Universitat Oberta de Catalunya, pp. 1–11, Jun. 2018, Accessed: Apr. 29, 2025. [Online]. Available: <http://hdl.handle.net/10609/64365>
- [33] S. D. Páez Chicaiza, “Evaluación del desempeño de una red de sensores para la monitorización de variables físicas del suelo en un invernadero mediante tecnología

Sigfox.,” Universidad de las Fuerzas Armadas ESPE, 2021. Accessed: Apr. 29, 2025. [Online]. Available: <https://redi.cedia.edu.ec/document/232765>

[34] Fiona Kuan, “¿Qué es la tecnología LoRa y cómo funciona? – Una guía detallada,” Moko Smart. Accessed: Apr. 29, 2025. [Online]. Available: <https://www.mokosmart.com/es/lora-technology/>

[35] blogger, “Como funciona la red LPWAN Sigfox,” Descifrando Blockchain. Accessed: Jun. 23, 2024. [Online]. Available: <https://blockchainadministration.blogspot.com/2018/12/como-funciona-la-red-lpwan-sigfox.html>

[36] Edison Fernando Cando Toapanta, “SISTEMA DE COMUNICACIÓN IOT PARA EL RASTREO Y LOCALIZACIÓN DE NIÑOS CON TECNOLOGÍA SIGFOX PARA LA CIUDAD DE AMBATO, ECUADOR.,” UNIVERSIDAD TÉCNICA DE AMBATO, Ambato, 2024. Accessed: Apr. 29, 2025. [Online]. Available: <https://repositorio.uta.edu.ec/handle/123456789/40786>

[37] vencoel, “Qué es LoRa, cómo funciona y características principales,” venco electronica. Accessed: Jun. 23, 2024. [Online]. Available: <https://www.vencoel.com/que-es-lora-como-funciona-y-caracteristicas-principales/>

[38] Gonzalo David Paredes Cruz, “SISTEMA DE MONITOREO DE HUMEDAD DE SUELO, DE LARGO ALCANCE Y GEOREFERENCIACIÓN USANDO INTERNET DE LAS COSAS (IoT),” UNIVERSIDAD TÉCNICA DEL NORTE, Ibarra, 2021. Accessed: Apr. 29, 2025. [Online]. Available: <http://repositorio.utn.edu.ec/handle/123456789/11456>

[39] Bryan Salazar López, “Algoritmo de Dijkstra,” Ingeniería industrial. Accessed: Jun. 23, 2024. [Online]. Available: Bryan Salazar López, “Algoritmo de

Dijkstra,” Ingeniería industrial.

[40] Rudeus Greyrat, “¿Cuáles son las diferencias entre los algoritmos de Bellman Ford y Dijkstra?,” Barcelona Geeks. Accessed: Jun. 21, 2024. [Online]. Available: <https://barcelongeeks.com/cuales-son-las-diferencias-entre-los-algoritmos-de-bellman-ford-y-dijkstra/>

[41] Google maps, “Descripción general de la API de Roads ,” Google maps platform. Accessed: Apr. 29, 2025. [Online]. Available: <https://developers.google.com/maps/documentation/roads/overview?hl=es-419#:~:text=Roads%20API%20es%20un%20servicio,y%20los%201%C3%ADmites%20de%20velocidad.>

[42] HowStuffWorks, “MapQuest: Mapas y direccionesMapQuest: Mapas y direcciones,” Amazon . Accessed: Apr. 29, 2025. [Online]. Available: <https://money.howstuffworks.com/mapquest.htm>

[43] SenseCAP, “SenseCAP M2,” SenseCAP . Accessed: Apr. 27, 2025. [Online]. Available: <https://www.sensecapmx.com/docs/sensecap-m2/overview-m2/>

[44] Heltec Automation, “CubeCell Dev-Board Plus (HTCC-AB02),” Heltec Automation . Accessed: Apr. 27, 2025. [Online]. Available: <https://heltec.org/project/htcc-ab02/>

[45] Novatronic, “Sensor Ultrasónico JSN-SR04T ,” Novatronic.ec. Accessed: May 11, 2025. [Online]. Available: <https://novatronic.ec/index.php/product/sensor-ultrasonico-contra-el-agua-jsn-sr04t/>

[46] Unit Electronic, “Bateria LiPo 3.7V 1500mAh,” Unit Electronic. Accessed: May 11, 2025. [Online]. Available: <https://uelectronics.com/producto/bateria-lipo-3->

7v-1500mah-103050/

[47] Sensecap, “Sensecap M2,” Sensecap . Accessed: May 18, 2025. [Online].

Available: <https://www.sensecapmx.com/docs/sensecap-m2/quick-start-m2/>

ANEXOS

Anexo 1 Código Arduino ID para Medición de Nivel de Llenado y Transmisión vía LoRaWAN

```
#include "LoRaWan_APP.h"
#include "Arduino.h"

// Parámetros de LoRaWAN
uint8_t devEui[] = { 0x22, 0x32, 0x33, 0x00, 0x00, 0x88, 0x88, 0x02 };
uint8_t appEui[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
uint8_t appKey[] = { 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88,
0x88, 0x88, 0x88, 0x88, 0x66, 0x01 };
uint8_t nwkSKey[] = { 0x5A, 0x30, 0x1B, 0x99, 0x80, 0xB5, 0x28, 0xA5, 0x18, 0xB8,
0x71, 0x6D, 0xF2, 0x7F, 0x9F, 0xC0 };
uint8_t appSKey[] = { 0x65, 0xFC, 0x1D, 0x70, 0xC8, 0x59, 0xDF, 0xE6, 0xDE, 0xF7,
0xE1, 0x24, 0x2F, 0x06, 0x2C, 0x1C };
uint32_t devAddr = (uint32_t)0x260c222b;

uint16_t userChannelsMask[6] = { 0x00FF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000
};
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;
DeviceClass_t loraWanClass = LORAWAN_CLASS;

uint32_t appTxDutyCycle = 5000;
bool overTheAirActivation = LORAWAN_NETMODE;
bool loraWanAdr = LORAWAN_ADR;
bool keepNet = LORAWAN_NET_RESERVE;
bool isTxConfirmed = LORAWAN_UPLINKMODE;
uint8_t appPort = 2;
uint8_t confirmedNbTrials = 4;

#define TRIG_PIN GPIO7
#define ECHO_PIN GPIO6
int distance = 0;
int fillLevelPer = 0;
```

```

// Función para obtener promedio de varias lecturas
int getAverageDistance(int samples = 5) {
    long total = 0;
    int validSamples = 0;

    for (int i = 0; i < samples; i++) {
        digitalWrite(TRIG_PIN, LOW);
        delayMicroseconds(5);
        digitalWrite(TRIG_PIN, HIGH);
        delayMicroseconds(10);
        digitalWrite(TRIG_PIN, LOW);

        int duration = pulseIn(ECHO_PIN, HIGH, 30000); // Timeout de 30 ms
        int d = (duration / 2) / 29.1;

        if (d > 0 && d < 400) {
            total += d;
            validSamples++;
        }
        delay(60);
    }

    if (validSamples == 0) return 999;
    return total / validSamples;
}

```

```

// Preparamos el paquete a enviar
static void prepareTxFrame(uint8_t port) {
    distance = getAverageDistance(5);

    if (distance <= 20) {
        fillLevelPer = 90;
    } else if (distance >= 90) {
        fillLevelPer = 0;
    } else {
        fillLevelPer = map(distance, 20, 90, 100, 0);
        fillLevelPer = constrain(fillLevelPer, 0, 100);
    }

    Serial.print("Distancia promedio: ");
    Serial.print(distance);
    Serial.print(" cm | Nivel del contenedor: ");
    Serial.print(fillLevelPer);
    Serial.print(" %");

    if (fillLevelPer >= 90) {
        Serial.println(" Contenedor lleno");
    } else if (fillLevelPer >= 60) {
        Serial.println(" Contenedor medio lleno");
    } else {
        Serial.println(" Contenedor vacío");
    }

    appDataSize = 1;
    appData[0] = (uint8_t)fillLevelPer;
}

void setup() {
    Serial.begin(115200);
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);

    Serial.println("Iniciando sensor ultrasónico JSN-SR04T y LoRaWAN...");
    deviceState = DEVICE_STATE_INIT;
    LoRaWAN.ifskipjoin();
}

```

```

void loop() {
  switch (deviceState) {
    case DEVICE_STATE_INIT:
      printDevParam();
      LoRaWAN.init(loraWanClass, loraWanRegion);
      deviceState = DEVICE_STATE_JOIN;
      break;
    case DEVICE_STATE_JOIN:
      LoRaWAN.join();
      break;
    case DEVICE_STATE_SEND:
      prepareTxFrame(appPort);
      LoRaWAN.send();
      deviceState = DEVICE_STATE_CYCLE;
      break;
    case DEVICE_STATE_CYCLE:
      txDutyCycleTime = appTxDutyCycle + randr(0, APP_TX_DUTYCYCLE_RND);
      LoRaWAN.cycle(txDutyCycleTime);
      deviceState = DEVICE_STATE_SLEEP;
      break;
    case DEVICE_STATE_SLEEP:
      LoRaWAN.sleep();
      break;
    default:
      deviceState = DEVICE_STATE_INIT;
      break;
  }
}

```

Anexo 2 Implementación del flujo en Node-RED mediante archivo .json para la visualización y monitoreo de contenedores inteligentes

```
[
  {
    "id": "2aa3483423a16ac3",
    "type": "tab",
    "label": "Flow 4",
    "disabled": false,
    "info": "",
    "env": []
  },
  {
    "id": "5b821083e9349262",
    "type": "function",
    "z": "2aa3483423a16ac3",
    "name": "Procesar Contenedor 1",
    "func": "// Extraer datos TTN\nconst payload = msg.payload;\nconst level =\npayload.uplink_message.decoded_payload.water_level_percent;\nconst estado =\npayload.uplink_message.decoded_payload.estado_contenedor;\n\nconst lat = -\n2.203461;\nconst lon = -80.872671;\n\n// Determinar color\nlet iconColor =\n\"gray\";\nif (level < 20) {\n  iconColor = \"red\";\n} else if (level < 60) {\n  iconColor = \"yellow\";\n} else if (level > 90) {\n  iconColor = \"green\";\n}\n\n// Mensaje para worldmap\nmsg.payload = {\n  name: \"Contenedor 1\", \n  lat: lat, \n  lon: lon, \n  icon: \"trash\", \n  iconColor: iconColor, \n  layer: \"Contenedores\", \n  message: `Nivel: ${level}% - ${estado}`;\n}\n\n// Mensaje para debug\nmsg.debug = {\n  contenedor: \"Contenedor 1\", \n  nivel: level, \n  color: iconColor;\n};\n\nreturn [msg, msg];",
    "outputs": 2,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 710,
    "y": 140,
    "wires": [
      [
        "a9cb3aa93b8bf22a"
      ],
      [
        "8faed95f2bc9fc56"
      ]
    ]
  }
],
```

```
{
  "id": "09c1126641cf2360",
  "type": "function",
  "z": "2aa3483423a16ac3",
  "name": "Procesar Contenedor 2",
  "func": "// Extraer datos TTN\nconst payload = msg.payload;\nconst level =
payload.uplink_message.decoded_payload.water_level_percent;\nconst estado =
payload.uplink_message.decoded_payload.estado_contenedor;\nconst lat = -2.201910; // Coordenadas diferentes\nconst lon =
-80.871411;\n\n// Determinar color\nlet iconColor = \"gray\";\nif (level < 20) {\n  iconColor = \"red\";\n} else if (level < 60)
{\n  iconColor = \"yellow\";\n} else if (level > 90) {\n  iconColor = \"green\";\n}\n\n// Mensaje para worldmap\nmsg.payload
= {\n  name: \"Contenedor 2\",\n  lat: lat,\n  lon: lon,\n  icon: \"trash\",\n  iconColor: iconColor,\n  layer:
\"Contenedores\",\n  message: `Nivel: ${level}% - ${estado}`\n};\n\n// Mensaje para debug\nmsg.debug = {\n  contenedor:
\"Contenedor 2\",\n  nivel: level,\n  color: iconColor\n};\n\nreturn [msg, msg];",
  "outputs": 2,
  "timeout": "",
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 710,
  "y": 220,
  "wires": [
    [
      "fc5ebb937f882192"
    ],
    [
      "8faed95f2bc9fc56"
    ]
  ]
},
{
  "id": "a9cb3aa93b8bf22a",
  "type": "change",
  "z": "2aa3483423a16ac3",
  "name": "Preparar debug (Contenedor 1)",
  "rules": [
    {
      "t": "set",
      "p": "payload",
      "pt": "msg",
      "to": "debug",
      "tot": "msg"
    }
  ],
  "action": "",
  "property": "",
  "from": "",
  "to": "",
  "reg": false,
  "x": 1030,
  "y": 60,
  "wires": [
    [
      "51ac66612b4a87f0",
      "c9eedee9cb89d5f6"
    ]
  ]
},
}
```

```
{
  "id": "fc5ebb937f882192",
  "type": "change",
  "z": "2aa3483423a16ac3",
  "name": "Preparar debug (Contenedor 2)",
  "rules": [
    {
      "t": "set",
      "p": "payload",
      "pt": "msg",
      "to": "debug",
      "tot": "msg"
    }
  ],
  "action": "",
  "property": "",
  "from": "",
  "to": "",
  "reg": false,
  "x": 1030,
  "y": 160,
  "wires": [
    [
      "51ac66612b4a87f0",
      "77b8a37d027dd4de"
    ]
  ]
},
{
  "id": "8faed95f2bc9fc56",
  "type": "worldmap",
  "z": "2aa3483423a16ac3",
  "name": "mapa",
  "lat": "-2.207051",
  "lon": "-80.872091",
  "zoom": "",
  "layer": "EsriS",
  "cluster": "",
  "maxage": "",
  "usermenu": "show",
  "layers": "show",
  "panit": "true",
  "hiderightclick": "false",
  "coords": "none",
  "path": "/mapa",
  "overlist": "DR,CO,RA,DN,HM",
  "maplist": "OSMG,OSMC,EsriC,EsriS,EsriT,EsriO,EsriDG,NatGeo,UKOS,OpTop",
  "mapname": "",
  "mapurl": "",
  "mapopt": "",
  "mapwms": false,
  "x": 1290,
  "y": 240,
  "wires": []
},
```

```
{
  "id": "51ac66612b4a87f0",
  "type": "debug",
  "z": "2aa3483423a16ac3",
  "name": "debug 4",
  "active": true,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "false",
  "statusVal": "",
  "statusType": "auto",
  "x": 1320,
  "y": 100,
  "wires": []
},
{
  "id": "ded6cb7948277ea4",
  "type": "mqtt in",
  "z": "2aa3483423a16ac3",
  "name": "",
  "topic": "v3/cubecell12@ttn/devices/cubecell/up",
  "qos": "2",
  "datatype": "auto-detect",
  "broker": "d084f29be27f73c0",
  "nl": false,
  "rap": true,
  "rh": 0,
  "inputs": 0,
  "x": 350,
  "y": 140,
  "wires": [
    [
      "5b821083e9349262",
      "09c1126641cf2360"
    ]
  ]
},
},
```

```

{
  "id": "c9eedee9cb89d5f6",
  "type": "function",
  "z": "2aa3483423a16ac3",
  "name": "Verificar Nivel 1",
  "func": "// Obtiene el nivel del contenedor 1 desde el mensaje entrante\nlet
nivelContenedor1 = msg.payload.nivelContenedor1 || msg.payload.nivel || 0;\n\n//
Coordenadas\nconst camion = { lat: -2.207133, lon: -80.872054, name: \"Camión de basura\",
icon: \"truck\" };\nconst contenedor1 = { lat: -2.203461, lon: -80.872671, name: \"Contenedor
1\", icon: \"trash\" };\n\n// Inicializa el payload para el Worldmap\nlet worldmapPayload = [\n
{ name: camion.name, lat: camion.lat, lon: camion.lon, icon: camion.icon },\n { name:
contenedor1.name, lat: contenedor1.lat, lon: contenedor1.lon, icon: contenedor1.icon }\n];\n\n//
Variable para almacenar el mensaje de Telegram\nlet telegramMsg = null;\n\n// Maneja la línea
dependiendo del nivel\nif (nivelContenedor1 >= 90) {\n // Agrega la línea roja\n
worldmapPayload.push({\n name: \"Ruta Camión a Contenedor 1\", \n line: [\n
[camion.lat, camion.lon],\n [contenedor1.lat, contenedor1.lon]\n ], \n color:
\"red\", \n weight: 3, \n opacity: 1\n });\n\n // Crea el link de Google Maps\n const
mapLink = `https://www.google.com/maps?q=${contenedor1.lat},${contenedor1.lon}`;\n\n //
Mensaje de Telegram con link incluido\n telegramMsg = {\n payload: {\n chatId:
8009183194, \n type: \"message\", \n content: ` 🚨 ALERTA: El contenedor 1 ha
alcanzado el ${nivelContenedor1}% de su capacidad. Es necesario vaciarlo
inmediatamente.\n 📍 Ver ubicación: ${mapLink}`\n };\n } else {\n // Envía una
línea vacía para eliminar la existente\n worldmapPayload.push({\n name: \"Ruta Camión
a Contenedor 1\", \n line: []\n });\n}\n\n// Configura el mensaje para el Worldmap\nlet
worldmapMsg = {\n payload: worldmapPayload, \n worldmap: {\n command: \"set\", \n
layer: \"RutaContenedor1\"\n };\n}\n\n// Devuelve los mensajes (uno al Worldmap, otro a
Telegram si aplica)\nreturn telegramMsg ? [worldmapMsg, telegramMsg] : [worldmapMsg,
null];\n",
  "outputs": 2,
  "timeout": "",
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 820,
  "y": 400,
  "wires": [
    [
      "8faed95f2bc9fc56"
    ],
    [
      "0d4afa94ef45edf9"
    ]
  ]
},

```

```
{
  "id": "0d4afa94ef45edf9",
  "type": "telegram sender",
  "z": "2aa3483423a16ac3",
  "name": "Enviar a Telegram",
  "bot": "09c1e053584860e2",
  "haserroroutput": false,
  "outputs": 1,
  "x": 1110,
  "y": 400,
  "wires": [
    [
      "dc70b0d49872a54b"
    ]
  ]
},
{
  "id": "dc70b0d49872a54b",
  "type": "debug",
  "z": "2aa3483423a16ac3",
  "name": "debug 1",
  "active": true,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "false",
  "statusVal": "",
  "statusType": "auto",
  "x": 1360,
  "y": 400,
  "wires": []
},
```

```

{
  "id": "77b8a37d027dd4de",
  "type": "function",
  "z": "2aa3483423a16ac3",
  "name": "Verificar Nivel 2",
  "func": "// Obtiene el nivel del contenedor 2 desde el mensaje entrante\nlet
nivelContenedor2 = msg.payload.nivelContenedor2 || msg.payload.nivel || 0;\n\n//
Coordenadas\nconst camion = { lat: -2.207133, lon: -80.872054, name: \"Camión de
basura\", icon: \"truck\" };const puntoIntermedio = { lat: -2.203461, lon: -80.872671 };//
Antiguo contenedor 1\nconst contenedor2 = { lat: -2.201910, lon: -80.871411, name:
\"Contenedor 2\", icon: \"trash\" };;\n\n// Inicializa el payload para el Worldmap\nlet
worldmapPayload = [\n  { name: camion.name, lat: camion.lat, lon: camion.lon, icon:
camion.icon },\n  { name: contenedor2.name, lat: contenedor2.lat, lon: contenedor2.lon,
icon: contenedor2.icon }];;\n\n// Variable para almacenar el mensaje de Telegram\nlet
telegramMsg = null;\n\n// Si el contenedor 2 está lleno (>= 90%), agrega la ruta pasando
por el punto intermedio\nif (nivelContenedor2 >= 90) {\n  worldmapPayload.push({\n
name: \"Ruta Camión a Contenedor 2 (vía Contenedor 1)\",\n  line: [\n
[camion.lat, camion.lon],\n    [puntoIntermedio.lat, puntoIntermedio.lon],\n
[contenedor2.lat, contenedor2.lon]\n  ],\n  color: \"red\", \n  weight: 3,\n
opacity: 1\n  });;\n\n  // Link de ubicación en Google Maps\n  const mapLink =
`https://www.google.com/maps?q=${contenedor2.lat},${contenedor2.lon}`;\n\n  //
Mensaje de alerta con link\n  telegramMsg = {\n    payload: {\n      chatId:
8009183194,\n      type: \"message\", \n      content: `🚨 ALERTA: El contenedor 2
ha alcanzado el ${nivelContenedor2}% de su capacidad. Es necesario vaciarlo
inmediatamente.\n 📍 Ver ubicación: ${mapLink}`\n    };;\n  } else {\n    // Elimina
la línea roja si el contenedor ya no está lleno\n    worldmapPayload.push({\n      name:
\"Ruta Camión a Contenedor 2 (vía Contenedor 1)\",\n      line: []\n    });;\n\n\n//
Configura el mensaje para el Worldmap\nlet worldmapMsg = {\n  payload:
worldmapPayload,\n  worldmap: {\n    command: \"set\", \n    layer:
\"RutaContenedor2\"\n  };\n};;\n\n// Devuelve los mensajes (uno al Worldmap, otro a
Telegram si aplica)\nreturn telegramMsg ? [worldmapMsg, telegramMsg] : [worldmapMsg,
null];;\n",
  "outputs": 2,
  "timeout": "",
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 820,
  "y": 480,
  "wires": [
    [
      "8faed95f2bc9fc56"
    ],
    [
      "e1ada7aeee38a9eb"
    ]
  ]
},

```

```
{
  "id": "e1ada7aeee38a9eb",
  "type": "telegram sender",
  "z": "2aa3483423a16ac3",
  "name": "Enviar a Telegram",
  "bot": "09c1e053584860e2",
  "haserroroutput": false,
  "outputs": 1,
  "x": 1090,
  "y": 480,
  "wires": [
    [
      "ae66115c99fa5c43"
    ]
  ]
},
{
  "id": "ae66115c99fa5c43",
  "type": "debug",
  "z": "2aa3483423a16ac3",
  "name": "debug 2",
  "active": true,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "false",
  "statusVal": "",
  "statusType": "auto",
  "x": 1360,
  "y": 480,
  "wires": []
},
```

```
{
  "id": "d084f29be27f73c0",
  "type": "mqtt-broker",
  "name": "cubecell1",
  "broker": "nam1.cloud.thethings.network",
  "port": 1883,
  "clientid": "",
  "autoConnect": true,
  "usetls": false,
  "protocolVersion": 4,
  "keepalive": 60,
  "cleansession": true,
  "autoUnsubscribe": true,
  "birthTopic": "",
  "birthQos": "0",
  "birthRetain": "false",
  "birthPayload": "",
  "birthMsg": {},
  "closeTopic": "",
  "closeQos": "0",
  "closeRetain": "false",
  "closePayload": "",
  "closeMsg": {},
  "willTopic": "",
  "willQos": "0",
  "willRetain": "false",
  "willPayload": "",
  "willMsg": {},
  "userProps": "",
  "sessionExpiry": ""
},
{
  "id": "09c1e053584860e2",
  "type": "telegram bot",
  "botname": "Tesisupse_bot",
  "usernames": "",
  "chatids": "",
  "baseapiurl": "",
  "testenvironment": false,
  "updatemode": "polling",
  "pollinterval": 300,
  "usesocks": false,
  "sockshost": "",
  "socksprotocol": "socks5",
  "socksport": 6667,
  "socksusername": "anonymous",
  "sockspassword": "",
  "bothost": "",
  "botpath": "",
  "localbothost": "0.0.0.0",
  "localbotport": 8443,
  "publicbotport": 8443,
  "privatekey": "",
  "certificate": "",
  "useselfsignedcertificate": false,
  "sslterminated": false,
  "verboselogging": false
}
]
```