



Universidad  
Nacional  
de Loja

# Universidad Nacional De Loja

**Facultad de la Energía, las Industrias y los Recursos Naturales no  
Renovables**

**Carrera De Ingeniería en Electrónica y Telecomunicaciones**

**DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE ESTACIÓN HIDROLÓGICA  
AUTÓNOMA PARA EL CENTRO DE INVESTIGACIÓN E INNOVACIÓN  
TECNOLÓGICA I2TEC DE LA UNIVERSIDAD NACIONAL DE LOJA**

**Trabajo de Titulación, previo a la  
obtención del Título de Ingeniero en  
Electrónica y Telecomunicaciones.**

**AUTOR:**

Edison Stalin Salinas Nalvay

**DIRECTOR:**

Ing. Renato Benjamín Torres Carrión, Mg. Sc

Loja – Ecuador

2024

## Certificación

Loja, 29 de febrero de 2024

Ing. Renato Benjamín Torres Carrión. Mg. Sc.

**DIRECTOR DEL TRABAJO DE TITULACIÓN**

### **CERTIFICO:**

Que he revisado y orientado todo proceso de la elaboración del Trabajo de Titulación denominado: **DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE ESTACIÓN HIDROLÓGICA AUTÓNOMA PARA EL CENTRO DE INVESTIGACIÓN E INNOVACIÓN TECNOLÓGICA I2TEC DE LA UNIVERSIDAD NACIONAL DE LOJA**, previo a la obtención del título de **Ingeniero en Electrónica y Telecomunicaciones** de la autoría de la estudiante: **Edison Stalin Salinas Nalvay, con cédula de identidad Nro. 1150816989**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Ing. Renato Benjamín Torres Carrión, Mg. Sc.

**DIRECTOR DEL TRABAJO DE TITULACIÓN**

## **Autoría**

Yo, **Edison Stalin Salinas Nalvay**, declaro ser autora del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

**Firma:**

**Cedula de Identidad:** 1150816989

**Fecha:** 29/02/2024

**Correo electrónico:** edison.s.salinas@unl.edu.ec

**Teléfono:** 0980114737

**Carta de autorización del Trabajo de Titulación por parte del autor, para la consulta, reproducción parcial o total, y publicación electrónica del texto completo, del Trabajo de Titulación.**

Yo, **Edison Stalin Salinas Nalvay**, declaro ser autor del Trabajo de Titulación denominado: **DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE ESTACIÓN HIDROLÓGICA AUTÓNOMA PARA EL CENTRO DE INVESTIGACIÓN E INNOVACIÓN TECNOLÓGICA I2TEC DE LA UNIVERSIDAD NACIONAL DE LOJA**, como requisito para optar al título de **Ingeniero en Electrónica y Telecomunicaciones**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veintinueve días del mes de febrero del dos mil veinticuatro.

**Firma:**

**Autor:** Edison Stalin Salinas Nalvay

**Cédula:** 1150816989

**Dirección:** Catamayo- (Barrio las Canoas)

**Correo electrónico:** edison.s.salinas@unl.edu.ec

**Teléfono:** 0980114737

**DATOS COMPLEMENTARIOS:**

**Director del Trabajo de Titulación:** Ing. Renato Benjamín Torres Carrión, Mg. Sc.

## **Dedicatoria**

Esta investigación, se dedicó a mis padres Ángel Salinas y Sonia Nalvay, todo esto ha sido posible gracias a su inquebrantable apoyo, amor y sacrificio. Su constante aliento y guía han sido fundamentales en este camino.

A mis queridos hermanos Estefanía, Mariela, Erika y Dilan quienes han sido un apoyo invaluable a lo largo de mi vida y especialmente en esta etapa de mi formación académica. Su amor, comprensión y apoyo inquebrantable me han impulsado a seguir adelante en los momentos más desafiantes.

A mis amigos y compañeros de la universidad, con quienes he compartido cada etapa de este camino académico. Agradezco profundamente su apoyo, amistad y compañía a lo largo de esta travesía. Sus palabras de aliento, las horas de estudio compartidas y las risas que hemos compartido.

A todos ustedes, les dedico este logro, que es el resultado de su amor, apoyo y comprensión. Gracias por ser mi fuente de inspiración y mi roca en momentos de dificultad. Comparto este logro con ustedes con gratitud y amor.

***Edison Stalin Salinas Nalvay***

## **Agradecimiento**

Agradezco a Dios por guiarme en este arduo camino de aprendizaje y superación. A mi familia, por su apoyo incondicional a lo largo de este viaje. Sus sacrificios y amor han sido mi pilar y mi inspiración.

Además, quiero expresar mi profundo agradecimiento a todos los docentes de la Universidad Nacional de Loja, quienes me han brindado la dirección y las herramientas necesarias para mi formación como profesional. En particular, al Ingeniero Renato Torres, director de mi Trabajo de Titulación, su paciencia, orientación y sabiduría como docente han sido esenciales para la culminación de esta investigación.

También deseo agradecer al ingeniero Luis Duque, quien ha compartido sus conocimientos y brindado su apoyo en el desarrollo de este proyecto. Su colaboración ha sido de gran valor.

A todos los que han contribuido a mi formación y éxito, les agradezco de corazón. Este logro es el resultado de sus enseñanzas, apoyo y orientación. Lo comparto con todos ustedes con profunda gratitud y reconocimiento.

***Edison Stalin Salinas Nalvay***

## Índice de contenidos

<b>Portada</b> .....	<b>i</b>
<b>Certificación</b> .....	<b>ii</b>
<b>Autoría</b> .....	<b>iii</b>
<b>Carta de autorización</b> .....	<b>iv</b>
<b>Dedicatoria</b> .....	<b>v</b>
<b>Agradecimiento</b> .....	<b>vi</b>
<b>Índice de contenidos</b> .....	<b>vii</b>
<b>Índice de tablas</b> .....	<b>x</b>
<b>Índice de figuras</b> .....	<b>x</b>
<b>Índice de anexos</b> .....	<b>xiii</b>
<b>1. Título</b> .....	<b>1</b>
<b>2. Resumen</b> .....	<b>2</b>
Abstract.....	3
<b>3. Introducción</b> .....	<b>4</b>
<b>4. Marco teórico</b> .....	<b>6</b>
4.1. Estaciones Hidrológicas Autónoma .....	6
4.2. Variables Hidrométricas a censar.....	8
4.2.1. Nivel de agua .....	8
4.2.2. Precipitación .....	8
4.2.3. Humedad .....	10
4.2.4. Temperatura .....	10
4.2.5. Presión atmosférica.....	10

4.3. IoT.....	11
4.4. Redes LPWAN.....	11
4.4.1. Características.....	11
4.4.2. Diferencias de LPWAN con otras Redes.....	12
4.4.3. Comparativa de las tecnologías de LPWAN.....	12
4.5. Tecnología de acceso a radio .....	14
4.5.1. Características de LoRa.....	14
4.5.2. Elementos de una red LoRaWAN.....	15
4.5.3. Servidores en LoraWAN.....	16
4.5.4. Modulación LoRa.....	17
4.5.5. Factor de dispersión.....	19
4.5.6. Características de la modulación LoRa según su región.....	20
4.6. The Things Network.....	22
4.7. Plataformas de IoT.....	22
4.7.1. Comparativa de las plataformas IoT.....	23
4.7.2. Selección de la Plataforma IoT.....	24
4.8. Microsoft Azure .....	25
4.8.1. Centro de IoT de Microsoft Azure .....	25
4.8.2. Funciones de Microsoft Azure .....	26
4.8.3. Base de Datos Cosmos de Microsoft Azure.....	27
4.9. Métodos para la medición del nivel de Agua.....	28
<b>5. Metodología.....</b>	<b>31</b>
5.1. Estudio de Campo .....	32
5.2. Diseño y Arquitectura de la Red LoRaWAN.....	35
5.3. Configuración de Gateway.....	36
5.4. Registro y Configuración del Gateway en The Things Networks .....	40
5.5. Configuración de aplicación y dispositivo final en TTN .....	43
5.6. Integración de TTN con Azure IoT Hub.....	48
5.7. Instalación/Implementación de servicios de App functions y cosmos DB.....	53

5.7.1. Azure Functions .....	53
5.6.2. Cosmos DB .....	54
5.8. Integración de base de datos de Cosmos DB, App Functions e IoT Hub .....	56
5.9. Simulación en OMNeT ++6.....	62
5.9.1. Definición de parámetros de la red .....	64
5.9.2. Resultados de simulación .....	64
5.10. Materiales .....	68
5.10.1. Equipo para la medición del nivel de agua.....	68
5.10.2. Equipos para la medición de precipitaciones .....	69
5.10.3. Equipo para la medición de humedad, temperatura y presión barométrica .....	70
5.10.4. Nodo Heltec WIFI LoRA 32 .....	71
5.10.5. Gateway wAP LR9 .....	73
<b>6. Resultados .....</b>	<b>74</b>
6.1. Simulación de radio enlace y cobertura .....	74
6.4. Integración de servicios de Azure .....	78
<b>7. Discusión .....</b>	<b>85</b>
<b>8. Conclusiones .....</b>	<b>86</b>
<b>9. Recomendaciones .....</b>	<b>88</b>
<b>10. Bibliografía .....</b>	<b>89</b>
<b>11. Anexos .....</b>	<b>92</b>

## Índice de Tablas:

<b>Tabla 1.</b> Comparativa de las tecnologías LPWAN más destacadas .....	13
<b>Tabla 2.</b> Comparativa de las tecnologías de LPWAN dedicados a IoT .....	14
<b>Tabla 3.</b> Factor de dispersión de Lora .....	20
<b>Tabla 4.</b> Características de Modulación Lora.....	21
<b>Tabla 5.</b> Parámetros de configuración en los dispositivos LoRa .....	22
<b>Tabla 6.</b> Comparativa de las plataformas IoT .....	23
<b>Tabla 7.</b> Comparativa de los precios y niveles gratuitos .....	24
<b>Tabla 8.</b> Parámetros Generales para la simulación FloRA.....	63
<b>Tabla 9.</b> Resultados tasa de extracción de datos (DER).....	65
<b>Tabla 10.</b> Consumo de energía promedio y colisiones.....	66
<b>Tabla 11.</b> Características del sensor JSN-SR04T-V30.....	69
<b>Tabla 12.</b> Características técnicas del pluviómetro Misol.....	70
<b>Tabla 13.</b> Características del sensor BME280.....	71
<b>Tabla 14.</b> Características del módulo Heltec Wifi Lora 32 .....	72
<b>Tabla 15.</b> Características wAP LR9 .....	73
<b>Tabla 16.</b> Materiales usados en el montaje de la estación EHA.....	93
<b>Tabla 17.</b> Materiales usados en el montaje de la estación EMA.....	94

## Índice de Figuras:

<b>Figura 1.</b> Estaciones hidrológicas ubicadas en la cuenca del río Catamayo .....	6
<b>Figura 2.</b> Estaciones hidrológicas ubicadas en la cuenca del río Puyango .....	7
<b>Figura 3.</b> Red de estaciones automáticas Hidrometereológicas en la Provincia de Loja Ecuador.....	7
<b>Figura 4.</b> Niveles de precipitaciones .....	9
<b>Figura 5.</b> Comparativa de tecnologías Inalámbricas con LPWAN .....	12
<b>Figura 6.</b> Tecnología de LoRa y LoRaWAN .....	15
<b>Figura 7.</b> Clases de dispositivos finales de LoraWAN .....	16
<b>Figura 8.</b> Elementos de la red LoRaWAN .....	17
<b>Figura 9.</b> Modulación de espectro ensanchado chirp .....	19
<b>Figura 10.</b> Frecuencia de Canal.....	20
<b>Figura 11.</b> Vertedero rectangular rectangular sin contracciones.....	29

<b>Figura 12.</b> Vertedero rectangular con dos contracciones .....	29
<b>Figura 13.</b> Vertedero triangular .....	30
<b>Figura 14.</b> Vertedero Trapezoidal .....	30
<b>Figura 15.</b> Fases para el diseño de cada sistema .....	32
<b>Figura 16.</b> Parque PUEAR .....	33
<b>Figura 17.</b> Ubicación del gateway y las estaciones .....	34
<b>Figura 18.</b> Vertedero triangular .....	34
<b>Figura 19.</b> Diagrama de Red .....	36
<b>Figura 20.</b> Acceso administrativo al Gateway .....	37
<b>Figura 21.</b> Servidor LoRa.....	38
<b>Figura 22.</b> Configuración de Servidor.....	39
<b>Figura 23.</b> Habilitación de servidor.....	40
<b>Figura 24.</b> Suscripción a TTN.....	41
<b>Figura 25.</b> Ingreso a la plataforma del Gateway .....	41
<b>Figura 26.</b> Creación de un Nuevo GW .....	42
<b>Figura 27.</b> Registro del Gateway.....	42
<b>Figura 28.</b> Datos de conexión en vivo.....	43
<b>Figura 29.</b> Ventana de Aplicación.....	44
<b>Figura 30.</b> Formulario para la creación de la aplicación .....	44
<b>Figura 31.</b> Interfaz de Aplicación.....	45
<b>Figura 32.</b> Registro de dispositivo final .....	46
<b>Figura 33.</b> Configuración y creación del dispositivo final .....	47
<b>Figura 34.</b> Dispositivo final creado correctamente .....	47
<b>Figura 35.</b> Inicio Azure IoT Hub.....	48
<b>Figura 36.</b> Credenciales de integración .....	49
<b>Figura 37.</b> Plantilla personalizada .....	50
<b>Figura 38.</b> Salida de Azure IoT Hub .....	51
<b>Figura 39.</b> Habilitar la integración de Azure IoT Hub .....	51
<b>Figura 40.</b> Dispositivo vinculado .....	52
<b>Figura 41.</b> Servicio de App Functions.....	53
<b>Figura 42.</b> Creación de cuenta App functions .....	54
<b>Figura 43.</b> Servicio de Cosmos DB.....	55
<b>Figura 44.</b> Creación de cuenta Azure Cosmos DB .....	56
<b>Figura 45.</b> Creación de contenedores Cosmos DB.....	57

<b>Figura 46.</b> Parámetros de contenedor .....	58
<b>Figura 47.</b> Creación de una nueva función.....	59
<b>Figura 48.</b> Conexión de la función con IoT Hub .....	60
<b>Figura 49.</b> Salida a la base de datos .....	61
<b>Figura 50.</b> Código y prueba de consultas .....	61
<b>Figura 51.</b> Topología de red LoRa .....	63
<b>Figura 52.</b> Simulación con 100 nodos.....	64
<b>Figura 53.</b> Comportamiento del DER .....	66
<b>Figura 54.</b> Consumo promedio de energía .....	67
<b>Figura 55.</b> Colisiones en relación con el número de nodos.....	68
<b>Figura 56.</b> Sensor ultrasónico JSN-SR04T-V3.0 .....	68
<b>Figura 57.</b> Pluviómetro de balancín Misol WH-SP-RG.....	70
<b>Figura 58.</b> Sensor BME/BMP280 .....	71
<b>Figura 59.</b> Modulo Wifi Lora 32.....	72
<b>Figura 60.</b> Gateway wAP LR9.....	73
<b>Figura 61.</b> Radioenlace entre el nodo 1 EHA con el Gateway.....	75
<b>Figura 62.</b> Radioenlace entre el Nodo 2 EMA con el Gateway .....	75
<b>Figura 63.</b> Cobertura del Gateway .....	76
<b>Figura 64.</b> Gateway conectado a TTN .....	77
<b>Figura 65.</b> Nodo final conectado a TTN .....	78
<b>Figura 66.</b> Lectura y almacenamiento de variables.....	79
<b>Figura 67.</b> Datos almacenados en Cosmos DB .....	80
<b>Figura 68.</b> Conexión de Power BI con Cosmos DB.....	81
<b>Figura 69.</b> Datos de Conexión.....	82
<b>Figura 70.</b> Tabla de datos mostrada desde Power Bi .....	83
<b>Figura 71.</b> Visualización diaria de datos en Power BI.....	83
<b>Figura 72.</b> Visualización general de datos Power BI .....	84
<b>Figura 73.</b> Descripción de materiales usados en el vertedero .....	96
<b>Figura 74.</b> Diagrama electrónico.....	97
<b>Figura 75.</b> Descripción de materiales usados en el nodo EHA .....	97
<b>Figura 76.</b> Diagrama electrónico.....	99
<b>Figura 77.</b> Descripciones de materiales usados en el nodo EMA .....	99

## **Índice de Anexos:**

<b>Anexo 1.</b> Manual técnico de montaje y mantenimiento de las estaciones .....	92
<b>Anexo 2.</b> Código implementado para la conexión de la estación EHA con el equipo Heltec .....	101
<b>Anexo 3.</b> Código implementado para la conexión de la estación EMA con el equipo Heltec. .....	105
<b>Anexo 4.</b> Código “Payload” usado en TTN para poder interpretar los datos .....	109
<b>Anexo 5.</b> Código implementado Azure Funciones para filtrar y almacenar los datos.....	110
<b>Anexo 6.</b> Configuración inicial de simulación de la red LoraWAN en OMNeT++ .....	112
<b>Anexo 7.</b> Registro de la instalación de la red LoraWAN.....	114
<b>Anexo 8.</b> Certificado de traducción del resumen al idioma inglés .....	117

## **1. Título**

**DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE ESTACIÓN  
HIDROLÓGICA AUTÓNOMA PARA EL CENTRO DE INVESTIGACIÓN E  
INNOVACIÓN TECNOLÓGICA I2TEC DE LA UNIVERSIDAD NACIONAL DE  
LOJA**

## 2. Resumen

La necesidad de monitorear y recopilar datos hidrológicos de manera precisa y eficiente ha impulsado el desarrollo de tecnologías innovadoras. Tradicionalmente, la recopilación de datos hidrológicos implicaba procedimientos manuales, lo que resultaba en una serie de limitaciones como datos incompletos o imprecisos. En el presente trabajo de investigación se planteó el diseño y construcción de un prototipo de estación hidrométrica autónoma (EHA) ubicado en el parque universitario de educación ambiental y recreación (PUEAR) de la Universidad Nacional de Loja. La estación integra tecnologías LoRaWAN y servicios en la nube de Microsoft Azure, que permite la recolección de datos hidrometeorológicos de manera autónoma y eficiente, que serán utilizados para el análisis e investigaciones futuras. Las pruebas de simulación y de campo validaron la capacidad de la estación para recopilar datos con precisión para transmitirlos y almacenarlos de manera confiable.

**Palabras clave:** LoRaWAN, Microsoft Azure, OMNeT++, Estación hidrométrica autónoma, Estación meteorológica autónoma.

## **Abstract**

The need to monitor and collect hydrological data in an accurate and efficient way has driven the development of innovative technologies. Traditionally, hydrological data compilation involved manual procedures, resulting in a number of limitations such as incomplete or imprecise data. In the present research work, the design and construction of a prototype of an autonomous hydrological station (EHA) was raised, located at the University Park of Environmental Education and Recreation (PUEAR) of the National University of Loja. The station integrates LoRaWAN technologies and Microsoft Azure cloud services, which allows the collection of hydrometeorological data in an autonomous and efficient manner, which will be used for analysis and future research. Simulation and field tests validated the station's ability to accurately collect data for transmission and storage in a reliable way.

**Keywords:** *LoRaWAN, Microsoft Azure, OMNeT++, Autonomous hydrological station, Autonomous meteorological station.*

### 3. Introducción

A lo largo de la historia, el ser humano ha buscado formas de entender el comportamiento del clima para poder aprovechar los recursos naturales de su entorno. Sin embargo, hoy en día, el cambio climático se está acelerando a pasos agigantados debido a la influencia de la actividad humana, lo que provoca cambios repentinos en el comportamiento del clima. Estos cambios se manifiestan en forma de lluvias torrenciales o temperaturas extremas, lo que pone de relieve la importancia de comprender mejor este fenómeno y tomar medidas para mitigar sus efectos.

En este sentido, la recolección y análisis de datos sobre los parámetros climáticos como precipitaciones, nivel de agua, humedad y temperatura, son fundamentales para entender el impacto del clima en nuestro medio ambiente. En este sentido, la Universidad Nacional de Loja está dando un paso importante para el estudio de los fenómenos hidrológicos y la investigación al construir una Estación Hidrológica Autónoma EHA. Esta tecnología será un punto de partida para obtener datos sobre el clima, será de ayuda para el estudio del comportamiento del clima.

La OMM plantea como objetivos fomentar la creación y mantenimiento de sistemas para el intercambio rápido de información meteorológica y conexa, además de fomentar la normalización de las observaciones meteorológicas y conexas para asegurar la publicación uniforme de observaciones y estadística. En consecuencia, para el desarrollo del trabajo de titulación, se planteó un enfoque de investigación cuantitativa ya que es una forma estructurada de recopilar datos que son medibles.

En la EHA se construirá un vertedero que facilite la medición del nivel del agua junto con los demás sensores. Además, se usará una red LPWAN (*Low Power Wide Area Networks*) ya que son tecnologías de comunicación inalámbrica de baja potencia y consumo energético, además permiten transmitir datos a largas distancias, por sus características se emplea en aplicaciones IoT (*Internet of Things*, o “internet de las cosas”) ya que no requieren de un alto consumo de ancho de banda. LoRaWAN es un protocolo de comunicación basado en LoRa la cual utiliza bandas ISM con modulación de espectro ensanchado chirp cuya señal resultante tiene bajos niveles de ruido, lo que permite una alta resistencia a la interferencia, y es difícil de detectar.

La investigación se realizará con la finalidad de poder construir una EHA que permita recolectar datos del Parque Universitario de Educación Ambiental y Recreación (PUEAR) de la Universidad Nacional de Loja. Por tal razón se ha establecido como objetivo general lo siguiente:

Diseñar y construir una estación hidrológica autónoma usando sensores para la captación de variables hidrológicas en afluentes de agua, a través una red de sensores inalámbricos (WSN) para el parque PUEAR-UNL.

Y como objetivos específicos:

Realizar una revisión bibliográfica sobre las tecnologías existentes para determinar la más apropiada en la obtención y transmisión de datos hidrológicos como, nivel de agua, temperatura, humedad, presión atmosférica y precipitaciones.

Construir un prototipo de estación hidrológica con autonomía energética y transmisión de datos inalámbrica, que incluya el uso de una plataforma web de IoT con lo que se espera implementar y poner a disposición de docentes y estudiantes que requieran hacer análisis y estudios sobre el comportamiento de dichas variables.

Elaboración de un manual técnico de montaje y mantenimiento de la estación, el cual servirá como constancia de los materiales y elementos utilizados durante el montaje, y como guía en el desarrollo de proyectos similares.

## 4. Marco teórico

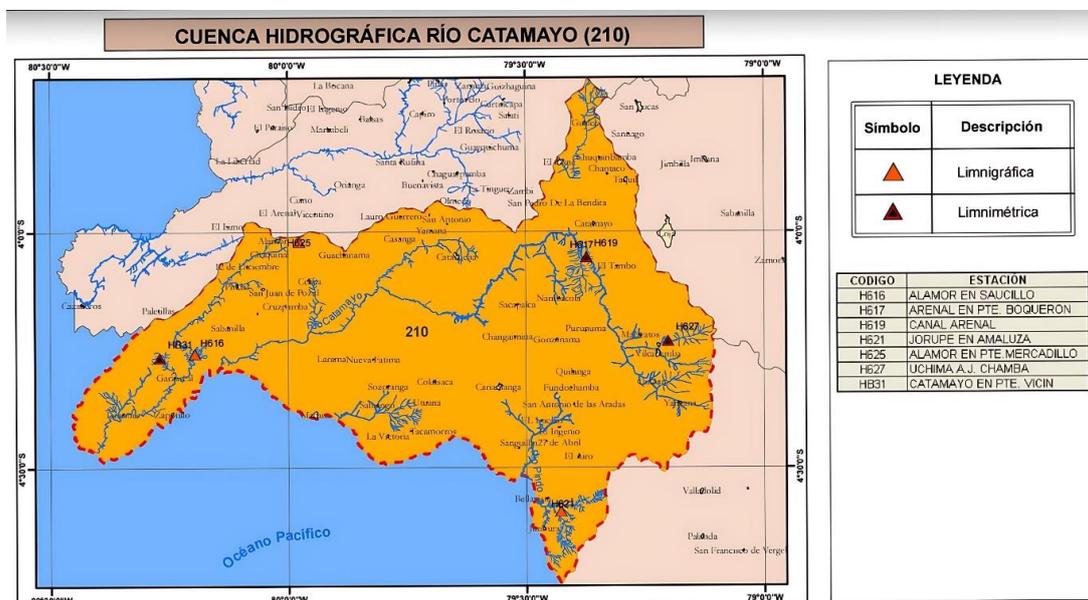
### 4.1. Estaciones Hidrológicas Autónoma

Una EHA es un conjunto de dispositivos e instrumentos que recogen datos de variables como nivel de agua en un río, lagos, afluentes naturales e incluso en estructuras creadas por el hombre como pueden ser canales, represas, reservorios entre otros, además, incorporan pluviómetros para registrar eventos de lluvia. La ubicación de las estaciones hidrológicas se selecciona cuidadosamente para proporcionar información valiosa según lo recomendado por la OMM. (Instituto Privado de Investigación sobre Cambio Climático [ICC], 2020).

En el Ecuador, la entidad encargada de la generación y difusión de la información hidrometeorológica es el Instituto Nacional de Meteorología e Hidrología (INAMHI) que cuenta con una red de estaciones meteorológicas e hidrológicas. En fig. 1 se muestra que en la provincia de Loja el INAMHI registra 7 estaciones Hidrológicas ubicadas en la cuenca hidrográfica del río Catamayo identificadas con los códigos H616, H617, H619, H621, H625, H627 y H831; mientras que en la fig. 2 se logran identificar 5 estaciones ubicadas en la cuenca hidrográfica del río Puyango con los siguientes identificativos H586, H587, H588, H589 y H591 (Instituto Nacional de Meteorología e Hidrología [INAMHI], 2011).

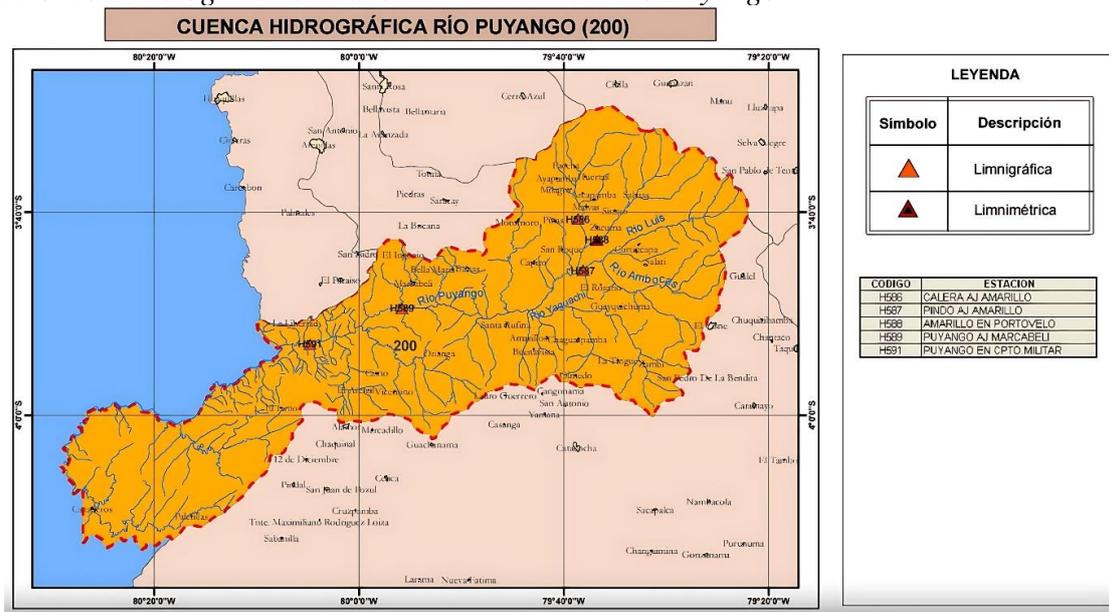
**Figura 1.**

*Estaciones hidrológicas ubicadas en la cuenca del río Catamayo*



*Nota:* La figura muestra la ubicación de cada estación hidrológica a lo largo de la cuenca del río Catamayo. Fuente: (Instituto Nacional de Meteorología e Hidrología [INAMHI], 2011).

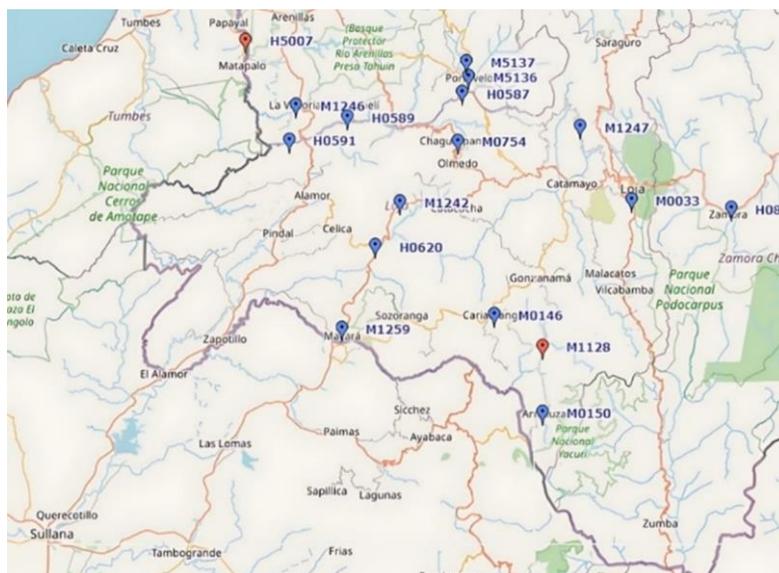
**Figura 2.**  
Estaciones hidrológicas ubicadas en la cuenca del río Puyango



*Nota.* La figura muestra la ubicación de cada estación hidrológica a lo largo de la cuenca del río Puyango. Fuente: (Instituto Nacional de Meteorología e Hidrología [INAMHI], 2011).

Actualmente el INAMHI cuenta con 3 estaciones funcionales, ubicadas en la Cuenca del Río Puyango, mientras que las estaciones hidrológicas de la Cuenca Catamayo únicamente se encuentra una estación en funcionamiento, en la fig. 3 se aprecia una red de estaciones hidrológicas las cuales están identificadas con una “H” seguidas con su código de identificación y con una “M” las estaciones meteorológicas seguidas de su código de identificación.

**Figura 3.**  
Red de estaciones automáticas Hidrometeorológicas en la Provincia de Loja Ecuador



*Nota.* Estaciones hidrometeorológicas funcionales en la provincia de Loja. Fuente: (Instituto Nacional de Meteorología e Hidrología [INAMHI], 2021).

## **4.2. Variables Hidrométricas a censar**

Existen variables hidrológicas físicas, tales como la precipitación, el nivel del agua, caudal, siendo los más relevantes. (Antonino, 2018) menciona que una estación hidrológica se usa para monitorear parámetros hidrológicos en ríos, lagos, canales, vertientes naturales y entre otras afluentes de agua Básicamente, cuando se trata de una estación Hidrológica, hay tres variables interesantes para estudiar: Precipitación, nivel del agua y caudal. La información hídrica se obtiene de las estaciones hidrológicas (EH), complementada con mediciones periódicas de caudal, nivel de agua obtenidos ya sea en un río o cualquier afluente de agua natural.

### **4.2.1. Nivel de agua**

El nivel del agua es la altura de la superficie de un río, lago u otro cuerpo de agua sobre una superficie en particular. La referencia, en el caso de un río, sería desde su fondo. En general, las medidas deben ser precisas al centímetro en estaciones de registro continuo, la precisión debe ser mucho mejor. Las mediciones del nivel, suelen ser usadas para pronóstico de inundaciones (altura máxima del agua), identificar o delinear áreas de riesgo de inundaciones y diseñar estructuras (puentes u otras obras hidro técnicas) además de que permite controlar el volumen y reserva de agua en estructuras construidas por el hombre, el nivel de agua se mide comúnmente utilizando unidades en metros (m), centímetros (cm) y pies (ft) (Chamorro, 2008).

Otro método para medir el nivel del agua es el método de estaciones de aforo de caudales, o también conocidos como vertederos. Estas estaciones tienen como objetivo obtener registros sistemáticos de la altura y el caudal fluvial. Los registros de caudal se calculan a partir de la relación entre la altura del agua y el caudal, la cual se establece mediante mediciones periódicas del caudal y un registro sistemático de la altura del agua. También es posible utilizar una estructura de medición previamente calibrada en un laboratorio o en condiciones reales (Organización Meteorológica Mundial [OMM], 2020).

### **4.2.2. Precipitación**

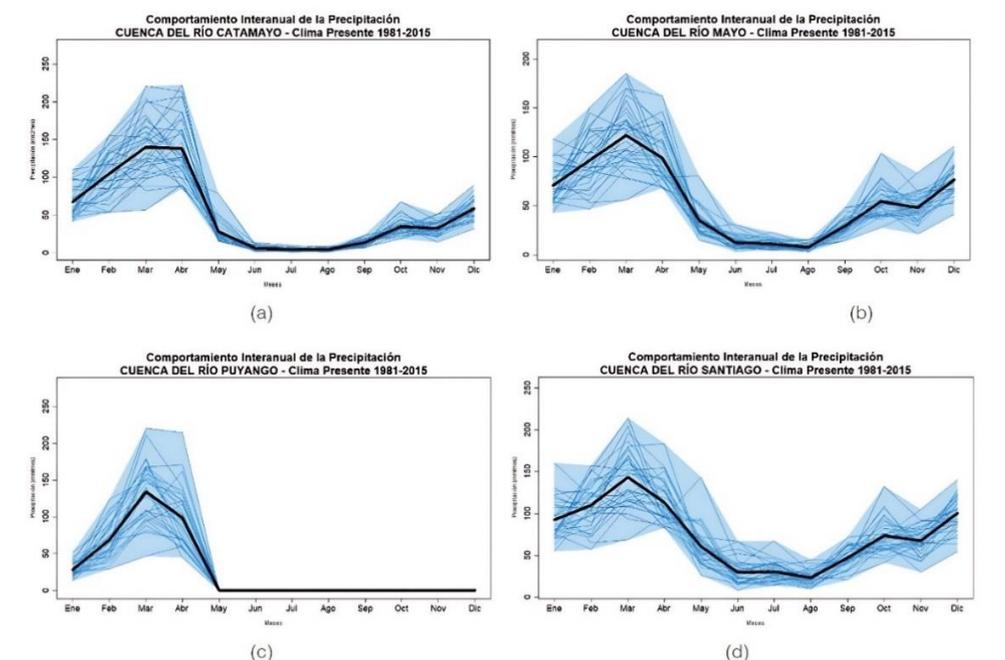
Según Luis (Molina, 2022), en meteorología, la precipitación es el fenómeno relacionado con la caída libre de agua desde el cielo en forma líquida o sólida (nieve y granizo). La precipitación es uno de los principales componentes del ciclo hidrológico, siendo la responsable de devolver la mayor parte del agua que se recicla en el planeta. La formación de gotas de agua en la atmósfera no es tan sencilla como puede parecer, ya que la probabilidad

de que varias gotas de agua choquen y se queden aglutinadas es mínima. En la atmósfera hay una gran cantidad de partículas, compuestas 12 por impurezas naturales o artificiales, y algunas de ellas son esenciales para la formación de gotas y cristales de hielo. Muchas de las partículas sólidas son fragmentos de sal común (cloruro de sodio), resultantes de las gotas de agua que se evaporan de los océanos y son transportadas por el viento. Las precipitaciones se miden comúnmente utilizando diversas unidades en Milímetros (mm), Centímetros (cm) y Pulgadas (in).

Las precipitaciones se pueden distinguir dependiendo de su origen. Estas pueden ser de convección, se originan por el ascenso de bolsas de aire caliente son las tormentas de verano, Precipitaciones Orográficas se presentan cuando masas de aire húmedo ascienden al toparse con una superficie montañosa. Los estudios de precipitación son esenciales para los estudios hidrológicos de cualquier región, para estimar la cantidad de recursos hídricos (Javier Sánchez, 2011).

De acuerdo a estudios realizados de Armenta., et al. (2019) las Cuenca San Diego, Cuenca Mayo, Cuenca Puyango y Catamayo, tiene la temporada más lluviosa tiene lugar de enero a abril y durante el descanso El año con la menor cantidad de precipitaciones la estación más seca es de junio a agosto, el rango de variabilidad de la precipitación ha sido alto especialmente en los meses de la temporada lluviosa como se aprecia en la fig. 4.

**Figura 4.**  
*Niveles de precipitaciones*



*Nota.* La línea negra indica los valores Promedio Mensual, las líneas de color azul indican el rango de Variabilidad Interanual. Fuente: (Armenta., et al. 2019).

#### **4.2.3. Humedad**

La humedad está presente en gran cantidad en la atmósfera cercana a la superficie de la tierra. Tiene dos formas de medirse: Humedad Absoluta y Humedad Relativa. Humedad absoluta: se mide en gramos de vapor por metro cúbico. Humedad relativa (Hr): está relacionado con el grado de saturación de vapor de agua en el aire, es decir, mide la cantidad de vapor de agua en el aire en relación a la cantidad máxima que puede llegar a retener y se expresa en porcentaje (Martínez et al., 2018).

#### **4.2.4. Temperatura**

El concepto de temperatura deriva de la idea de medir grados relativos de calor o frío y la observación de que los cambios en el calor de un cuerpo provocan cambios en su temperatura mientras no se produzca la fusión o ebullición. De acuerdo a (Marie & Carpi, 2003) hay tres escalas que se usan hoy en día para medir la temperatura. Escala Fahrenheit (°F), escala Celsius (°C) y escala Kelvin (K).

Los valores más comunes que se miden son la temperatura máxima y mínima, o registros continuos en el tiempo, los llamados termogramas, en tiempos fijos específicos. Estas mediciones se pueden utilizar para realizar cálculos estadísticos de explicaciones climatológicas comunes, tales como temperaturas medias diarias, mensuales, estacionales o anuales, así como valores extremos (máximas y mínimas), amplitudes térmicas que es la diferencia entre el valor máximo y mínimo desviaciones estándar, etc. (Juan Inzunza, 2007).

#### **4.2.5. Presión atmosférica**

La presión atmosférica o también conocida como presión barométrica es el peso del aire que está actuando sobre la superficie de la tierra. La atmósfera es la capa de aire que rodea la tierra. Esta capa soporta peso sobre la superficie terrestre. Esto es lo que se llama presión atmosférica. La presión del aire en un lugar no es estable y tiene valores normales, pero cambia a lo largo del día o de la hora debido al aumento o la disminución de las temperaturas o simplemente porque la atmósfera es dinámica y hay cambios en las masas de aire circundantes. Aunque no se ha demostrado que los cambios en la presión atmosférica en condiciones naturales afecten directamente a las plantas, y los animales tienden a sufrir, su verdadera importancia radica en su impacto en las características climáticas locales (Fernández, 2020). Las unidades de medida más comunes Pascal (Pa), Hectopascal (hPa), Bar y Milímetros de mercurio (mmHg). Estas unidades son utilizadas en diferentes contextos y regiones del mundo, pero en general, el hectopascal es ampliamente aceptado en meteorología y pronósticos del tiempo, mientras que el milímetro de mercurio y el bar son más comunes en contextos como la aviación y la navegación.

### **4.3. IoT**

El internet de las cosas o IoT (Internet of Things) se refiere a la red de dispositivos físicos que están conectados entre sí y a internet, permitiéndoles recopilar y compartir datos. La implementación de una red IoT puede estar enfocada a diferentes aplicaciones como en agricultura, industria, salud, ciudades inteligentes, transporte, energía y medio ambiente, entre otros (Flores & Cossio, 2021).

En base a lo mencionado el enfoque que tiene este proyecto en al medio ambiente, específicamente a la medición de variables hidrometeorológicas de forma autónoma lo que implica que el sistema va contar con sensores, algoritmos y controles programados para realizar tareas específicas.

### **4.4. Redes LPWAN**

LPWAN (Low Power Wide Area Network) es un tipo de red de área amplia diseñada para permitir la comunicación de dispositivos de bajo consumo de energía a través de largas distancias. Estas redes están optimizadas para proporcionar conectividad de manera eficiente en términos de consumo de energía y costos, lo que las hace ideales para aplicaciones de IoT (Internet de las cosas).

Las redes de área amplia (LPWAN) de baja potencia no solo permiten una comunicación de máquina a máquina (M2M) de muy baja potencia, sino también de largo alcance y baja costo. Las comunicaciones de bajo consumo y largo alcance tienen aplicaciones evidentes en áreas rurales donde la cobertura de la red celular es escasa. La infraestructura deficiente y la implementación de redes Wi-Fi o similares pueden ser costosas debido al terreno desafiante o la dispersión de los nodos de sensores en grandes áreas (Peña Queralt et al., 2019a).

#### **4.4.1. Características**

Entre las herramientas de comunicación inalámbrica más utilizadas destacan Bluetooth y WiFi, que ofrecen una excelente velocidad de transferencia de datos, pero tienen un alcance limitado. En contraste, las redes 3G, 4G y 5G ofrecen un mayor alcance, pero requieren una mayor demanda energética para la transmisión de datos por parte de los dispositivos (Carrasco & Elí, 2020). Las principales características de las redes LPWAN son las siguientes:

- Largo alcance entre base/Gateway, pueden ser superiores a los 10km dependiendo de la tecnología, y la zona geográfica.
- Bajo costo de conectividad.

- Bajo costo de los dispositivos de comunicación.
- Bajo potencia de transmisión, lo que requiere bajos niveles de consumo de energía.

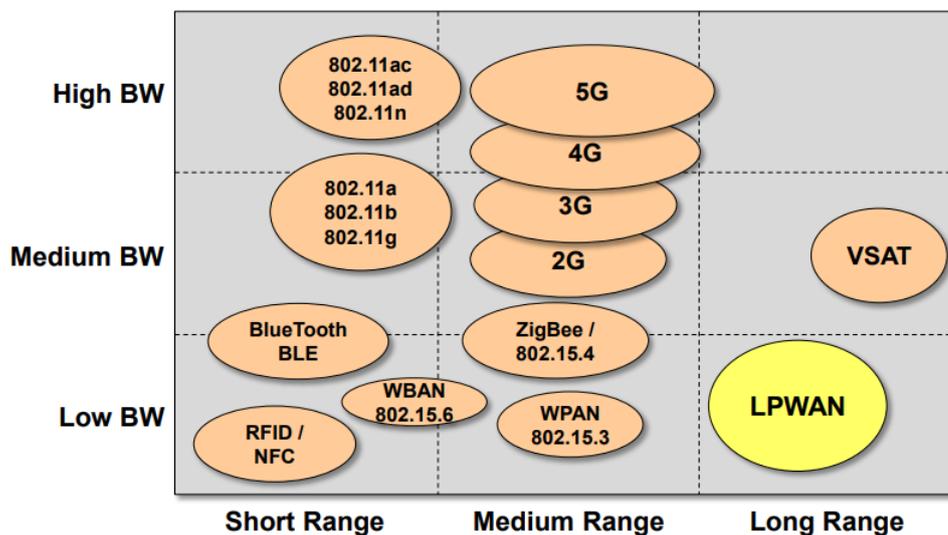
#### 4.4.2. Diferencias de LPWAN con otras Redes

Las diferentes tecnologías inalámbricas cubren diferentes aplicaciones en cuanto a alcance y ancho de banda. Aplicaciones de largo alcance con bajos requisitos de ancho de banda son típicos de IoT, en la fig. 5 se compara LPWAN con otras tecnologías inalámbricas, considerando el ancho de banda y el alcance de la señal (Egli, 2015).

Comparativa de tecnologías Inalámbricas con LPWAN

**Figura 5.**

*Comparativa de tecnologías Inalámbricas con LPWAN*



*Nota:* Comparación de LPWAN con otras redes inalámbricas en términos de ancho de banda y cobertura. Fuente: (Egli, 2015)

La gráfica muestra la diferencia de una red LPWAN que se caracteriza principalmente por su largo alcance o gran cobertura a diferencia de las demás tecnologías inalámbricas sin embargo se limita en un ancho de banda bajo lo que resulta en una transmisión de datos de baja velocidad. Estas características la convierten en una alternativa para aplicaciones de IoT que requieran transmitir pequeñas cantidades de datos, a largas distancias.

#### 4.4.3. Comparativa de las tecnologías de LPWAN

Es fundamental reconocer que cada tecnología LPWAN busca abordar una necesidad particular, lo que conlleva a la existencia de parámetros y características distintivas. Es esencial comprender estas diferencias para poder seleccionar la tecnología adecuada según la situación específica. En la tabla 1 se presenta una comparativa exhaustiva de todas las características de las diversas tecnologías de LPWAN (Hernández, 2020).

**Tabla 1.**  
Comparativa de las tecnologías LPWAN más destacadas

	<b>Sigfox</b>	<b>LoRaWAN</b>	<b>LTE-M</b>	<b>NB-IoT</b>
<b>Modulación</b>	BPSK	CSS	QPSK	QPSK
<b>Frecuencia</b>	Bandas ISM sin licencia (868MHz en Europa, 915MHz en América del Norte y 433Mhz en Asia)	Bandas ISM sin licencia (868MHz en Europa, 915MHz en América del Norte y 433Mhz en Asia).	Bandas de frecuencia LTE con licencia	Bandas de frecuencia LTE con licencia 2G
<b>ANCHO DE BANDA</b>	100 Hz	250 kHz 125 kHz	1,4 MHz	200 kHz
<b>VELOCIDAD MAX. DE DATOS</b>	100bps	50 kbps	1Mbps	100 kbps
<b>BIDIRECCIONAL</b>	Half-duplex	Half-duplex	Full-duplex	Half-duplex
<b>MÁXIMO MENSAJES/DÍA</b>	140(UP), 4(DL)	Ilimitado	Ilimitado	Ilimitado
<b>LONGITUD MÁXIMA DE CARGA ÚTIL</b>	12 bytes, 8 bytes (DL)	243 bytes	27.2 (DL), 62.5 (UL)	1600 bytes
<b>DISTANCIA</b>	10 km (urbano), 40 km (rural)	5km (urbano), 20km (rural)	1km (urbano), 5km (rural)	1km (urbano), 5km (rural)
<b>DURACIÓN BATERIA</b>	20 años	15 años	10 años	10 años
<b>INMUNIDAD A INTERFERENCIA</b>	Muy Alto	Muy Alto	Medio	Bajo
<b>AUTENTICACION Y ENCRIPCIÓN</b>	No soportado	Si (AES-128)	Si (LTE)	Si (LTE)
<b>TASA DE DATOS ADAPTATIVA</b>	No	Si	No	No
<b>HANDOVER</b>	Los dispositivos finales no se unen a una sola estación base	Los dispositivos finales no se unen a una sola estación base	Los dispositivos finales se unen a una sola estación base	Los dispositivos finales se unen a una sola estación base
<b>COMUNICACIÓN EN TIEMPO REAL</b>	No	No	Si	No
<b>LOCALIZACIÓN</b>	Si (RSSI)	Si (TDOA)	Si	No
<b>PERMITE RED PRIVADA</b>	No	Si	No	No
<b>ESTANDARIZACIÓN</b>	La compañía Sigfox está colaborando con ETSI en la estandarización de la red basada en Sigfox	LoRa-Alliance	3GPP	3GPP

*Nota:* Se presenta una comparativa de los principales parámetros que distinguen entre las tecnologías LPWAN. Fuente: (Hernández, 2020).

Al momento de seleccionar una tecnología, es crucial considerar dos características fundamentales: el coste y la velocidad. Sin embargo, alcanzar un equilibrio entre ambas es

imprescindible dadas las especificaciones requeridas por los parámetros IoT. Si realizamos una comparativa basada en estos parámetros IoT para elegir entre distintas tecnologías, en la tabla 2 se consideran los siguientes aspectos: calidad del servicio, duración de la batería, latencia, escalabilidad, longitud de la carga útil, cobertura, alcance, implementación y costo (Hernández, 2020).

**Tabla 2.**

*Comparativa de las tecnologías de LPWAN dedicados a IoT*

	<b>Sigfox</b>	<b>LoRa</b>	<b>LTE-M</b>	<b>NB-IoT</b>
<b>Calidad de servicio (QoS)</b>	Media	Media	Alta	Muy Alta
<b>Batería</b>	20 años	15 años	10 años	10 años
<b>Latencia</b>	Alta	Alta clase A y B Baja en clase C	Baja	Baja
<b>Escalabilidad</b>	Baja	Baja	Alta	Alta
<b>Rango</b>	Alto	Media	Bajo	Bajo
<b>Implementación</b>	Medio	Alto	Baja	Baja
<b>Costo</b>	Bajo	Bajo	Alto	Alto

*Nota:* Comparativas de las tecnologías LPWAN en términos de IoT. Fuente: (Hernández, 2020).

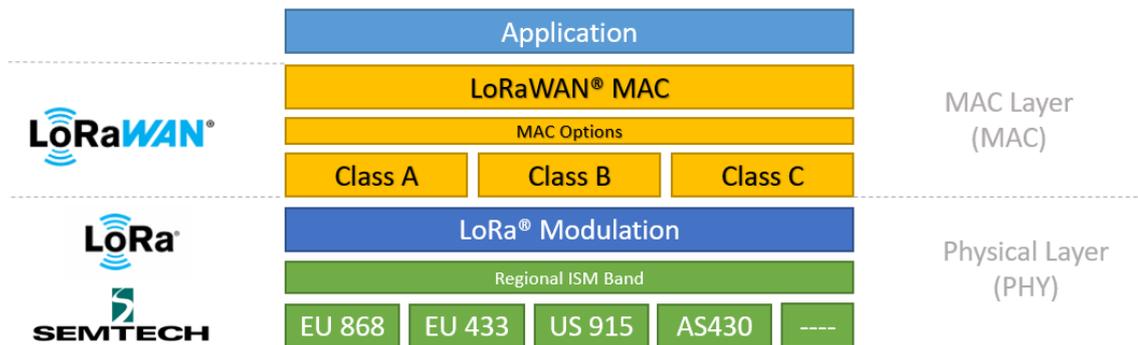
#### **4.5. Tecnología de acceso a radio**

La tecnología de acceso a radio (*RAT*) está compuesta por diferentes protocolos y estándares que permiten la conexión y transmisión de datos a través de una red inalámbrica usando señales de radiofrecuencia. La selección de *RAT* depende de diferentes factores, como lo es la accesibilidad de la red, disponibilidad de recursos, preferencia del operador o del usuario, calidad de servicio, etc.

##### **4.5.1. Características de LoRa**

LoRa es un protocolo de comunicación enfocada en LPWAN (Low Power Wide Area Network), que cuenta con grandes ventajas para proyectos de IoT. El nombre LoRa proviene del nombre de (Long Range) e incluye un método de modulación de RF patentado por Semtech. Esta es la capa física que proporciona a los dispositivos inalámbricos un largo alcance y bajo consumo de energía opera en la banda ISM en las frecuencias de 433[MHz], 868[MHz] o 915[MHz], dependiendo en la región en la que sea desplegado, sus características lo hacen muy interesante para diversas aplicaciones IoT (Semtech, 2021). A continuación, en la fig. 6, se puede apreciar las capas de operación en las que trabajan.

**Figura 6.**  
Tecnología de LoRa y LoRaWAN



*Nota:* LoRa es una tecnología de comunicación inalámbrica de largo alcance que utiliza una señal de radio y modulación específicas. Mientras que LoRaWAN es un protocolo de red abierto que permite la comunicación bidireccional segura entre dispositivos finales. Fuente: (Semtech, 2021).

#### 4.5.2. Elementos de una red LoRaWAN

Un dispositivo final en LoRaWAN es un sensor o actuador que está conectado de forma inalámbrica a una red LoRaWAN a través de puertas de enlace de radio que utilizan la modulación LoRa RF.

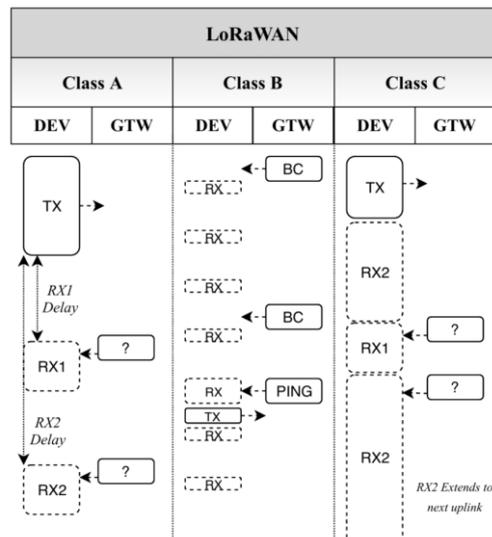
En la mayoría de las aplicaciones, el terminal es un sensor independiente, generalmente alimentado por batería, que digitaliza las condiciones físicas y los eventos ambientales. Los dispositivos finales pueden funcionar en uno de tres modos como se observa en la fig. 7 (Semtech, 2021).

- **Dispositivos de Clase A:** los dispositivos que pertenecen a esta clase pasan la mayor parte del tiempo en estado inactivo o en modo suspensión. Siempre que exista un cambio en la variable que está midiendo el sensor, el dispositivo se activa e inicia un enlace ascendente, transmitiendo a la red los datos acerca del cambio de estado. Si no recibe un enlace descendente durante esta ventana de recepción se vuelve a dormir brevemente, despertando momentos después a la espera de una respuesta. Si no se recibe respuesta durante esta segunda ventana de recepción, el dispositivo vuelve a dormir hasta la próxima vez que tenga datos para informar.
- **Dispositivos de clase B:** estos dispositivos brindan un monitoreo constante y continuo del tiempo de recepción de la red de terminales, lo que hace que los terminales de clase B sean adecuados para monitorear tanto sensores como actuadores. Todos los puntos finales basados en LoRa se inician en modo de clase A; sin embargo, los dispositivos programados

con la como Clase B en producción se pueden cambiar al modo Clase B en la capa de aplicación.

- **Dispositivos de clase C:** los dispositivos de clase C permanecen encendidos. Estos dispositivos siempre están escuchando mensajes de enlace descendente a menos que estén transmitiendo un enlace ascendente. Como resultado, aseguran la menor latencia en la comunicación entre el servidor y el terminal.

**Figura 7.**  
*Clases de dispositivos finales de LoRaWAN*



*Nota:* Ventanas Tx/Rx para LoRaWAN Clases A, B y C. Los paquetes con (?) son enlaces descendentes opcionales. Fuente: (Peña Queraltá et al., 2019b).

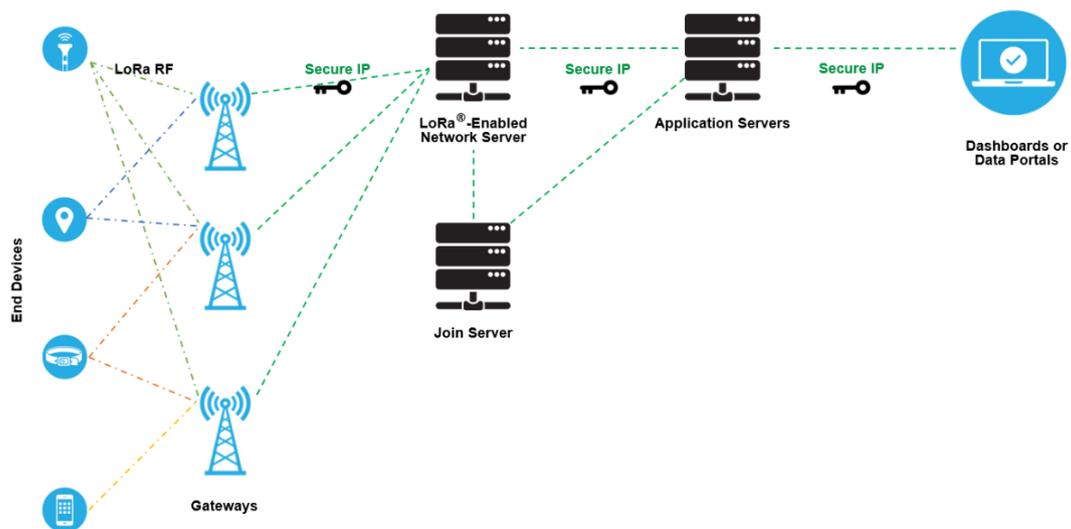
#### 4.5.3. Servidores en LoRaWAN

Como parte de la arquitectura LoRaWAN fig. 8, están presentes servidores los cuales se los describen a continuación (Semtech, 2021).

- **Servidor de red:** un servidor de red LoRaWAN se encarga de administrar toda la red, controla de forma dinámica los parámetros de la red para adaptar el sistema a las condiciones cambiantes, además establece conexiones AES seguras de 128 bits para el transporte de datos de extremo a extremo, además se encarga del control del tráfico que fluye desde el dispositivo final al LNS y viceversa. El LSN se encarga de garantizar la autenticidad de todos los sensores de la red y la integridad cada uno de los mensajes.
- **Servidor de aplicaciones:** el servidor de aplicaciones es responsable de procesar, administrar e interpretar datos seguros de las aplicaciones de sensores. También generan toda la carga útil del enlace descendente de la capa de aplicación para los terminales conectados.

- **Servidor de Unión:** el servidor de conexión gestiona la activación inalámbrica de los terminales añadidos a la red. El servidor de anclaje contiene la información necesaria para procesar tramas de solicitud de anclaje de enlace ascendente y generar tramas de confirmación de anclaje de enlace descendente. Señala al servidor de red qué servidor de aplicaciones conectar al dispositivo final y genera la clave de cifrado de sesión de red y aplicación. Reenvía la clave de sesión de red del dispositivo al servidor web y la clave de sesión de la aplicación al servidor de aplicaciones adecuado.

**Figura 8.**  
*Elementos de la red LoRaWAN*



*Nota:* En este ejemplo se muestra una implementación típica de LoRaWAN de extremo a extremo. Fuente: (Semtech, 2021).

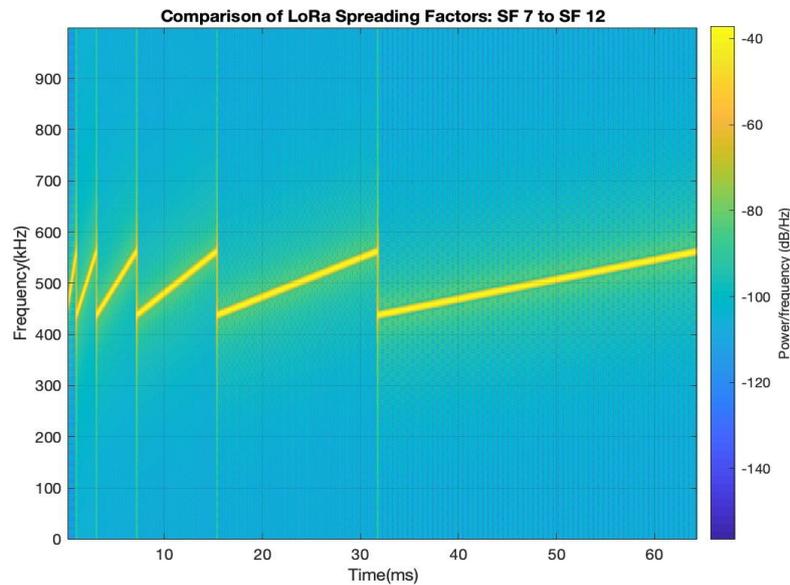
#### 4.5.4. Modulación LoRa

LoRa es un esquema patentado de modulación de espectro ensanchado basado en la modulación Chirp Spread Septum (CSS) como se ilustra en la fig. 9, el cual permite la transmisión de datos a largas distancias. La modulación (CSS) es una técnica de modulación que se utiliza en sistemas de comunicación inalámbricos para mejorar la robustez de la transmisión de datos en presencia de interferencia y ruido, Proporciona un equilibrio óptimo entre la sensibilidad para detectar señales débiles y la velocidad de transmisión de datos. Lo hace al funcionar en canales con un ancho de banda predefinido, que puede ser de 125 KHz o 500 KHz para los canales de enlace ascendente, y de 500 KHz para los canales de enlace descendente (Semtech, 2021).

Para entender cómo funciona la modulación de espectro ensanchado LoRa, es importante comenzar con el concepto de modulación DSSS (modulación de espectro ensanchado por secuencia directa), que es una técnica donde la señal portadora cambia según una secuencia de código específica, también llamada código de expansión o secuencia de chip. Esta modificación de la portadora se logra multiplicándola por el código de expansión, lo que resulta en una señal más rápida con componentes de frecuencia más altos que la señal de datos original. Esta característica permite una transmisión con un mayor alcance. Un término clave que aparece aquí es la Ganancia de Procesamiento ( $G_p$ ), que representa la relación logarítmica entre la velocidad de chips de la secuencia de código y la velocidad de bits de la señal de datos.

La Ganancia de Procesamiento ( $G_p$ ) es fundamental, ya que posibilita que el receptor pueda recuperar la señal de datos original incluso si la relación señal-ruido (SNR) es negativa. En el contexto de LoRa, su  $G_p$  es mayor que el de la modulación por desplazamiento de frecuencia FSK, lo que resulta en una menor potencia de salida del transmisor con la misma velocidad de transmisión de datos y un presupuesto de enlace similar. Aunque LoRa parte de los principios de la modulación DSSS, se diferencia al utilizar un reloj de referencia de alta precisión. Aquí es donde entra en juego la tecnología LoRa Chirp Spread Spectrum de Semtech, que elimina la necesidad de un reloj de referencia de alta precisión. En resumen, la tecnología LoRa mejora la transmisión de datos a larga distancia a través de la modulación de espectro ensanchado, combinando principios DSSS con innovaciones propias, como la eliminación de la necesidad de relojes altamente precisos (Semtech, 2021).

**Figura 9.**  
*Modulación de espectro ensanchado chirp*



*Nota.* LoRa utiliza una técnica de modulación de espectro ensanchado chirp. Los símbolos LoRa se modulan sobre un chirrido ascendente de un ancho de banda de 125 kHz, y SF 9 toma exactamente el doble de tiempo que SF 8. (Kim et al., 2019).

#### **4.5.5. Factor de dispersión**

La modulación LoRa incluye un total de seis factores de dispersión (SF7 a SF12). A medida que se emplea un factor de dispersión mayor, la señal podrá viajar distancias más largas y aún ser recibida sin errores por el receptor RF.

La tabla 3 presenta cuatro factores de dispersión distintos [SF7-SF10], que pueden emplearse para mensajes de enlace ascendente (UL) en un canal de 125 KHz. La tabla muestra la tasa de bits equivalente y el alcance estimado (esto depende del tipo de terreno; se lograrán distancias mayores en entornos rurales que en urbanos). Además, se indican los valores del tiempo de permanencia o tiempo en el aire (TOA) para una carga útil de 11 bytes correspondiente a cada uno de los cuatro factores de dispersión. En canales de 500 KHz, los mensajes de enlace descendente tienen la capacidad de utilizar los seis factores de dispersión disponibles, que van desde SF7 hasta SF12 (Semtech, 2021).

**Tabla 3.**  
*Factor de dispersión de Lora*

Factor de dispersión Para UL a 125 KHz	Tasa de Bits	Rango Depende del terreno	Tiempo en el aire para una carga útil de 11-bytes
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

*Nota.* La modulación LoRa cuenta con un total de seis factores de dispersión (SF7 a SF12).  
Fuente: (Semtech, 2021)

#### 4.5.6. Características de la modulación LoRa según su región

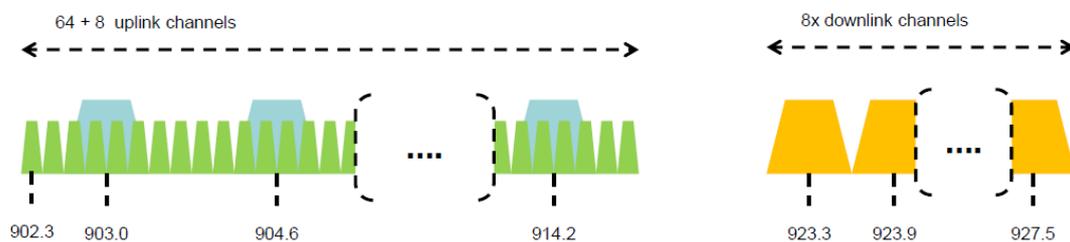
De acuerdo a los parámetros de regulación de LoRa en el Ecuador (CE) Funciona el plan de canal US902-928 (US915) con las bandas 902 – 928 MHz. La banda ISM de 915 MHz se dividirá en los siguientes planes de canales (Alliance, 2019).

*Upstream:* con 64 canales desde el 0 al 63, utilizando LoRa 125 kHz BW tal como se ilustra en la fig. 10, variando de DR0 a DR3 (*Data Rate*) usando una velocidad de codificación 4/5, comenzando en 902.3 MHz con un incremento lineal 200kHz a 914.9 MHz.

*Upstream:* con 8 canales numerados 64 a 71 utilizando LoRa 500 kHz BW en DR4 529 comenzando en 903.0 MHz con un incremento linealmente desde 1.6 MHz a 914.2 MHz.

*Downstream:* con 8 canales numerados del 0 al 7 utilizando LoRa 500 kHz BW en DR8 a 531 DR13, comenzando en 923.3 MHz e incrementándose linealmente en 600 kHz a 927.5 MHz.

**Figura 10.**  
*Frecuencia de Canal*



*Nota.* Frecuencia de canal US902-928. Fuente: (Alliance, 2019).

**Tabla 4.**  
*Características de Modulación Lora*

Uplink o Downlink	Data Rate (DR)	Factor de Dispersión (SF)	Frecuencia de Canal	Tasa de Bits	Tamaño de la carga útil (bytes)
Uplink	0	SF10	125 kHz	980 bps	11
Uplink	1	SF9	125 kHz	1,760 bps	53
Uplink	2	SF8	125 kHz	3,125 bps	125
Uplink	3	SF7	125 kHz	5,470 bps	242
Uplink	4	SF8	500 kHz	12,500 bps	242
5-7					
Downlink	8	SF12	500 kHz	980 bps	63
Downlink	9	SF11	500 kHz	1,756 bps	129
Downlink	10	SF10	500 kHz	3,125 bps	242
Downlink	11	SF9	500 kHz	5,470 bps	242
Downlink	12	SF8	500 kHz	12,500 bps	242
Downlink	13	SF8	500 kHz	21,900 bps	242

*Nota:* Características de Modulación LoRa. Fuente: (Semtech, 2021)

Existen parámetros a considerar para poder optimizar los recursos y aumentar la cobertura de una red. (Araujo et al., 2023) mencionan que el rendimiento de una red LPWAN basada en LoRa está determinado por varios factores, como el factor de dispersión (SF), el ancho de banda (Bw) y la velocidad de codificación (CR). Un SF más alto aumenta el tiempo de señal en el aire (ToA) y el consumo de energía por bit transmitido, para una recepción de distancia largar. Sin embargo, esto reduce la velocidad de datos efectiva del sistema. Por otro lado, un Bw más amplio proporciona velocidades de datos más altas pero también es más susceptible al ruido, lo que limita el rango. Además, LoRa utiliza corrección de errores hacia delante (FEC) para mejorar la confiabilidad y el alcance, pero esto agrega redundancia y aumenta ligeramente el tiempo de señal en el aire. En resumen, se debe encontrar un equilibrio entre el alcance y la velocidad de datos adecuado para cada aplicación específica, teniendo en cuenta los factores de SF, Bw y CR. A continuación, en la siguiente tabla se muestra los parámetros a considerar para la configuración de dispositivos LoRa.

**Tabla 5.***Parámetros de configuración en los dispositivos LoRa*

Parámetro	Símbolo	Valor	Unidad
Potencia transmitida	Pt	-4 a 20	dBm
Frecuencia portadora	fc	137;198;1020	MHz
Ancho de banda	Bw	7,8 a 500	KHz
Factor de Propagación	SF	7; 8;9;10;11;12	...
Tasa de Código	CR	4/5;4/6;4/7 o 4/8	...

*Nota:* Parámetros para ajustar el rendimiento del c y el consumo de energía. Fuente: (Araujo et al., 2023)

#### 4.6. The Things Network

The Things Network (TTN) es una red pública que admite un verdadero cifrado de extremo a extremo, protege contra varios ataques de intermediarios y admite diferentes claves de cifrado para cada dispositivo de entrada. TTN tiene como objetivo construir una red de datos de Internet de las cosas abierta de forma global y de colaboración colectiva, actualmente cuenta con una red de 108899 gateways distribuidos por todo el mundo. (The Things Network, 2022).

Cuenta con un acceso abierto y gratuito ofrece acceso abierto y gratuito a su infraestructura de red LoRaWAN. Esto significa que los usuarios pueden conectar sus dispositivos a la red de TTN sin incurrir en costos adicionales asociados con la construcción y operación de una red privada. Prioriza la seguridad y privacidad de los datos transmitidos a través de su red. Utiliza cifrado y autenticación para proteger la información sensible y garantizar la integridad de los datos. La versatilidad de TTN permite la implementación de una amplia gama de aplicaciones de IoT, como monitoreo ambiental, seguimiento de activos, agricultura inteligente, ciudades inteligentes y mucho más. (The Things Network, 2022).

#### 4.7. Plataformas de IoT.

En el creciente panorama de la Internet de las Cosas (IoT), la arquitectura despliega una estructura estratégica para la interconexión entre el mundo físico y los procesos empresariales. Debido a ello han surgido varias empresas que brindan la infraestructura necesaria para la implementación de este tipo de proyecto. Algunas de las principales plataformas incluyen:

- Amazon Web Service (AWS) IoT platform
- Microsoft Azure IoT.
- Google Cloud IoT
- Cisco IoT
- IBM Watson IoT platform

#### 4.7.1. Comparativa de las plataformas IoT

La infraestructura y los servicios que ofrecen las plataformas de IoT resultan ser bastantes amplias, las cinco opciones mencionadas tienen una solución central con funcionalidad básica, además de un conjunto de módulos adicionales que se pueden agregar según sea necesario. La tabla 6 se muestra una tabla comparativa de las plataformas IoT (Altexsoft, 2020).

**Tabla 6.**  
*Comparativa de las plataformas IoT*

	Protocolos de comunicación	Principales funciones	Soluciones informáticas de borde.	Casos de uso principales
<b>Amazon web services</b>	HTTP MQTT WebSockets	AWS IoT Core: <ul style="list-style-type: none"> <li>• Conectividad</li> <li>• Autenticación</li> <li>• Motor de reglas</li> <li>• Entorno de desarrollo</li> </ul>	<b>FreeRTOS</b> Sistema operativo de borde.  <b>IoT GreenGrass</b> Plataforma informática de borde	<ul style="list-style-type: none"> <li>• Ciudad inteligente</li> <li>• Hogar conectado</li> <li>• Agricultura</li> </ul>
<b>Microsoft Azure</b>	HTTP MQTT AMQP over WebSockets	Azure IoT Hub: <ul style="list-style-type: none"> <li>• Conectividad</li> <li>• Monitoreo de dispositivos</li> <li>• Gestión de dispositivos</li> <li>• IoT Edge</li> </ul>	<b>IoT Edge</b> Como parte integral de IoT Hub.	<ul style="list-style-type: none"> <li>• Cuidado de la salud</li> <li>• Comercio minorista</li> <li>• Producción</li> <li>• Ciudades Inteligentes</li> <li>• Agricultura de Precision</li> <li>• Gestión de Edificios Inteligentes</li> </ul>
<b>Google Cloud</b>	HTTP MQTT	Google Cloud IoT core: <ul style="list-style-type: none"> <li>• Conectividad</li> <li>• Gestión de dispositivos</li> </ul>	<b>Edge TPU chip</b> Habilitación de IA en equipos de borde.	<ul style="list-style-type: none"> <li>• Energía</li> <li>• Aparcamiento Inteligente</li> <li>• Transporte y Logística</li> </ul>
<b>Cisco</b>	MQTT	Cisco IoT control Center: <ul style="list-style-type: none"> <li>• Conectividad móvil</li> <li>• eSIM como servicio</li> <li>• Machine Learning para mejorar la seguridad</li> </ul>	<b>Cisco iOX</b> Plataforma de desarrollo de borde  <b>Cisco Edge Intelligence</b>	<ul style="list-style-type: none"> <li>• Vehículos Conectados</li> <li>• Industria</li> <li>• Ciudades Inteligentes</li> </ul>
<b>IBM Watson</b>	HTTP MQTT	IBM Watson IoT Platform: <ul style="list-style-type: none"> <li>• Conectividad</li> <li>• Gestión de dispositivos</li> <li>• Análisis en tiempo real</li> <li>• Blockchain</li> </ul>	<b>IBM Edge Application Manager Platform</b>	<ul style="list-style-type: none"> <li>• Manufactura</li> <li>• Agricultura</li> <li>• Edificios Inteligentes</li> </ul>

*Nota:* Las cinco opciones tienen una solución principal con funcionalidad básica y un conjunto de módulos adicionales que puede agregar cuando sea necesario. Adaptado de: (Altexsoft, 2020)

#### 4.7.2. Selección de la Plataforma IoT

Las características de las plataformas de IoT se superponen en gran medida, aunque se presentan de manera diferente (Altexsoft, 2020). Además de ofrecer una funcionalidad robusta, todas ellas garantizan:

- Alta escalabilidad, adaptable a las necesidades de cualquier tipo de negocio, desde startups hasta empresas con millones de dispositivos.
- Seguridad integrada en cada capa de un sistema IoT.
- Soporte técnico y documentación detallada de sus productos.

**Tabla 7.**

*Comparativa de los precios y niveles gratuitos*

Plataforma	Plan de precios	Nivel gratuito	Nivel gratuito en servicios de lote adicionales
<b>Amazon IoT Core</b>	<ul style="list-style-type: none"> <li>• Conectividad: \$ 0.08 por millón de minutos de conexión.</li> <li>• Mensajería: entre \$0.7-1 por millón de mensaje.</li> <li>• Gestión de dispositivos: \$1.25 por millón de operaciones.</li> <li>• Motor de reglas: \$0.15 por millón de reglas activadas/acciones ejecutadas</li> </ul>	<p>Disponibles por 12 Meses</p> <ul style="list-style-type: none"> <li>• 2.250.000 minutos de conexión.</li> <li>• 500.000 mensajes</li> <li>• 225.000 registros de dispositivos operativos</li> <li>• 225.000 reglas activadas</li> <li>• 225.000 acciones ejecutadas</li> </ul>	<p>Disponibles por 12 Meses</p> <ul style="list-style-type: none"> <li>• Gestión de dispositivos: 50 remotas acciones por mes</li> <li>• AWS Greengrass: 3 dispositivos</li> <li>• Eventos de AWS IoT: 250.000 evaluaciones de mensajes por mes</li> <li>• AWS IoT Analytics: 100 MB de procesos de datos y 10 GB de almacenamiento de datos.</li> </ul>
<b>Azure IoT Hub</b>	<ul style="list-style-type: none"> <li>• Nivel 24 básico: \$10 a 2500 por unidad/ por mes.</li> <li>• Nivel estándar: \$25 a 2500 por unidad/ por mes.</li> <li>• El precio dentro del nivel depende de la cantidad de mensajes intercambiados por día (hasta 400.000, 6 millones o 300 millones)</li> </ul>	<p>Hasta 8.000 mensajes por día y hasta 500 dispositivos registrados</p>	<ul style="list-style-type: none"> <li>• Prueba gratuita de 12 meses de servicios populares de Azure.</li> <li>• Crédito de \$200 para explorar Azure durante 30 días.</li> <li>• Ofrece 25 GB de almacenamiento de datos en Cosmos DB.</li> <li>• Mas de 25 servicios siempre gratuitos</li> </ul>
<b>Google Cloud IoT Core</b>	<p>\$0.00045-0.0045 por MB de datos intercambiados.</p>	<p>Primeros 250MB</p>	<ul style="list-style-type: none"> <li>• Prueba gratuita de 12 meses con crédito de \$300 para adquirir cualesquiera servicios.</li> <li>• Gran conjunto de recursos siempre gratuitos.</li> </ul>
<b>Cisco IoT Control Center</b>	<p>Estos detalles están disponibles bajo pedido</p>	<p>Sin nivel gratuito</p>	<p>Sin nivel Gratuito</p>

<b>IBM Watson IoT Platform</b>	Desde \$500 por unidad/por mes	Sin nivel gratuito	Sin nivel gratuito
--------------------------------	--------------------------------	--------------------	--------------------

*Nota:* Todas las cinco plataformas adoptan un modelo de pago por uso, lo que significa que el costo total está directamente ligado al volumen de utilización, ya sea en términos de mensajes enviados, megabytes intercambiados, dispositivos conectados, acciones ejecutadas, entre otros factores. Adaptado de: (Altexsoft, 2020)

Para abordar la complejidad del modelo de precios, Amazon, Google y Microsoft ofrecen a sus clientes una calculadora de precios que se encuentra disponible en sus respectivos sitios web oficiales. Sin embargo, a menudo, la única forma de estimar el costo de un proyecto de IoT es ejecutarlo durante un mes y pagar la factura resultante. En cuanto a IBM y Cisco, son las plataformas con más costo. Además, Amazon, Google y Microsoft son independientes del hardware y admiten soporte de dispositivos. ofrecen la posibilidad de realizar una prueba de concepto utilizando sus planes de nivel gratuito. Esto le permite evaluar si la plataforma cumple con sus requisitos antes de comprometerse a invertir en servicios específicos.

Es importante destacar que Azure es una plataforma independiente del hardware y es compatible con una amplia variedad de dispositivos IoT. Con su nivel gratuito, permite hasta 8,000 mensajes por día y hasta 500 dispositivos registrados. Además, Azure ofrece una amplia gama de servicios populares gratuitos durante 12 meses, un crédito de \$200 para explorar Azure durante 30 días, y más de 25 servicios siempre gratuitos. Esta combinación de funcionalidades y beneficios hace de Azure una alternativa sólida y atractiva para la implementación de nuestro proyecto centrado en la recolección de datos hidrometeorológicos.

#### **4.8. Microsoft Azure**

Microsoft Azure es una plataforma de servicios en la nube proporcionada por Microsoft que ofrece una amplia gama de herramientas y recursos para ayudar a empresas y desarrolladores a construir, implementar y administrar aplicaciones y servicios en la nube de manera eficiente y escalable. Azure es conocido como uno de los principales proveedores de servicios de computación en la nube a nivel mundial.

##### **4.8.1. Centro de IoT de Microsoft Azure**

Microsoft Azure IoT Hub es un servicio en la nube que forma parte de la plataforma Azure y está diseñado específicamente para habilitar y gestionar soluciones de Internet de las cosas (IoT) de manera segura y escalable. IoT Hub facilita la conectividad y comunicación bidireccional entre dispositivos IoT y la nube, permitiendo a las organizaciones recopilar, analizar y actuar sobre datos de dispositivos en tiempo real (Microsoft, 2022c).

A continuación, se presentan algunas características clave de Azure IoT Hub:

- **Conectividad de dispositivos:** IoT Hub admite una amplia variedad de protocolos de comunicación, lo que permite a los dispositivos IoT conectarse a la nube de manera segura, ya sea a través de MQTT, AMQP, HTTPS o MQTT sobre WebSockets.
- **Escalabilidad y fiabilidad:** IoT Hub está diseñado para manejar una gran cantidad de dispositivos conectados, lo que permite a las empresas escalar sus soluciones de IoT para satisfacer las demandas cambiantes sin preocuparse por la infraestructura subyacente.
- **Seguridad avanzada:** el servicio proporciona una capa de seguridad sólida con autenticación a nivel de dispositivo, cifrado de extremo a extremo y control de acceso basado en roles, lo que garantiza la protección de los datos y la integridad de las comunicaciones.
- **Gestión de dispositivos:** IoT Hub permite el registro, seguimiento y administración de dispositivos de manera centralizada, lo que facilita la gestión de flotas de dispositivos IoT a gran escala.
- **Análisis en tiempo real:** los datos recopilados de los dispositivos se pueden analizar en tiempo real utilizando servicios de Azure como Azure Stream Analytics, lo que permite tomar decisiones informadas y oportunas.
- **Integración con servicios de Azure:** IoT Hub se integra perfectamente con otros servicios de Azure, como Azure Functions, Azure Logic Apps y Azure Event Grid, lo que permite crear flujos de trabajo complejos y soluciones personalizadas de IoT.
- **Compatibilidad con el ecosistema de IoT:** IoT Hub es compatible con una amplia variedad de dispositivos y plataformas, lo que permite a los desarrolladores y fabricantes de dispositivos elegir la mejor opción para sus soluciones de IoT.

#### **4.8.2. Funciones de Microsoft Azure**

Es un servicio de cómputo en la nube ofrecido por Microsoft Azure que permite a los desarrolladores ejecutar código de manera serverless (sin necesidad de administrar la infraestructura subyacente) y escalar automáticamente según la demanda. Con Azure Functions, los desarrolladores pueden implementar pequeñas unidades de funcionalidad, denominadas funciones, que responden a eventos y desencadenadores específicos, como solicitudes HTTP, mensajes en una cola, eventos de IoT, cambios en bases de datos, entre otros (Microsoft, 2022b).

Características clave de Azure Functions:

- **Serverless:** los desarrolladores pueden concentrarse en el código y la lógica de su aplicación, sin preocuparse por la administración de servidores o la infraestructura. La

plataforma se encarga automáticamente de la implementación, escalado y administración de los recursos necesarios para ejecutar las funciones.

- **Escala automática:** el número de instancias de funciones en función del volumen de eventos que se reciban. Esto garantiza que las aplicaciones puedan manejar grandes cantidades de eventos sin interrupciones ni problemas de rendimiento.
- **Event-driven:** las funciones de Azure se ejecutan en respuesta a eventos específicos, como solicitudes HTTP, mensajes en una cola o cambios en una base de datos. Esto permite que las funciones sean altamente reactivas y se activen solo cuando se necesita ejecutar una acción.
- **Soporte de múltiples lenguajes:** Azure Functions admite varios lenguajes de programación, incluyendo C#, JavaScript, Python, Java, TypeScript y PowerShell, lo que brinda a los desarrolladores la flexibilidad de elegir el lenguaje que mejor se adapte a sus necesidades.
- **Integración con servicios de Azure:** las funciones de Azure se pueden integrar fácilmente con otros servicios de Azure, como Azure Blob Storage, Azure Cosmos DB, Azure IoT Hub, entre otros. Esto permite crear flujos de trabajo complejos y soluciones completas que aprovechen el ecosistema de Azure.
- **Facturación basada en el consumo:** Azure Functions ofrece un modelo de precios basado en el consumo, lo que significa que solo se paga por el tiempo de ejecución y los recursos utilizados por las funciones. Esto hace que sea una opción rentable, especialmente para cargas de trabajo con fluctuaciones de uso.

#### **4.8.3. Base de Datos Cosmos de Microsoft Azure**

Azure Cosmos DB es un servicio de base de datos en la nube de Microsoft Azure, diseñado para proporcionar almacenamiento de datos de alta disponibilidad, escalable y globalmente distribuido para aplicaciones modernas. Se trata de una base de datos NoSQL altamente flexible que admite múltiples modelos de datos, lo que permite a los desarrolladores construir aplicaciones con diferentes necesidades de almacenamiento de datos (Microsoft, 2022a).

Características clave de Azure Cosmos DB:

- **Escalabilidad global:** Cosmos DB está diseñado para escalar de forma horizontal y global. Permite distribuir datos a través de múltiples regiones de Azure para lograr baja latencia y alta disponibilidad, lo que garantiza que las aplicaciones puedan mantener un rendimiento óptimo en cualquier parte del mundo.

- **Consistencia configurable:** Cosmos DB ofrece opciones de consistencia configurables, lo que permite a los desarrolladores equilibrar la coherencia y el rendimiento según las necesidades de la aplicación. Se puede elegir entre consistencia fuerte, eventual o consistencia coherente en sesión.
- **SLA garantizado:** Azure Cosmos DB ofrece un SLA (Acuerdo de Nivel de Servicio) del 99.99% para lecturas y escrituras, lo que garantiza un alto nivel de disponibilidad y rendimiento para las aplicaciones.
- **Seguridad y cumplimiento:** el servicio ofrece varias capas de seguridad, incluyendo cifrado en reposo y en tránsito, autenticación basada en roles y control de acceso, para garantizar la protección de los datos almacenados.
- **Integración con Azure y herramientas populares:** Azure Cosmos DB se integra fácilmente con otros servicios de Azure, como Azure Functions, Azure Logic Apps y Azure Event Grid. Además, ofrece conectores para herramientas populares como Apache Spark, Apache Kafka y MongoDB.

#### 4.9. Métodos para la medición del nivel de Agua

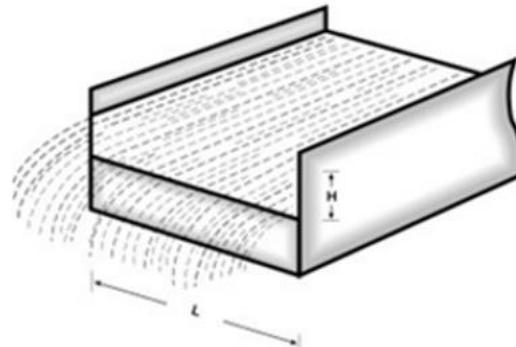
Consiste en interponer un muro ante una corriente de agua, de manera que se dé una caída de agua que pase a través de una sección predeterminada. Conociendo las características del vertedero o del orificio que se engloban en el factor (determinado experimentalmente), la sección de la lámina de agua que pasa por ellos y la velocidad teórica de caída libre. (Alvarado, 2017).

##### 4.9.1. *Vertedero rectangular sin contracciones*

Este vertedero es uno de los más sencillos en su construcción, además cuenta con una sección de caudal en forma de rectángulo, además cuenta con una cresta biselada o cortada en declive para lograr un borde fino. La exactitud de la medición que brinda varía entre un 3% y 5% ((Lopez & Zurita, 2016).

**Figura 11.**

*Vertedero rectangular rectangular sin contracciones*



*Nota:* este diseño de vertedero forma parte de uno de los más simples. Fuente: (Lopez & Zurita, 2016).

$$Q = 1.84 Lx H^{3/2} \quad (1)$$

Q = Caudal que fluye por el vertedero, en m<sup>3</sup> /s

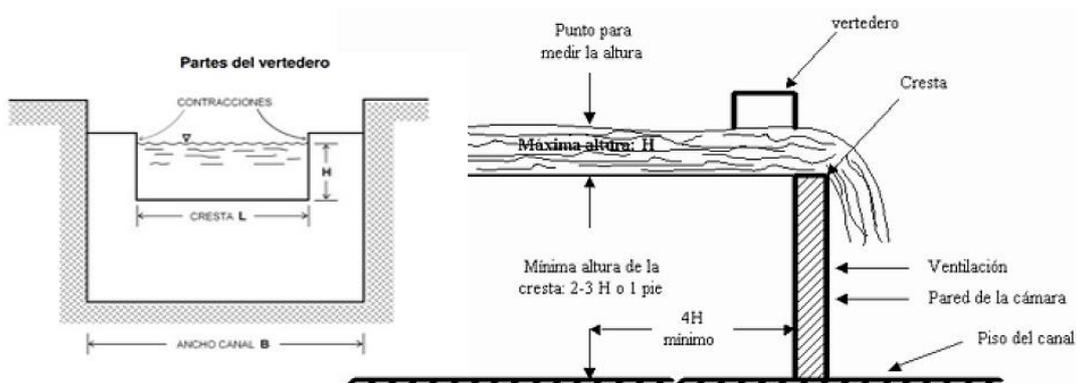
L = Ancho de la cresta, en m

H = Carga del vertedero, en m

#### **4.9.2. Vertedero rectangular con dos contracciones**

**Figura 12.**

*Vertedero rectangular con dos contracciones*



*Nota:* Vertedero rectangular con dos contracciones. Fuente: (Lopez & Zurita, 2016).

$$Q = 1.84 (L - 0.2H) x H^{3/2} \quad (2)$$

Donde:

Q = Caudal que fluye por el vertedero, en m<sup>3</sup>/s

L = Ancho de la cresta, en m

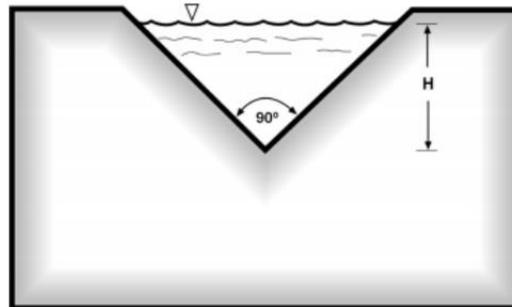
H = Carga del vertedero, en m

#### **4.9.3. Vertedero triangular**

El modelo del vertedero se puede apreciar en la fig. 13, el modelo permite obtener medidas más precisas de las alturas de carga (H) correspondientes a caudales reducidos. La

construcción se suele elaborar con placas metálicas. En la práctica únicamente se utilizan los que tienen forma de triángulos isósceles, siendo los más usuales los de 90° (Lopez & Zurita, 2016).

**Figura 13.**  
*Vertedero triangular*



Nota: Vertedero triangular. Fuente:(Alvarado, 2017).

$$Q = 1.4 \times H^{5/2} \quad (3)$$

Donde:

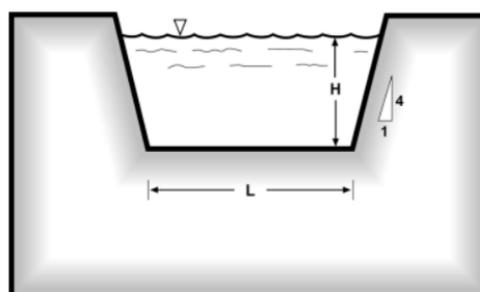
Q = Caudal que fluye por el vertedero, en m<sup>3</sup> /s

H = Carga del vertedero, en m

#### 4.9.4. *Vertedero Trapezoidal*

En la fig. 14, se muestra el diseño trapezoidal Tiene una forma trapezoidal en la apertura. Además, este diseño evita la necesidad de corregir los cálculos. El ángulo de talud es 1:4 (1 horizontal para 4 vertical).

**Figura 14.**  
*Vertedero Trapezoidal*



Nota: Vertedero Trapezoidal. Fuente:(Alvarado, 2017).

$$Q = 1.859 \times L \times H^{3/2} \quad (4)$$

Donde:

Q = Caudal que fluye por el vertedero, en m<sup>3</sup> /s

L = Ancho de la cresta, en m

H = Carga del vertedero, en m

## 5. Metodología

Para el desarrollo del trabajo de titulación se utilizaron los métodos cuantitativo y experimental.

### **Método cuantitativo**

Este método permite medir y describir fenómenos numéricamente y se basa en la recolección y análisis de datos cuantificables como información de las variables hidrometeorológicas que se obtienen con sensores.

### **Método Experimental**

Este método se caracteriza por observar, manipular y registrar distintas variables, el cual es de gran ayuda ya que se puede poner en práctica los diferentes conocimientos adquiridos en el diseño y construcción de un sistema de transmisión basado en LoRaWAN, así como en el sistema hidrometeorológico.

### **Fase 1: Diseño y construcción**

En esta fase, se llevaron a cabo las siguientes actividades:

1. **Selección del sitio estratégico:** a través de una inspección detallada en el parque PUEAR, se determinaron los puntos óptimos para la ubicación de las estaciones. El afluente de los Nogales fue seleccionado como el sitio ideal para la instalación de la estación hidrométrica y la meteorológica.
2. **Diseño del vertedero triangular:** se diseñó un modelo de vertedero triangular, seleccionado por su habilidad para proporcionar mediciones precisas de la altura del nivel de agua en el afluente. Esta elección se fundamentó en la necesidad de obtener datos hidrometeorológicos confiables
3. **Selección de sensores:** se determinaron los sensores adecuados para la captura de las variables hidrometeorológicas requeridas. Estos sensores fueron elegidos con base en su capacidad para medir de manera precisa y constante los parámetros relevantes.

### **Fase 2: Implementación de la red de comunicación LoRaWAN**

Durante esta fase, se realizaron las siguientes acciones:

1. **Elección de tecnología de comunicación:** se eligió una red LPWAN basada en la tecnología LoRa y LoRaWAN por su capacidad de ofrecer comunicaciones de largo alcance con bajo consumo energético, esencial para una monitorización autónoma.
2. **Selección y configuración del gateway:** se implementó un Gateway LoRaWAN que opera en las bandas de frecuencia 902 - 928 MHz. El acceso a internet del Gateway se estableció a través de un puerto Ethernet.

3. **Implementación del transmisor (Tx):** se eligió el nodo final Wifi LoRa 32 como transmisor, debido a sus características de conectividad y bajo consumo energético. En la fig.16 se observa cómo se divide este proceso en fases para su diseño y construcción.
4. **Integración con The Things Network (TTN):** se realizó la integración de los dos nodos finales o estaciones con la plataforma TTN. Esta plataforma administra de forma dinámica los parámetros de la red, asegurando un transporte eficiente de datos de manera integral, desde el origen hasta el destino.

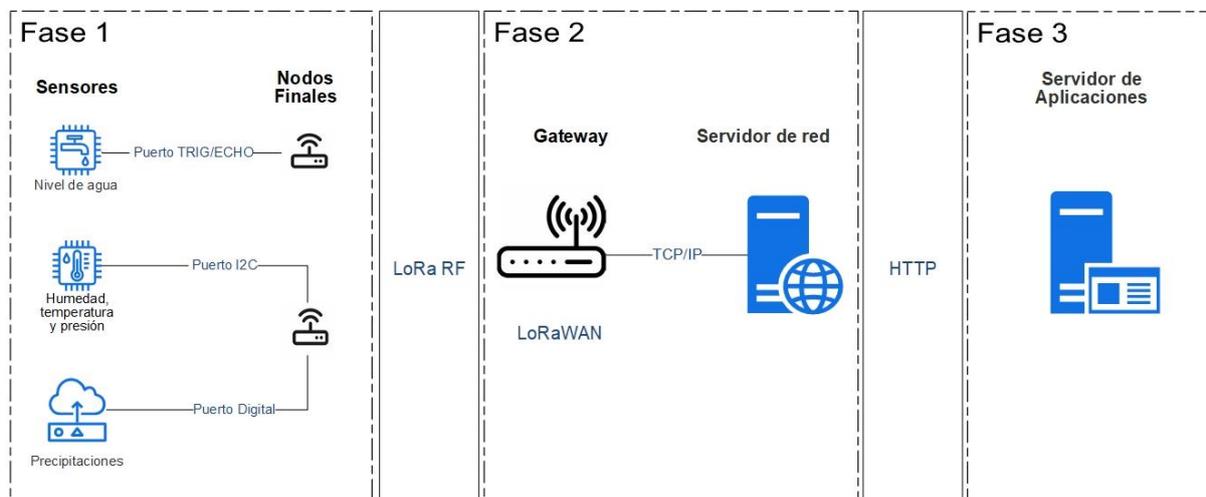
### Fase 3: Integración y comunicación con servidores

En esta etapa, se llevaron a cabo los siguientes procedimientos:

1. **Conexión con Servidores Azure:** se llevó a cabo una conexión con diversas herramientas de Azure: IoT Hub, Azure Functions y Cosmos DB. Esta integración proporcionó la capacidad de automatizar tanto el procesamiento como el almacenamiento de los datos hidrometeorológicos.
2. **Visualización de datos:** a través de la integración con Power BI, se consiguió una representación clara y comprensible de los datos recabados, lo que optimizó el proceso de interpretación y análisis de la información.

**Figura 15.**

*Fases para el diseño de cada sistema*



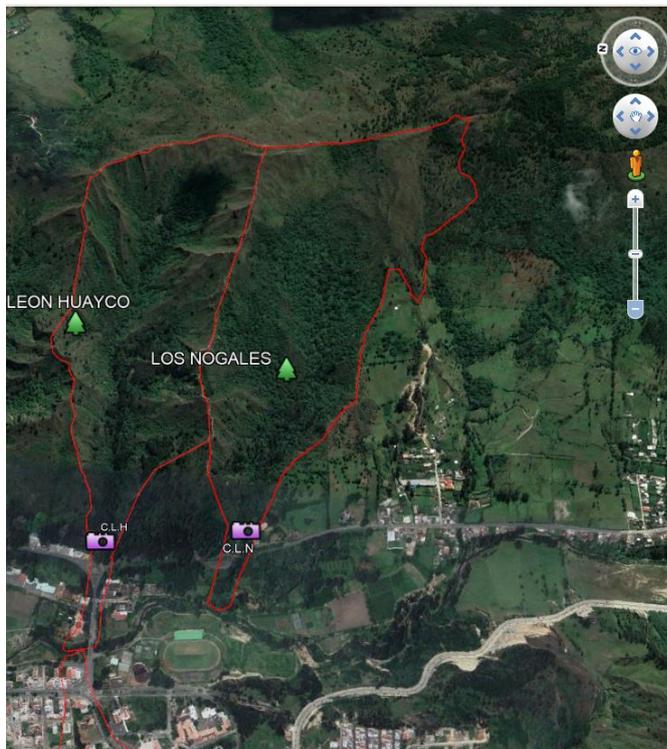
Nota: En cada fase se explica los pasos a seguir para la construcción de cada sistema. Fuente: Autor.

### 5.1. Estudio de Campo

En la fig. 16 a), se muestran las cuencas hidrográficas León Huayco y Los Nogales con las que cuenta el parque PUEAR. Durante el recorrido que se realizó en el estudio de campo se pudo apreciar que en la cuenca de los Nogales existe una densa vegetación que

aporta en la retención y suministro de agua constante durante todo el periodo del año. Mientras que en la cuenca León Huayco no existe un suministro de agua debido a varios factores como la poca vegetación, así como el terreno el cual es muy inclinado.

**Figura 16.**  
*Parque PUEAR*



*a). Cuencas hidrográficas pertenecientes al parque PUEAR*



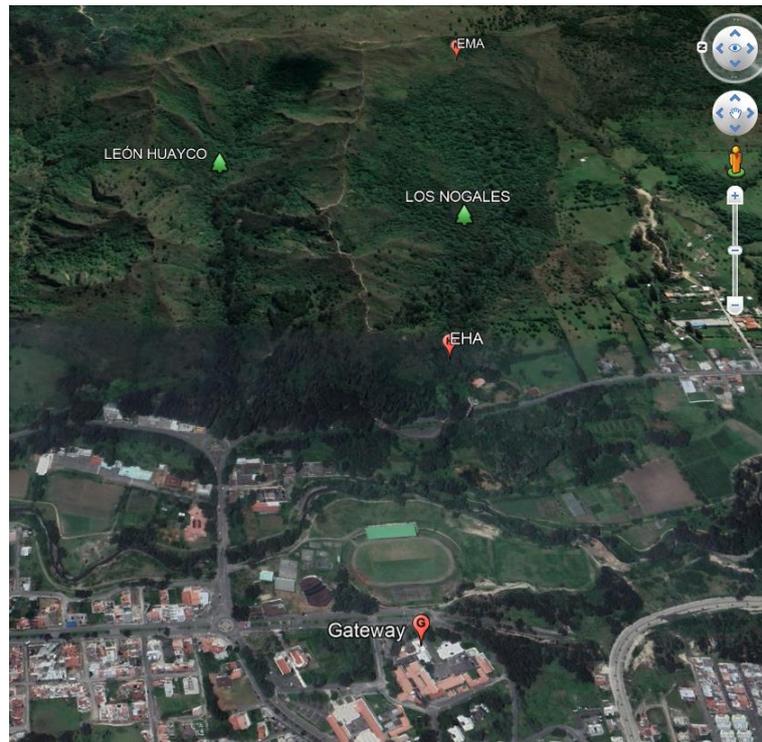
*b). Cuenca León Huayco*



*c). Cuenca Los Nogales*

Tras analizar ambas cuencas, se determinó que la cuenca de Los Nogales es la más adecuada para la instalación de las estaciones. La estación hidrométrica, localizada en las coordenadas (Latitud: 4° 2'12.27"S, Longitud: 79°11'49.58"O), será responsable de medir los niveles de agua en el vertedero. Por otro lado, la estación meteorológica, situada en (Latitud: 4° 2'17.55"S, Longitud: 79°11'18.00"O), tendrá la tarea de registrar variables como temperatura, humedad, presión atmosférica y precipitaciones.

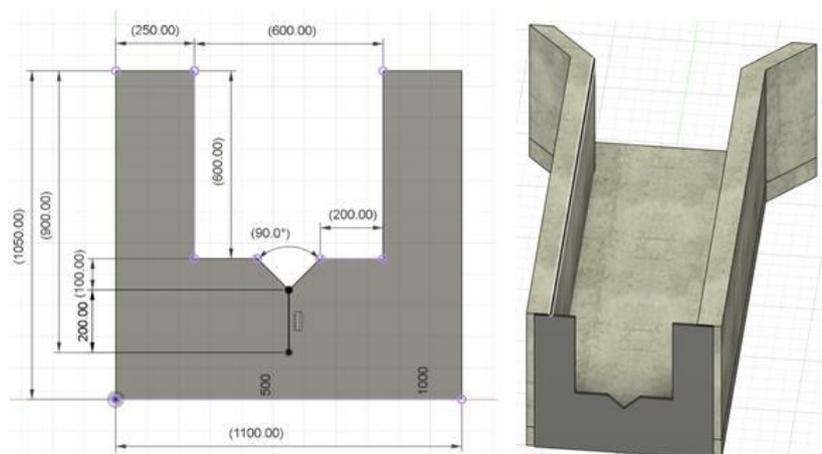
**Figura 17.**  
*Ubicación del gateway y las estaciones*



Fuente: Autor

Se instaló un vertedero de forma triangular con el objetivo de mejorar la precisión en la medición del nivel de agua en la cuenca Los Nogales. El diseño de esta construcción se encuentra ilustrado en la fig. 18.

**Figura 18.**  
Vertedero triangular



Nota: Se optó por un diseño de vertedero triangular debido a su capacidad para ofrecer mediciones exactas de la elevación del agua en el curso de agua.

## 5.2. Diseño y Arquitectura de la Red LoRaWAN

El diagrama de la red implementada se ilustra en la fig. 19, el cual consta de tres bloques principales que clasifican los equipos y su conectividad.

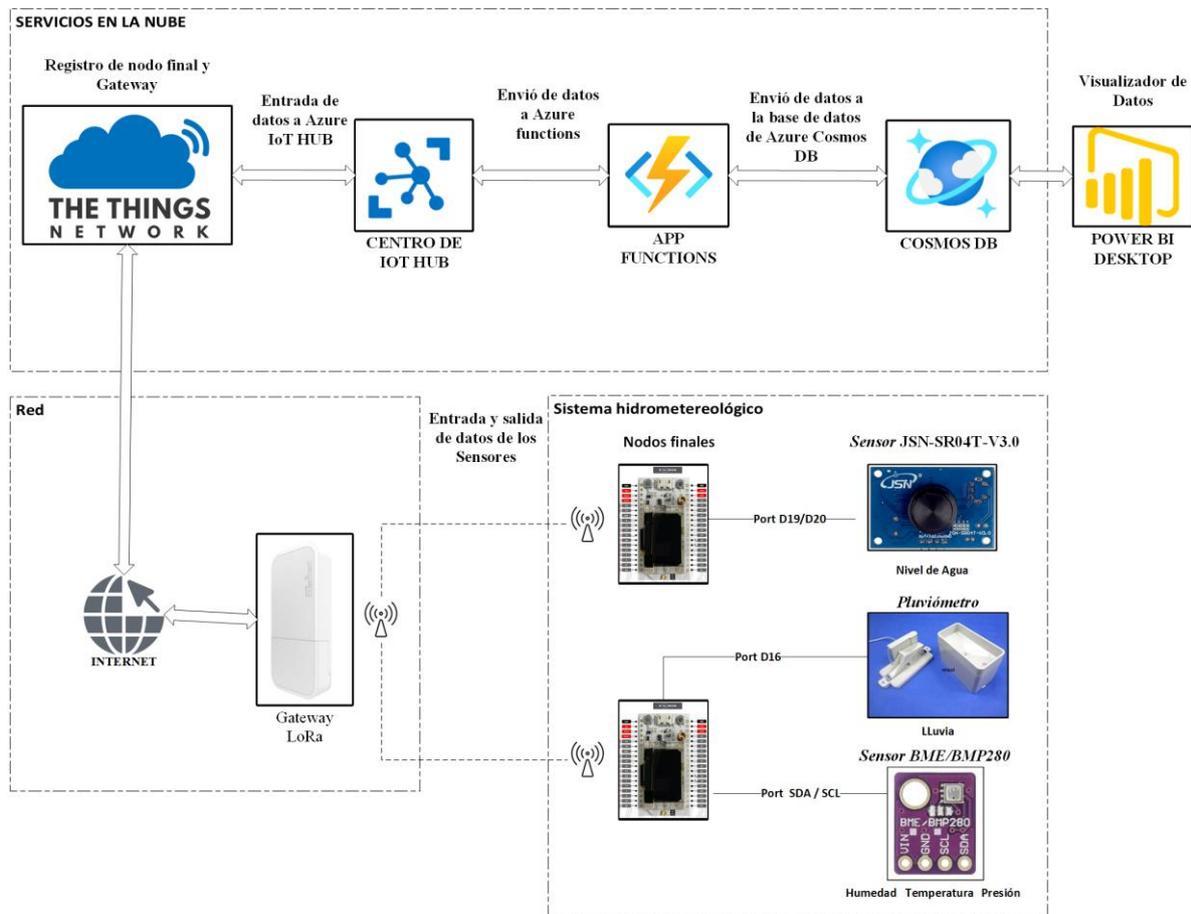
El sistema hidrometeorológico corresponde a los nodos finales conformados por el modelo Heltec WiFi LoRa 32. El primer nodo corresponde al nodo denominado EHA el cual se conecta el sensor ultrasónico JSN-SR04T-V3.0 por dos pines digitales los cuales corresponde a Trig/Echo. En el segundo nodo denominado EMA se encuentran conectados el sensor BME/BMP280 por medio del puerto I2C y el pluviómetro misol WH-SP-RG conectado a un pin digital. La comunicación de los nodos con el Gateway lo realizan mediante la modulación LoRa, y su configuración se la realizo teniendo en cuenta la región geográfica específica. En este contexto, se optó por operar a una frecuencia de 915 MHz, la cual está recomendada para el ámbito latinoamericano. Además, se incorporó el método de activación OTAA (Over-The-Air Activation).

El bloque denominado Red representa la conexión entre el Gateway, los nodos distribuidos y el servidor de The Things Network (TTN). La configuración y establecimiento de esta conexión se llevan a cabo mediante el protocolo TCP/IP, conectando el Gateway al servidor de red por medio de la siguiente dirección <https://nam1.cloud.thethings.network/>.

Como tercer bloque tenemos los servicios en la nube este bloque abarca la integración y configuración del Gateway, así como de las aplicaciones en TTN en la cual se registran los nodos finales. A continuación, se incluye la integración con los servicios de Azure.

En las secciones que siguen a continuación se muestran el proceso realizado para la configuración de los equipos y la creación de cada servicio en la nube.

**Figura 19.**  
*Diagrama de Red*



*Nota:* Arquitectura de la Red a implementar. Fuente: Autor

### 5.3. Configuración de Gateway

#### Paso 1. Configuración del Gateway wAP LR9

- Conecte el dispositivo y espere hasta que la red WiFi del equipo esté activa para poder acceder a ella.
- Acceda a la interfaz de configuración mediante un navegador web, utilizando la dirección IP 192.168.88.1.
- Al ingresar por primera vez, utilice las credenciales de inicio de sesión siguientes: Login: admin, Password: admin, tal como se muestra en la fig. 20.

**Figura 20.**

*Acceso administrativo al Gateway*

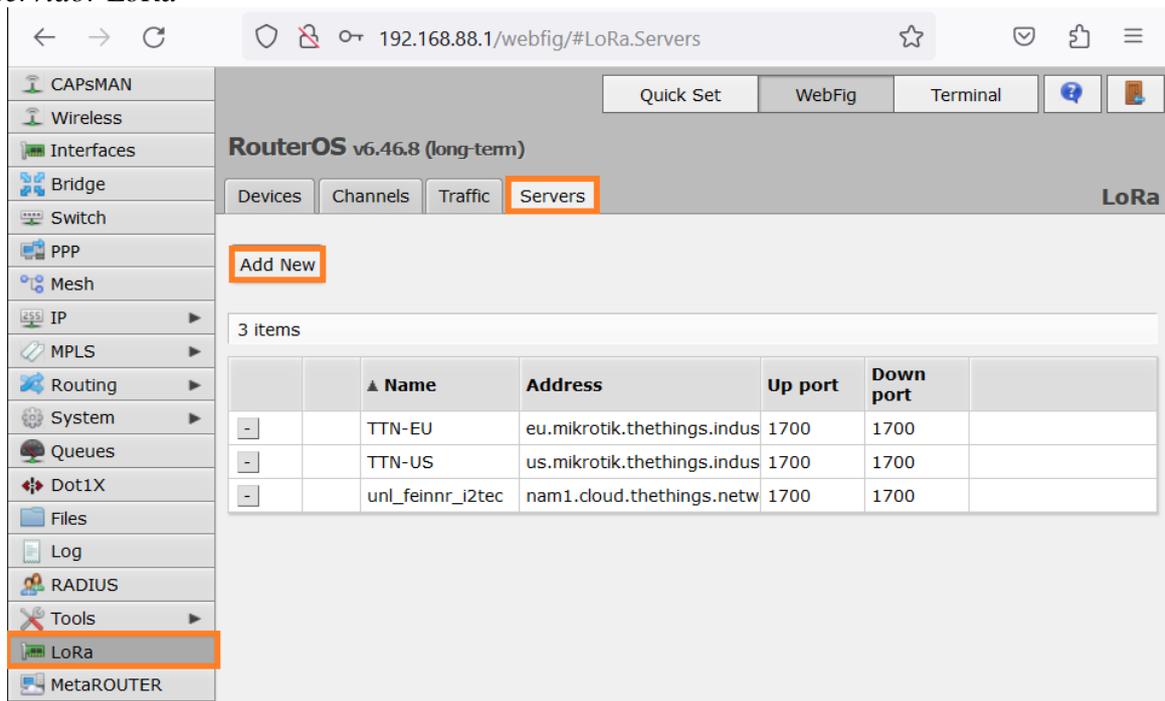


Nota: Se debe cambiar las credenciales de acceso por seguridad. Fuente: Autor

### **Paso 2. Creación de servidor**

- Desde la interfaz, seleccione la opción "LoRa" tal como se ilustra en la fig. 21. Acceda a la sección de servidores para crear uno nuevo.
- Elija la opción "Add New" para crear un servidor.

**Figura 21.**  
*Servidor LoRa*



Nota: Se debe contar con la dirección con el que se va comunicar el Gateway. Fuente: Autor

- Complete los parámetros requeridos:
  - Name:* se agrega el nombre del Gateway.
  - Address:* es la dirección de implementación proporcionada por TTN.
  - Up port:* puerto ascendente UDP del servidor de puerta de enlace.
  - Down port:* puerto descendente UDP del servidor de puerta de enlace.
- Una vez completado los parámetros, seleccionar la opción de aplicar y en Ok para guardar los cambios, como de observa en la fig. 22.

**Figura 22.**  
*Configuración de Servidor*

RouterOS v6.46.8 (long-term)

<unl\_feinnr\_i2tec>

OK Cancel Apply Remove

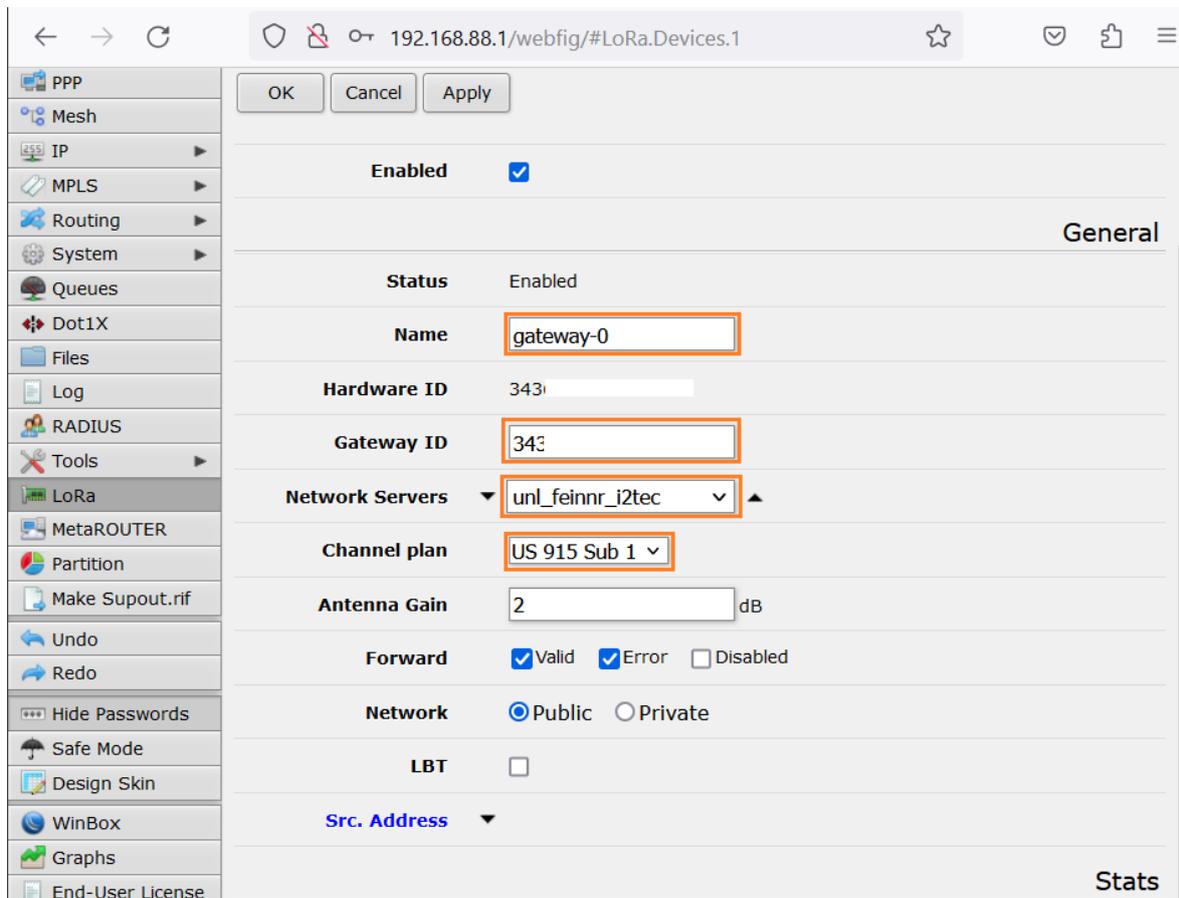
Name	unl_feinnr_i2tec
Address	nam1.cloud.thethings.net
Up port	1700
Down port	1700

Nota: De acuerdo a la región en la que nos encontremos TTN proporcionara la dirección.  
Fuente: Autor.

### **Paso 3. Conectividad del servidor con TTN**

- Seleccione un nuevo dispositivo y asigne un nombre al Gateway
- En la opción de Hardware ID es recomendable colocar el ID que viene en el datasheet del equipo wAP LR9.
- En Network Servers agregar el nombre del servidor que se creó en la figura 23.
- Seleccione el canal de frecuencia adecuado para su región, como por ejemplo "US 915 sub 1".
- Finalmente aplicar cambios en el dispositivo el cual debe estar en estado "Enable" para poder dar de alta el Gateway.

**Figura 23.**  
*Habilitación de servidor*



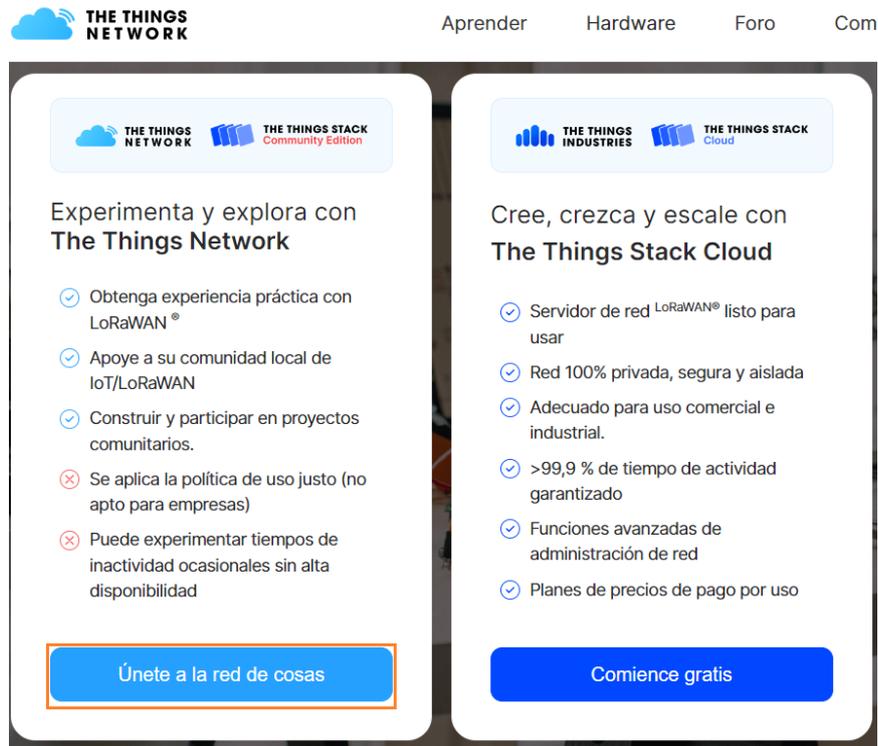
*Nota:* La interfaz muestra el ID del Equipo. Fuente: Autor

#### 5.4. Registro y Configuración del Gateway en The Things Networks

##### Paso 1. Creación de cuenta en The Things Stack community Edition

- Acceda al siguiente enlace: <https://www.thethingsnetwork.org/get-started/> aparecerá una ventana como se muestra en la fig. 24, y elija la opción The Things Stack Community Edition. Se le pedirá que se registre utilizando su dirección de correo electrónico.

**Figura 24.**  
*Suscripción a TTN*

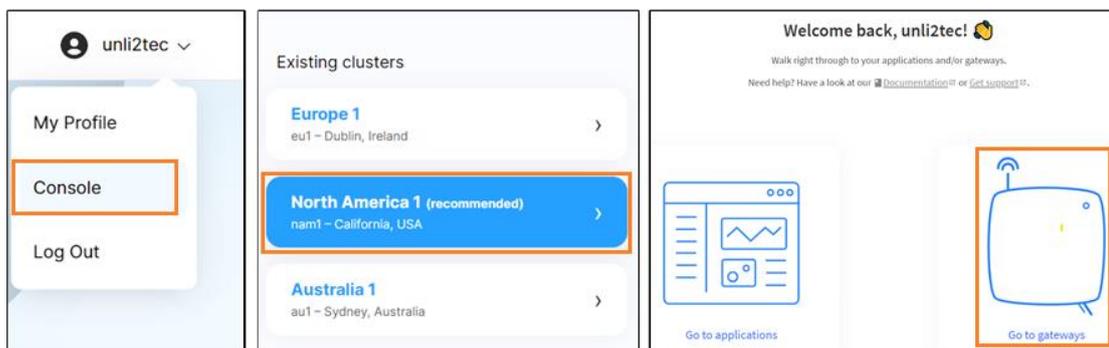


Nota: Elegimos la opción de community Edition Fuente: Autor

## Paso 2. Registro de GW MiKrotik

- Una vez creada la cuenta, inicie sesión en la consola utilizando su nombre de usuario, que se encuentra en la esquina superior derecha. Seleccionamos el “Clúster Picker en North América 1” que es la recomendada, según nuestra ubicación.
- Luego, haga clic en "Go to Gateway” como se observa en la fig. 25.

**Figura 25.**  
*Ingreso a la plataforma del Gateway*

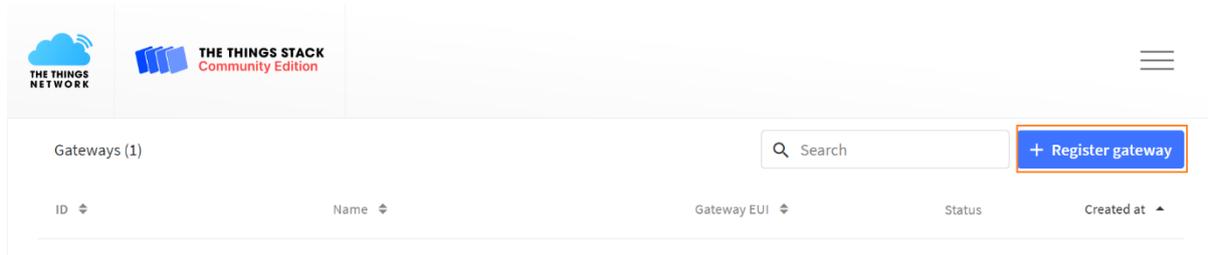


Nota: Pasos para ingresar a la ventana del Gateway. Fuente: Autor.

- En la interfaz que aparece, seleccione la opción "Registrar Gateway". Como se muestra en la fig. 26.

**Figura 26.**

Creación de un Nuevo GW



Fuente: Autor

- A continuación, se completan los siguientes parámetros, como se visualiza en la fig. 27.  
*Gateway EUI:* corresponde al ID del Equipo wAP LR9, debe ser el mismo que colocamos en la configuración de la interfaz de Mikrotik.

*Gateway ID:* se genera automáticamente después de haber ingresado el EUI.

*Gateway name:* se debe registrar con el mismo nombre con el que se registró en la interfaz de Mikrotik.

*Frequency plan:* seleccionamos el plan de frecuencia de acuerdo a nuestra ubicación (902-928Mhz para América).

**Figura 27.**

Registro del Gateway

## Register gateway

Register your gateway to enable data traffic between nearby end devices and the network.

Learn more in our guide on [Adding Gateways](#).

Gateway EUI ⓘ

34 3

Gateway ID ⓘ \*

eui-343

Gateway name ⓘ

un\_feinner\_j2tec

Frequency plan ⓘ \*

United States 902-928 MHz

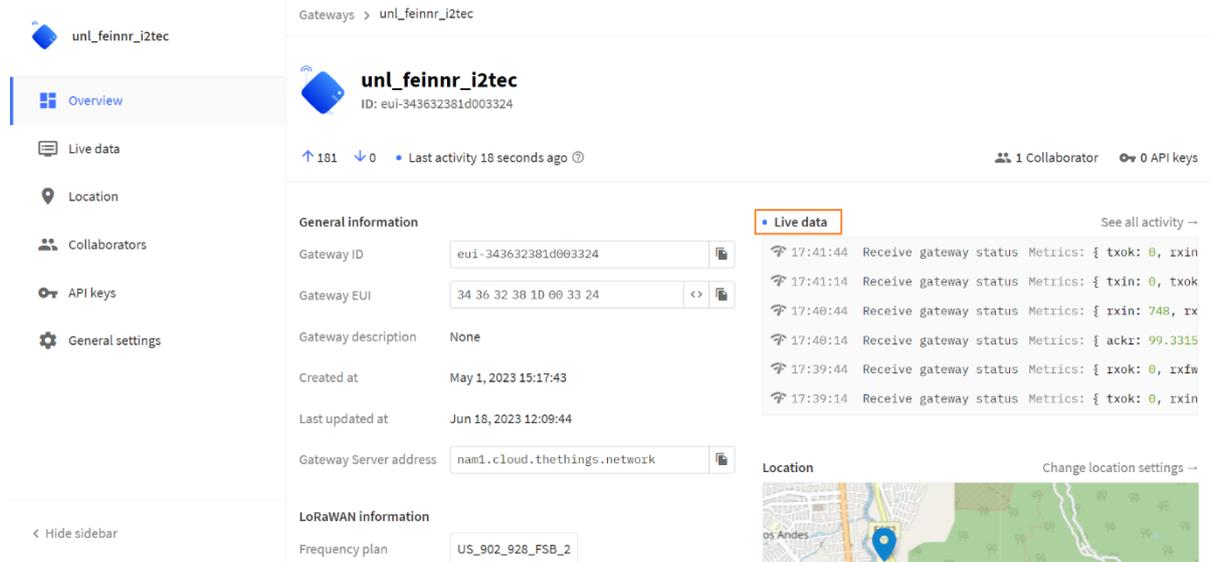
Nota: Los parámetros deben coincidir con los registrados en la interfaz de Mikrotik. Fuente: Autor.

### Paso 3. Revisión de conexión

- Una vez registrado el Gateway en la ventana aparecerá como estado activo y se podrá visualizar los datos en tiempo real de la comunicación del Gateway con la plataforma TTN como se ilustra en la fig. 28.
- Después de completar la configuración, ya es posible crear diferentes aplicaciones en las cuales se va a registrar el nodo final.

**Figura 28.**

*Datos de conexión en vivo*



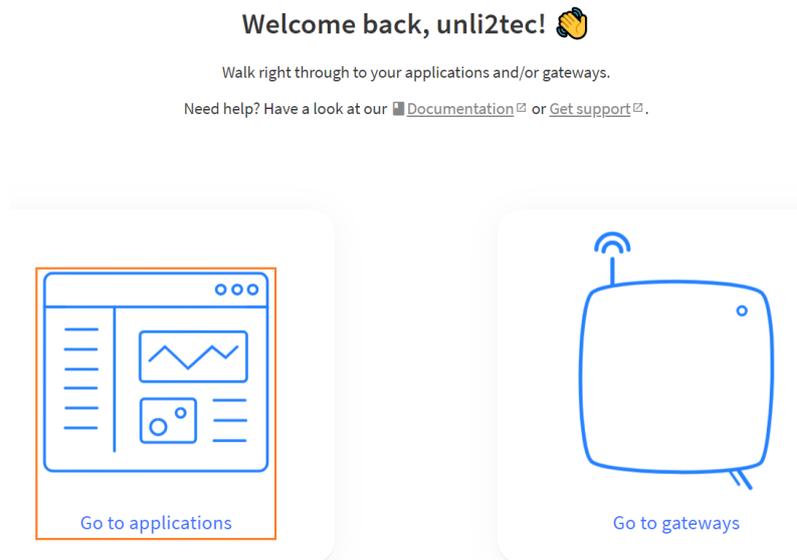
Nota: En la plataforma podemos configurar la ubicación del Gateway. Fuente: Autor.

## 5.5. Configuración de aplicación y dispositivo final en TTN

### Paso 1. Creación de una nueva aplicación

- Dentro de la interfaz seleccione la opción de “Go to applications” como se muestra en la fig. 29, Cree una nueva aplicación

**Figura 29.**  
*Ventana de Aplicación*



Nota: La creación de la aplicación permite configurar el dispositivo final. Fuente: Autor

- Complete los parámetros requeridos como se observa en la fig. 30.

*Aplicación ID:* agregue un código de identificación único para la aplicación.

*Aplicación name:* registre el nombre con el que desea identificar la aplicación.

*Descripción:* este campo es opcional y puede proporcionar una descripción breve del uso de la aplicación.

**Figura 30.**  
*Formulario para la creación de la aplicación*

### Create application

Within applications, you can register and manage end devices and their network data. After setting up your device fleet, use one of our many integration options to pass relevant data to your external services.

Learn more in our guide on [Adding Applications](#).

Application ID \*

puear-unl-esp32

Application name

puear unl esp32

Description

Esta aplicación permite la recolección de variables ambientales

Optional application description; can also be used to save notes about the application

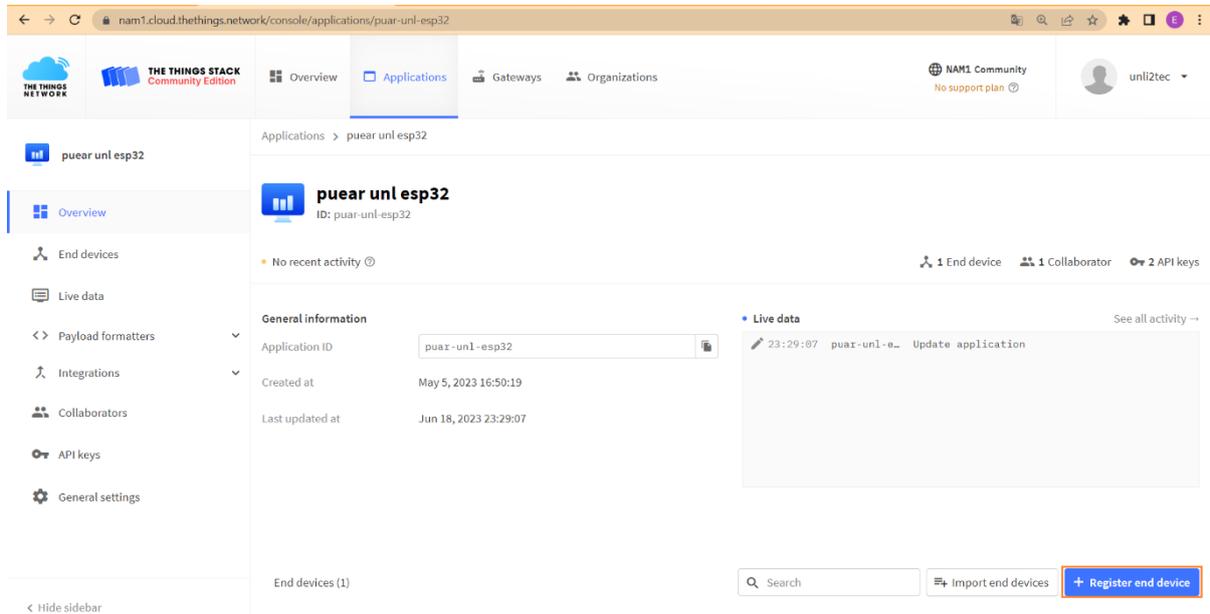
Create application

Fuente: Autor.

## Paso 2. Creación y configuración de dispositivo final Heltec Wifi LoRa 32

- Para agregar un dispositivo final seleccionamos “Add end device” como se muestra en la fig. 31.

**Figura 31.**  
*Interfaz de Aplicación*



Nota: Dentro de las aplicaciones, puede registrar y administrar dispositivos finales y sus datos de red. Fuente: Autor.

- TTN permite registrar los dispositivos de tres formas: a través de un código QR, de forma manual o utilizando el repositorio de dispositivos.
- Seleccione la opción de repositorio de dispositivos LoRaWAN como se ilustra en la fig. 32.

*End device Brand:* seleccione la marca del dispositivo HelTec.

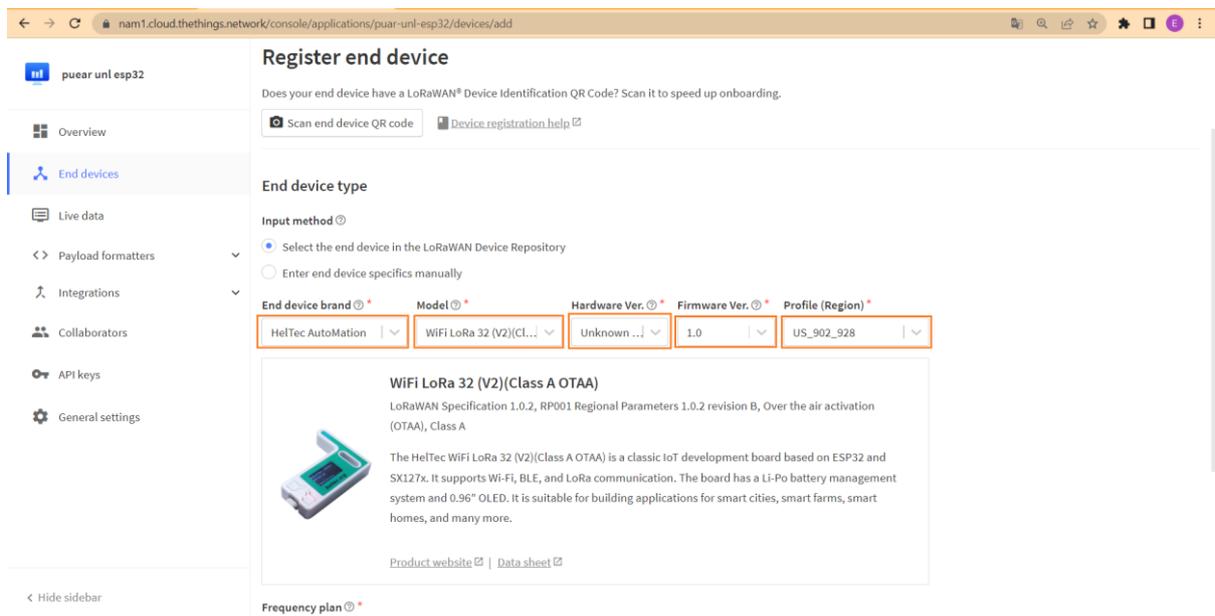
*Model:* elija el modelo del dispositivo final Wifi LoRa 32.

*Hardware Ver:* esta información es proporcionada por el fabricante del dispositivo y se encuentra en la hoja de especificaciones.

*Firmware Ver:* Es proporcionado por el fabricante.

*Profile (Region):* seleccione la frecuencia de trabajo del equipo según la región en la que se encuentra.

**Figura 32.**  
*Registro de dispositivo final*



Nota: El registro del dispositivo se lo realizo por medio del repositorio de TTN. Fuente:  
Autor

### **Paso 3. Información de aprovisionamiento**

- A continuación, se detalla la información de aprovisionamiento necesaria:

*Join EUI*: es un identificador único de 64 bits que se almacena en el dispositivo final antes de ejecutar el procedimiento de activación.

*DevEUI*: consta de 32 bits y se utiliza para identificar el dispositivo final dentro de la red actual. Es generado por la plataforma.

*AppKey*: es una clave raíz AES-128 específica para el dispositivo final. Se utiliza para cifrar y verificar la red de comunicación cuando un dispositivo final se une a una red a través de la activación por aire.

*End device ID*: se utiliza para identificar el dispositivo final en la plataforma TTN.

- En la fig. 33, se puede observar la correcta creación del dispositivo final.

**Figura 33.**  
*Configuración y creación del dispositivo final*

Frequency plan \*

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

---

**Provisioning information**

JoinEUI \*

C6 A6 0A 75 70 08 01 DE

This end device can be registered on the network

DevEUI \*

70 B3 D5 7E D0 05 EB CB 2/50 used

AppKey \*

37 B6 A4 B9 82 99 31 61 0D 7C 64 8F 0C 36 EA 47

End device ID \*

eui-70b3d57ed005ebcb

This value is automatically prefilled using the DevEUI

**After registration**

View registered end device

Register another end device of this type

Fuente: Autor.

**Figura 34.**  
*Dispositivo final creado correctamente*

**eui-e211**  
 ID: eui-e211

↑ 4 ↓ 4 • Last activity 50 seconds ago

[Overview](#) [Live data](#) [Messaging](#) [Location](#) [Payload formatters](#) [General settings](#)

---

**General information**

End device ID: eui-e211

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

LoRaWAN version: LoRaWAN Specification 1.0.2

Regional Parameters version: RP001 Regional Parameters 1.0.2 revision B

Created at: May 5, 2023 17:14:38

**Hardware**

Brand: heltec

Model: wifi-lora-32-class-a-otaa

Hardware version: \_unknown\_hw\_version\_

Firmware version: 1.0

**Live data** See all activity →

- 21:07:54 Update end device [ "locations" ]
- 21:03:19 Schedule data downlink for transmission on Gateway Server
- 21:03:19 Forward uplink data message DevAddr: 26 0C
- 21:03:19 Successfully processed data message DevAddr: 26 0C
- 20:58:20 Schedule data downlink for transmission on Gateway Server
- 20:58:20 Forward uplink data message DevAddr: 26 0C

**Location** Change location settings →

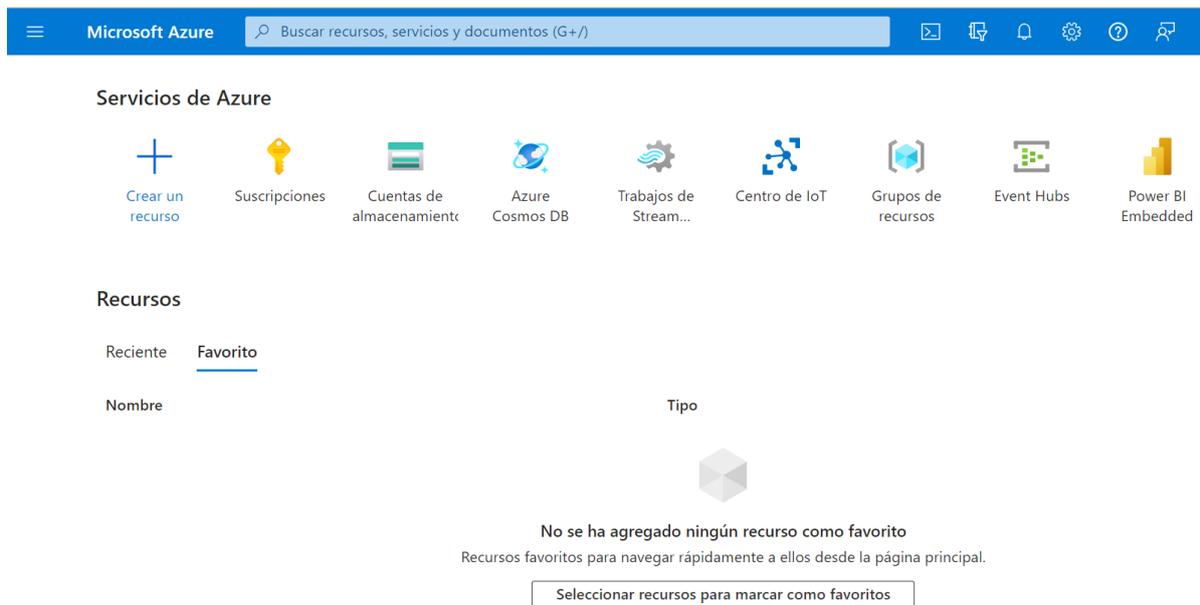
Nota: Una vez creada correctamente el dispositivo final se puede observar los datos en vivo.  
 Fuente: Autor

## 5.6. Integración de TTN con Azure IoT Hub

### Paso 1. Creación de cuenta en Azure IoT Hub

- Visite el sitio web de Microsoft Azure: <https://azure.microsoft.com/>. Desde allí, puede explorar los servicios y encontrar Azure IoT Hub.
- Si no tiene una cuenta de Azure, deberá crear una. Puede hacerlo haciendo clic en "Crear una cuenta gratuita" o "Comenzar gratis". Proporciona la información requerida como dirección de correo electrónico y detalles de facturación.
- Una vez que creada la cuenta, podrá acceder al portal de Azure como se ilustra en la fig. 35. Aquí es donde puede administrar los servicios y recursos en la nube.

**Figura 35.**  
*Inicio Azure IoT Hub*

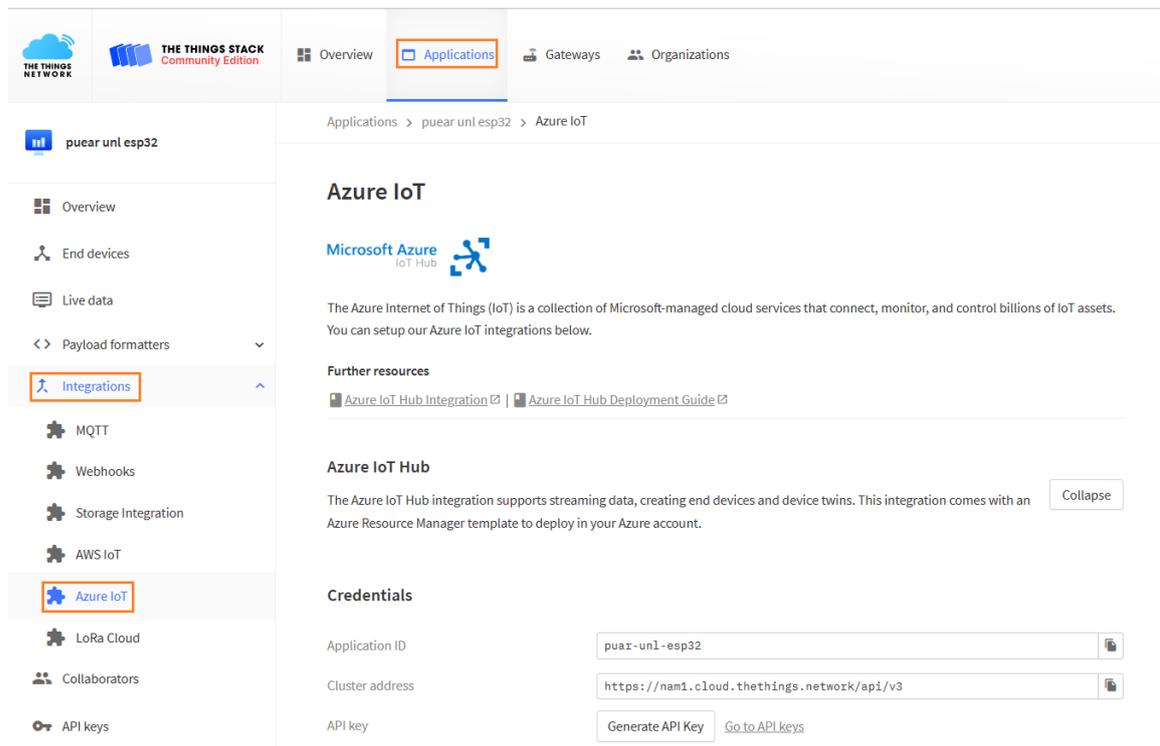


Nota: Página oficial de Azure IoT Hub. Fuente: Autor

### Paso 2. Integración de Azure Iot Hub con TTN

- Desde la sección de aplicaciones en TTN, seleccione la opción de integración "Azure IoT". y seleccionamos la opción "Expand" como se observa en la fig. 36.

**Figura 36.**  
*Credenciales de integración*



Nota: Al generar el API Key de lo debe guardar, ya que servirá para los siguientes pasos de integración. Fuente: Autor

### **Paso 3. Implementación de plantilla**

- TTN proporciona una guía de integración la cual detalla una plantilla de referencia, la guía de integración puede encontrarlo en el siguiente link: <https://www.thethingsindustries.com/docs/integrations/cloud-integrations/azure-iot-hub/deployment/>
- Para crear la plantilla se debe completar los siguientes parámetros de configuración:
  - Location:* Se refiere a la región en la cual se desplegarán los recursos de Azure. El valor por defecto es la región seleccionada para el grupo de recursos.
  - Name Suffix:* Es el sufijo que se agrega a los recursos de Azure. Por defecto, se utiliza una cadena aleatoria.
  - Stack Cluster Address:* Corresponde a la dirección de la implementación de The Things Stack. Puede obtenerse desde la página de integración de Azure IoT en The Things Stack.
  - Stack Application ID:* Es el identificador de pila. Puede obtenerse desde la página de integración de Azure IoT en The Things Stack

Stack API Key: Se trata de la clave API de la aplicación que se generó en el paso anterior, se debe copiar en este apartado.

*Enable fallback route:* Esta opción determina si se debe agregar la ruta de reserva de Azure IoT Hub. Si está habilitada, los mensajes de vínculo superior se enviarán al extremo de eventos predeterminado en su Azure IoT Hub.

- A continuación, debe hacer clic en “Review + Create” como se observa en la fig. 37.

**Figura 37.**  
*Plantilla personalizada*

**Implementación personalizada** ...

Implementar desde una plantilla personalizada

**Detalles del proyecto**

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \* ⓘ Azure subscription 1

Grupo de recursos \* ⓘ the-thing-network  
[Crear nuevo](#)

**Detalles de la instancia**

Región \* ⓘ (US) East US ✓

Location East Us ✓

Name Suffix puar-unl-esp32 ✓

Stack Cluster Address ⓘ https://nam1.cloud.thethings.network/api/v3 ✓

Stack Application ID \* ⓘ puar-unl-esp32 ✓

Stack API Key \* ⓘ ..... ✓

Event Hub Name Space SKU Standard ✓

IoT Hub SKU S1 ✓

IoT Hub Capacity 1 ✓

Storage Account SKU Standard\_LRS ✓

App Service Plan SKU Y1 ✓

App Service Plan Tier Dynamic ✓

Enable Fallback Route ⓘ true

[Revisar y crear](#) [< Anterior](#) [Siguiente: Revisar y crear >](#)

Nota: La región de implementación debe de ser la misma para los demás servicios que se crearan más adelante. Fuente: Autor.

#### **Paso 4. Configuración de integración nombre de host de Azure IoT Hub y clave de acceso de Azure IoT Hub.**

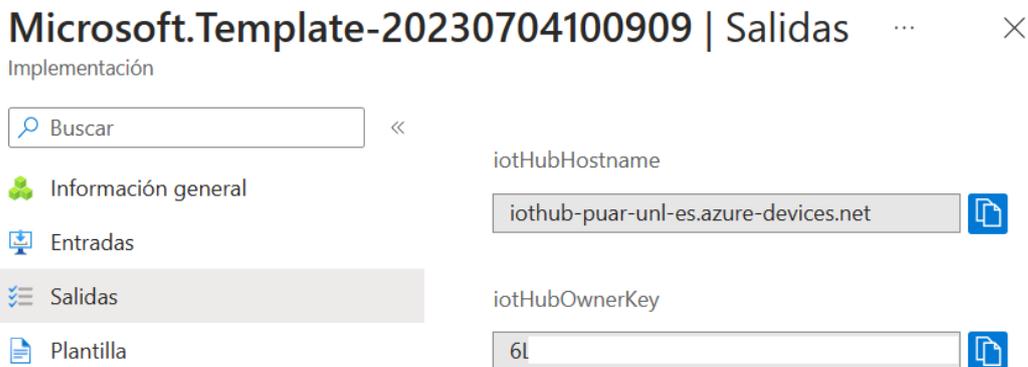
- Cuando se complete la implementación correctamente, seleccione la opción de salida. Se mostrará el hostname y el acces key como se observa en la fig. 37.

- Copie la dirección hostname y el acces key de Azure y aguegelo en la plataforma TTN como se ilustra en la fig. 38.

**Figura 38.**

*Salida de Azure IoT Hub*

Inicio > Microsoft.Template-20230704100909



Fuente: Autor.

**Paso 5. Habilitar la integración de Azure IoT Hub**

- En el aparatado de Azure IoT hostname agregamos las credenciales proporcionadas por Azure.
- En el apartado de zure IoT Hub Access key agregamos la clave de acceso key proporcionado por Azure

**Figura 39.**

*Habilitar la integración de Azure IoT Hub*

**Configuration**

**Azure IoT Hub hostname \***

iothub-puar-unl-es.azure-devices.net

Copy the Azure deployment "iotHubHostname" output

**Azure IoT Hub access key \***

.....

Copy the Azure deployment "iotHubOwnerKey" output

**Reported properties filter**

- |   |   |  |
|---|---|--|
| <input checked="" type="checkbox"/> Uplink message  | <input checked="" type="checkbox"/> Normalized uplink | <input checked="" type="checkbox"/> Join accept                |
| <input checked="" type="checkbox"/> Downlink ack    | <input checked="" type="checkbox"/> Downlink nack     | <input checked="" type="checkbox"/> Downlink sent              |
| <input checked="" type="checkbox"/> Downlink failed | <input checked="" type="checkbox"/> Downlink queued   | <input checked="" type="checkbox"/> Downlink queue invalidated |
| <input checked="" type="checkbox"/> Location solved | <input checked="" type="checkbox"/> Service data      |  |

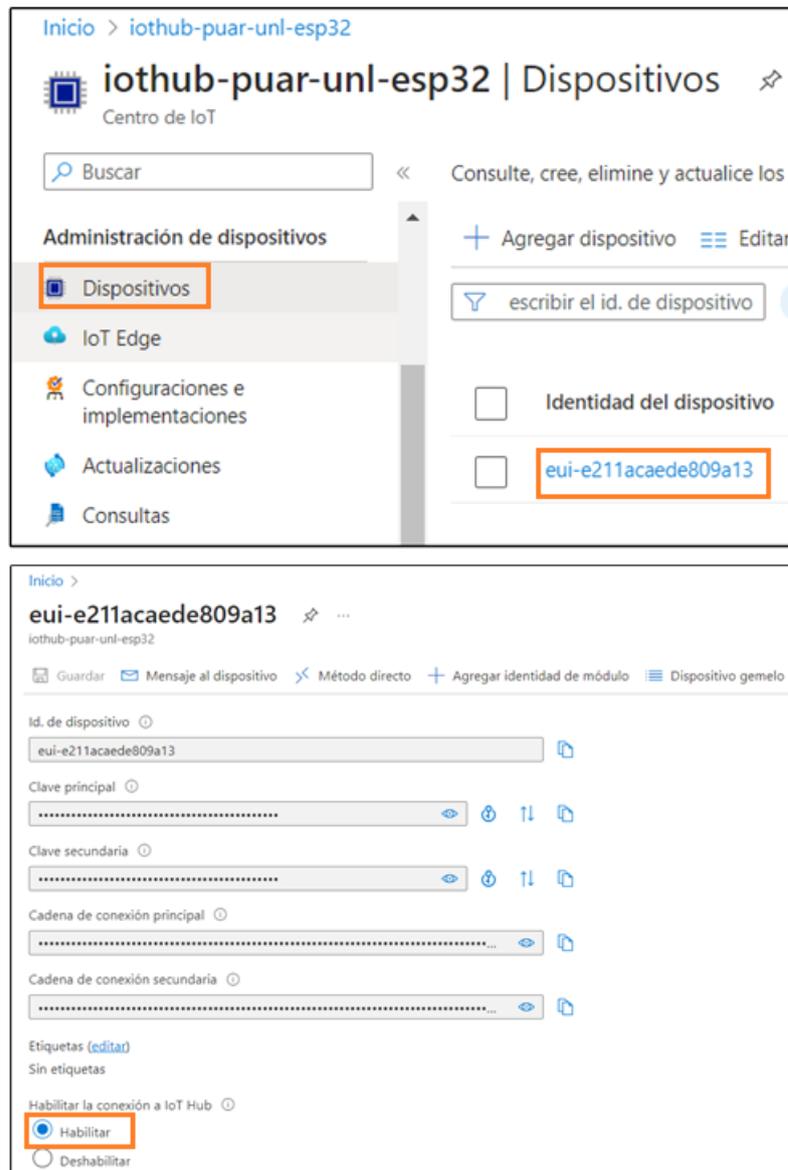
Fuente: Autor

## Paso 6. Habilitar dispositivo de IoT Hub

- Para verificar que la conexión se realizó correctamente, ingrese en dispositivos en el recurso de IoT Hub y verifique que el id del dispositivo es el mismo que el dispositivo registrado en TTN como se muestra en la fig. 40.
- Debe permanecer habilitado la conexión con IoT Hub.

**Figura 40.**

*Dispositivo vinculado*



Nota: En el apartado de dispositivo gemelo se puede observar los datos transmitidos desde la aplicación TTN. Fuente: Autor

## 5.7. Instalación/Implementación de servicios de App functions y cosmos DB

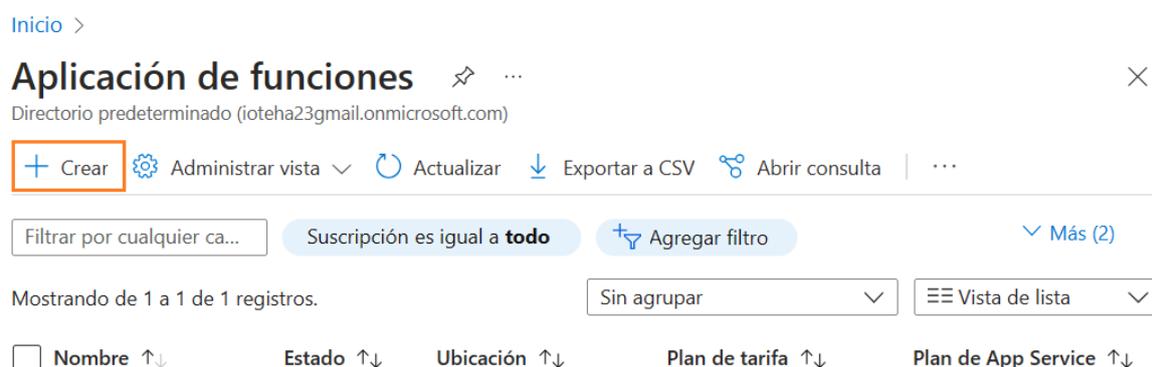
### 5.7.1. Azure Functions

#### Paso 1. Instalación del servicio App functions

- En el apartado de búsqueda escriba “Aplicación de funciones” e ingrese a la misma.
- Una vez dentro del directorio seleccione la opción “Crear” como se aprecia en la fig. 41.

**Figura 41.**

*Servicio de App Functions*



Fuente: Autor

- A continuación, se debe completar los parámetros solicitados como se visualiza en la fig. 42.

*Suscripción:* es el tipo de suscripción con el que se registró a Azure IoT Hub

*Grupo de recursos:* es el nombre que asignamos al grupo utilizado para agrupar todos los recursos que necesarios.

*Nombre de la aplicación de funciones:* es el nombre identificativo del servicio.

*Región:* se debe seleccionar la misma región con la que se agregaron los primeros servicios.

*Pila del entorno en tiempo de ejecución:* es una estructura de datos utilizada por el entorno de ejecución de Node.js para realizar un seguimiento de la ejecución de funciones y permite ejecutar código JavaScript.

*Versión:* representa una versión LTS de Node.js con el número de versión "18". Esto indicaría que es una versión que se mantendrá durante un período extendido y es adecuada para aplicaciones y proyectos que requieren estabilidad a largo plazo.

*Región:* es recomendado usar la una misma región para implementación de todos los servicios.

- Una vez completado lo solicitado seleccione “Revisar y Crear”.

**Figura 42.**  
*Creación de cuenta App functions*

Inicio > Aplicación de funciones >

## Crear aplicación de funciones

Suscripción \* ⓘ Azure subscription 1

Grupo de recursos \* ⓘ the-thing-network  
[Crear nuevo](#)

### Detalles de instancia

Nombre de la aplicación de funciones \* fn-cosmosdb ✓  
.azurewebsites.net

¿Desea implementar un código o imagen de contenedor? \*  Código  Imagen de contenedor

Pila del entorno en tiempo de ejecución \* Node.js

Versión \* 18 LTS

Región \* East US

### Sistema operativo

Le hemos recomendado el sistema operativo según la selección que ha hecho de la pila en tiempo de ejecución.

Sistema operativo \*  Linux  Windows

[Revisar y crear](#) [< Anterior](#) [Siguiendo: Storage >](#)

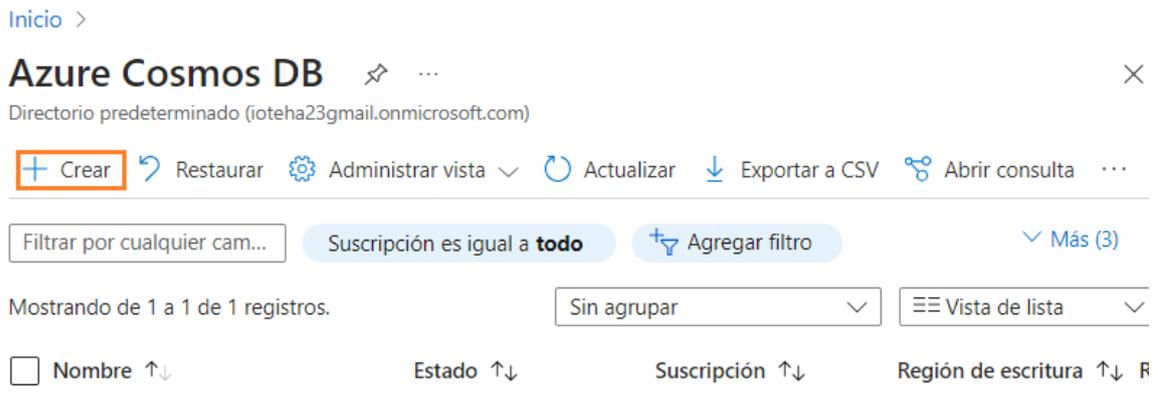
Fuente: Autor

### 5.6.2. Cosmos DB

#### Paso 1. Instalación del servicio de base de datos Cosmos DB

- En el apartado de búsqueda escriba “Azure Cosmos DB” y seleccione.
- Una vez dentro del directorio seleccione la opción “Crear” como se aprecia en la fig. 43.

**Figura 43.**  
*Servicio de Cosmos DB*



Fuente: Autor

- En el apartado de Selección de la opción de API Seleccionar la opción “Nucleo (SQL)”
- A continuación, se debe completar los parámetros solicitados:
  - Suscripción:* es el tipo de suscripción con el que se registró a Azure IoT Hub
  - Grupo de recursos:* es el nombre que asignamos al grupo utilizado para agrupar todos los recursos que necesarios.
  - Nombre:* es el nombre identificativo del servicio.
  - Región:* se debe seleccionar la misma región con la que se agregaron los primeros servicios.
  - Modo de capacidad:* el modo de rendimiento aprovisionado permite usar la capacidad de almacenamiento gratuito de 25 GB. De igual forma como el resto de servicios se puede escalar y emplear el almacenamiento.
- Una vez completado lo solicitado seleccione “Revisar y Crear”.

**Figura 44.**  
*Creación de cuenta Azure Cosmos DB*

Inicio > Azure Cosmos DB > Selección de la opción de API >

## Creación de una cuenta de Azure Cosmos DB - Core (SQL) ... ×

**Detalles del proyecto**

Seleccione la suscripción para administrar los recursos y costos implementados. Use grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \*  ▾

Grupo de recursos \*  ▾  
[Crear nuevo](#)

**Detalles de la instancia**

Nombre de cuenta \*  ✓

Ubicación \*  ▾

Modo de capacidad ⓘ  Rendimiento aprovisionado  Serverless  
[Obtenga más información sobre el modo de capacidad.](#)

With Azure Cosmos DB free tier, you will get the first 1000 RU/s and 25 GB of storage for free in an account. You can enable free tier on up to one account per subscription. Estimated \$64/month discount per account.

La suscripción que ha seleccionado ya tiene una cuenta con el nivel gratis habilitado.

Aplicar descuento del nivel Gratis  Aplicar  No aplicar

Limitar el rendimiento total de la cuenta  Limitar la cantidad total de rendimiento que se puede aprovisionar en esta cuenta

**i** Este límite evitará cargos inesperados relacionados con el rendimiento aprovisionado. Puede actualizar o quitar este límite después de crear la cuenta.

---

[Revisar y crear](#) [Anterior](#) [Siguiente: Distribución global](#)

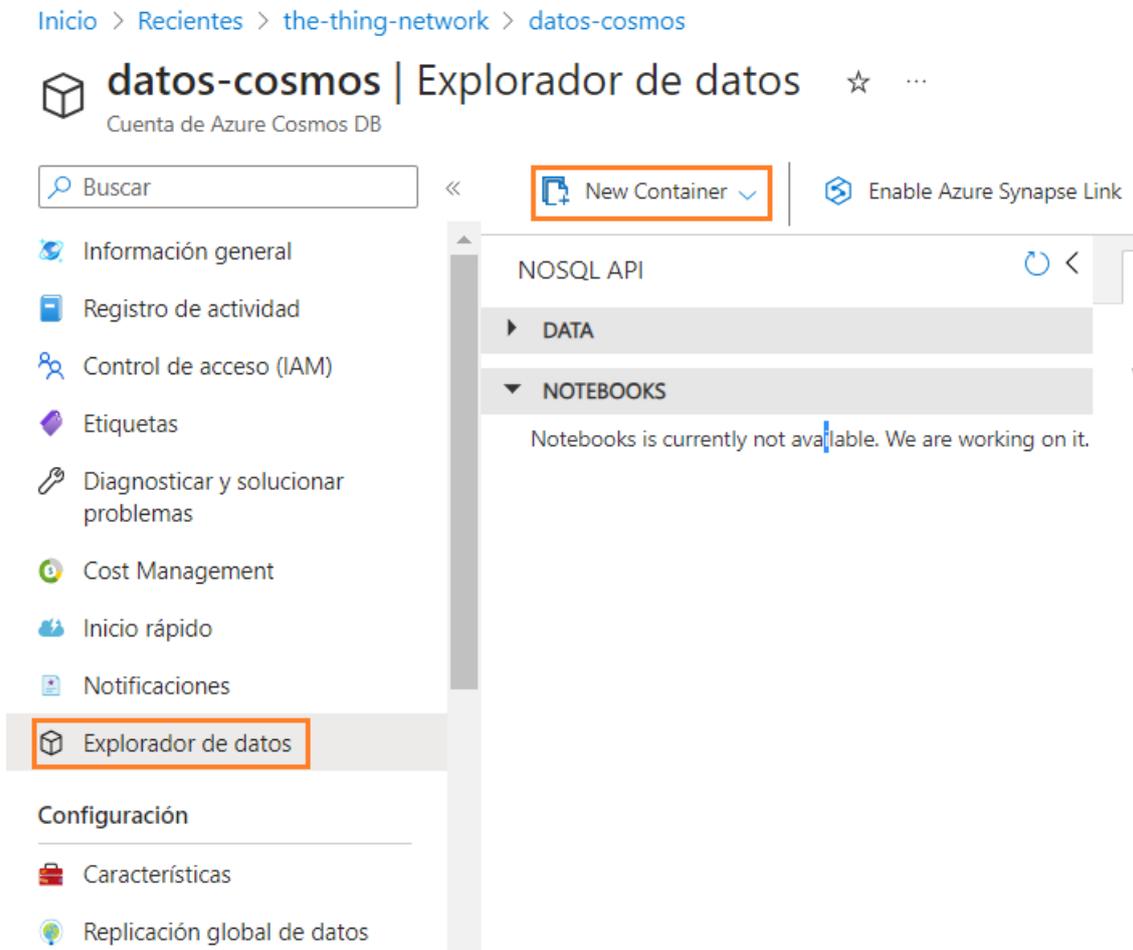
Nota: Se puede modificar el límite de rendimiento total, después de crear la cuenta. Fuente: Autor

## 5.8. Integración de base de datos de Cosmos DB, App Functions e IoT Hub

### Paso 1. Configuración de base de datos ID y contenedores.

- Para configurar este servicio se debe ingresar al grupo de recursos creado “the-thing-network” y seleccione el servicio “datos-cosmos” después debe seleccionar la opción “Explorador de Datos”.
- A continuación, seleccione la opción “New Container” como se visualiza en la fig. 45.

**Figura 45.**  
*Creación de contenedores Cosmos DB*



Fuente: Autor.

- A continuación, aparecerá una ventana como se observa en la fig. 46, con parámetros que se deben completar para la creación de la base de datos y los contenedores contenedor.
- Una vez que se haya completado la creación de la base de datos, es fundamental copiar la cadena de conexión principal. Esta información se localiza en el apartado de "claves" de Cosmos DB y será esencial para conectar de manera efectiva el servicio de App Functions con la base de datos.

**Figura 46.**  
*Parámetros de contenedor*

The image displays two side-by-side screenshots of the Azure Cosmos DB configuration interface. The left screenshot shows the 'Database id' section with 'PUEAR' entered in the text field, 'Create new' selected, and 'Share throughput across containers' checked. Below this, 'Database throughput (autoscale)' is set to 'Autoscale', and 'Database Max RU/s' is set to 200. The 'Container id' is 'EHA'. The right screenshot shows the 'Database id' section with 'PUEAR' selected from a dropdown menu, 'Use existing' selected, and 'Container id' as 'EMA'. 'Indexing' is set to 'Automatic', and the 'Partition key' is '/id'.

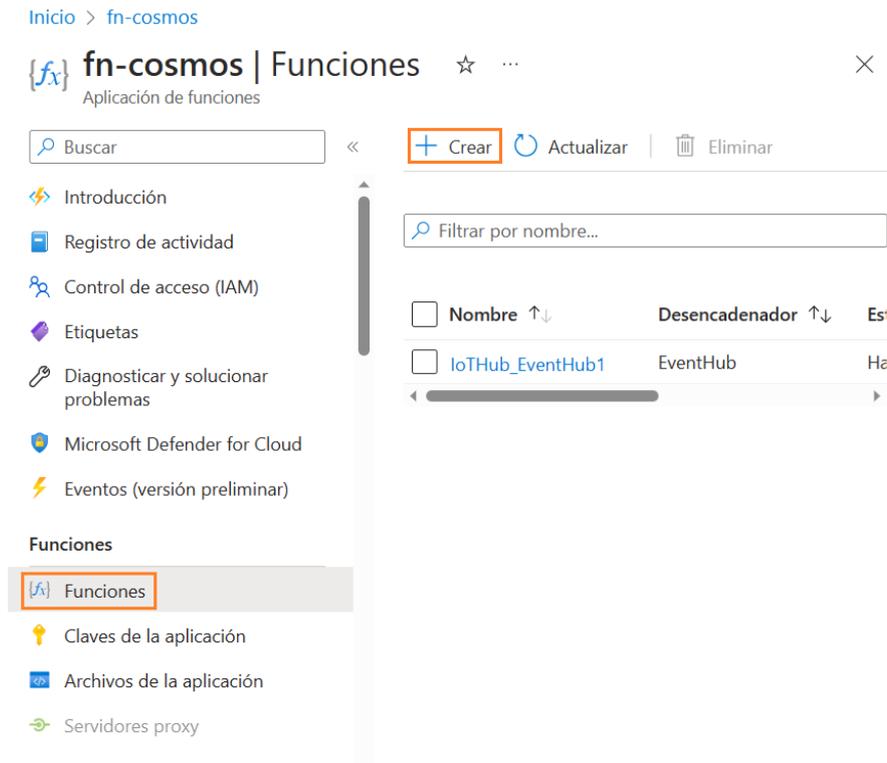
Nota: Se creó una base de datos denominado PUEAR y dos contenedores EHA y EMA.  
Fuente: Autor.

## **Paso 2. Configuración de App Función**

- Para configurar este servicio, acceda al grupo de recursos creado previamente con el nombre "the-thing-network" y seleccione el servicio "fn-cosmos". Luego, haga clic en la opción "Funciones" y seleccione "Crear" como se aprecia en la fig. 47.

**Figura 47.**

*Creación de una nueva función*

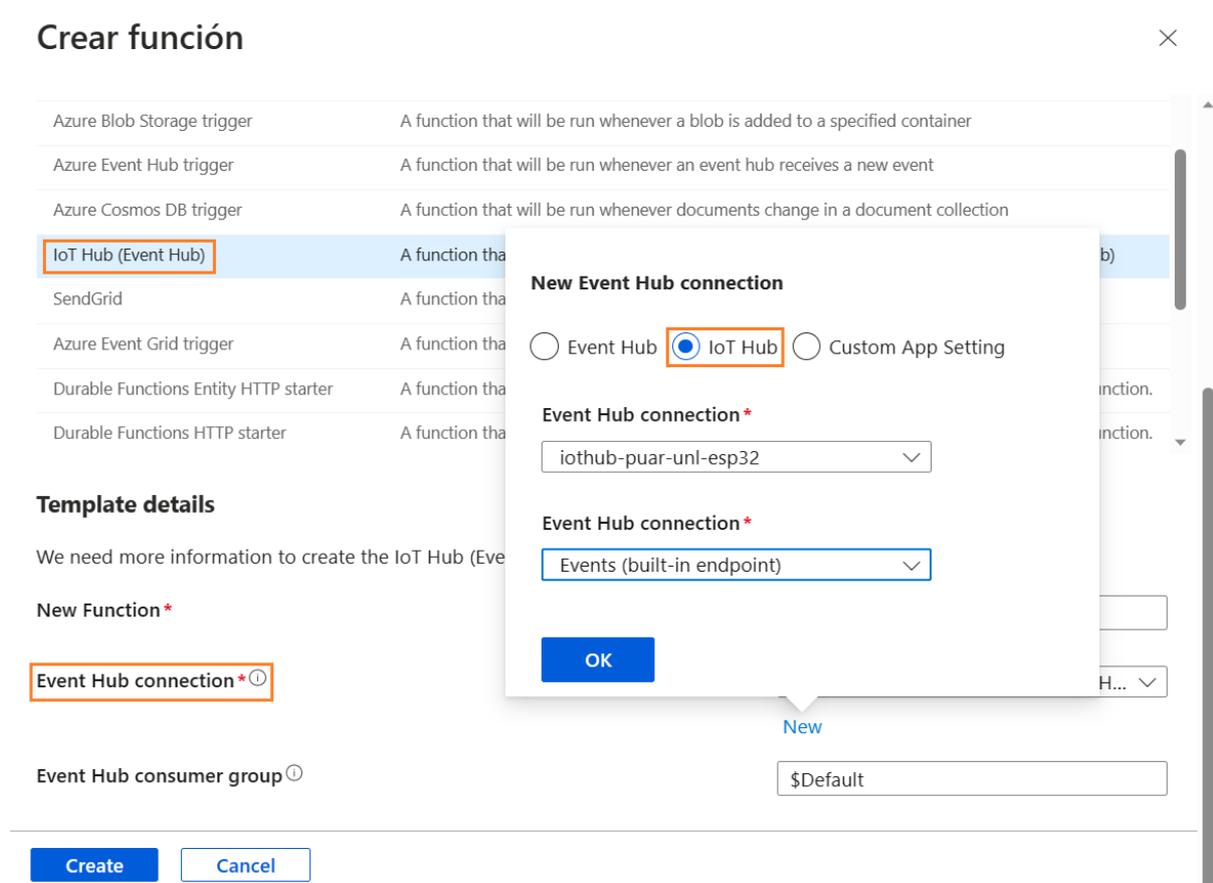


Fuente: Autor.

- A continuación, se debe seleccionar el entorno de desarrollo, en este caso desde el portal de Azure.
- Seleccione el desencadenador "IoT Hub (Event Hub)" para establecer la conexión entre la función y el centro de IoT.
- Una vez seleccionado el desencadenador, configure la conexión. Para ello, elija "New Event Hub connection" y proporcione el nombre del servicio de IoT Hub previamente creado. Asegúrese de seleccionar la opción "Event (built-in endpoint)", tal como se muestra en la fig. 46.

**Figura 48.**

*Conexión de la función con IoT Hub*

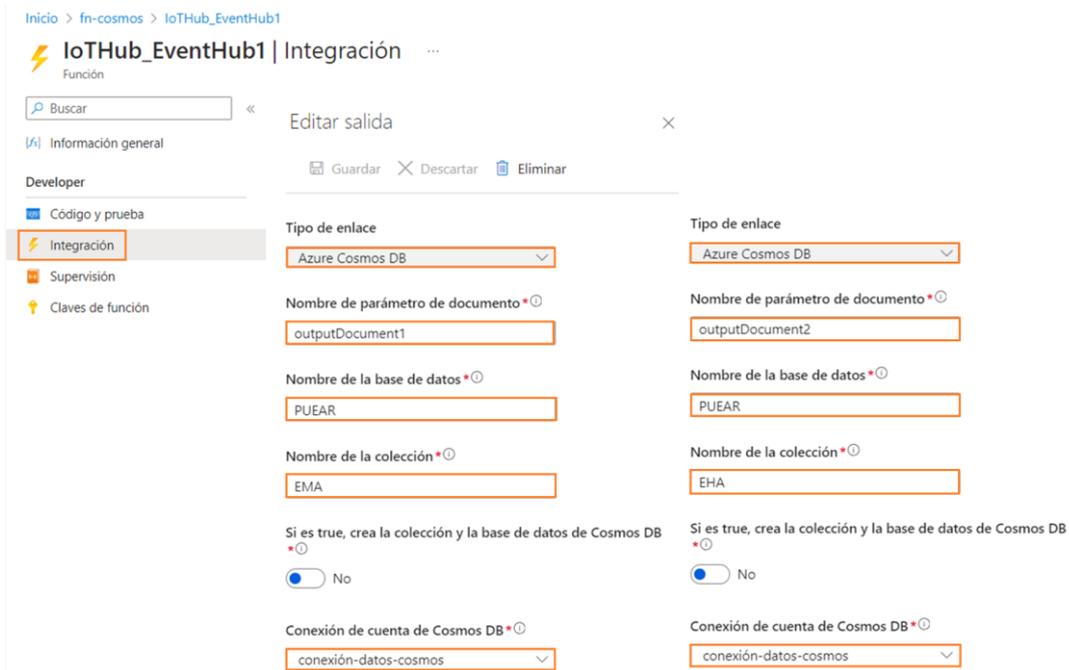


Fuente: Autor.

- Una vez creada la función, procedemos a configurar la salida de conexión con Cosmos DB. Para ello, accedemos al apartado de "Integración" y seleccionamos la opción "Outputs" como se observa en la fig. 49.
- Seleccionamos la opción de "Azure Cosmos DB" para establecer la conexión con la base de datos.
- Luego, ingresamos la cuenta de conexión previamente obtenida de Cosmos DB y la agregamos en la App Funciones.
- Asignamos un nombre al parámetro del documento.
- Por último, agregamos el nombre de la base de datos y el contenedor creados previamente.

**Figura 49.**

*Salida a la base de datos*



*Nota:* Para poder conectar la función con la base de datos fue necesario establecer una conexión previamente en App Functions con el nombre conexión-datos-cosmos. Fuente: Autor.

En la sección de "Código y Prueba", como se aprecia en la fig. 50, se realizó modificaciones en el archivo "index.js" con el objetivo es extraer los datos provenientes de los sensores y almacenarlos en Cosmos DB.

**Figura 50.**

*Código y prueba de consultas*



*Nota:* La función permite que se guarden los datos de cada nodo en un contenedor diferente. Fuente: Autor.

## 5.9. Simulación en OMNeT ++6

OMNeT++ es un entorno de simulación de código abierto y una plataforma de desarrollo utilizada principalmente para la simulación de redes de comunicación, sistemas distribuidos y otros sistemas complejos. OMNeT++ proporciona a los investigadores, ingenieros y desarrolladores una herramienta poderosa para diseñar, modelar y analizar sistemas complejos en un entorno de simulación virtual. Es ampliamente utilizado en diversos campos, incluyendo telecomunicaciones, redes de computadoras, Internet de las cosas (IoT), sistemas inalámbricos, protocolos de red, sistemas de transporte y muchos otros dominios.

Para el desarrollo de las simulaciones basadas en redes LoRa se utilizó FloRa (*Framework for LoRa*). FloRa es un software que permite realizar simulaciones de extremo a extremo de redes de largo alcance como lo es LoRa (FloRa, 2022).

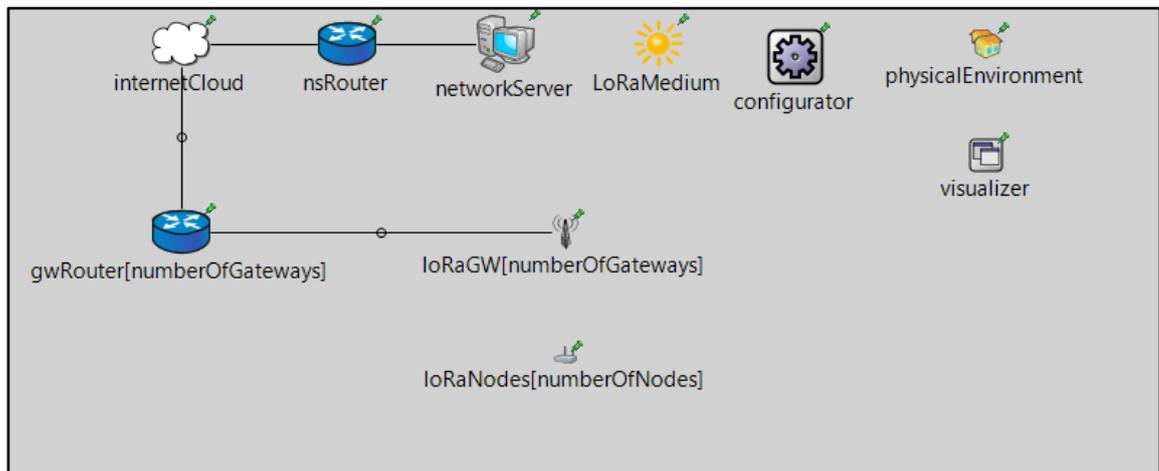
Este software permite realizar las siguientes simulaciones:

- Modelo preciso de la capa física de LoRa (incluidas las colisiones y el efecto de captura).
- Simulaciones con una (o más) puertas de enlace en la red.
- Simulaciones de extremo a extremo, incluido el modelado preciso de la red.
- Estadísticas de consumo de energía en red. (FloRa, 2022).

Las simulaciones que serán establecidos comparten una topología idéntica, como se aprecia en la fig. 51, pero se introducirán cambios en ciertos parámetros de comunicación y componentes. Posteriormente, se presentará una representación gráfica del archivo NED que describe la estructura de la red empleada, y el código asociado se encuentra disponible en el anexo adjunto.

En esta simulación realizada en OMNeT++, se ha evaluado el comportamiento de la tasa de extracción de datos o DER (Data Extraction Rate). con diferentes cantidades de nodos y diferentes valores del factor de esparcimiento (SF). Un DER del 100% significaría que todos los paquetes enviados se entregaron con éxito, mientras que un DER menor indicaría que algunos paquetes se perdieron o no fueron entregados correctamente. Como complemento a esta evaluación, también se analizaron otros parámetros cruciales, como el consumo energético de los nodos y la incidencia de colisiones durante la transmisión de datos. Estos elementos adicionales ayudan a proporcionar una imagen más completa del comportamiento de la red bajo diferentes circunstancias.

**Figura 51.**  
*Topología de red LoRa*



*Nota:* esta simulación se lo realizo con los parámetros mencionados en la tabla 4. Fuente: Autor

Las simulaciones se las realizara tomando en cuenta las características técnicas de los dispersivos como el nodo final, heltec Wifi LoRa 32 y el Gateway wAP LR9. A continuación, en la tabla 8 se muestran los parámetros generales de configuración.

**Tabla 8.**  
*Parámetros Generales para la simulación FloRA*

Parámetro	Valor
Numero de nodos	50,100,150,200,250
Ancho de banda	125Khz
Core Rate	4
Potencia máxima de transmisión (TP)	2dBm±19dB
Puerta local GW	2000
Puerta de destino GW	1000
Puerto local NetworkServer	2000
Puerto de Destino NetworkServer	1000
Periodo total de simulación de la red	1 d
Sigma para zonas urbanas	3.57
Factor de dispersión (SF)	7,8,9,10,11,12
Resolución de simulación	-11

Fuente: Autor

### 5.9.1. Definición de parámetros de la red

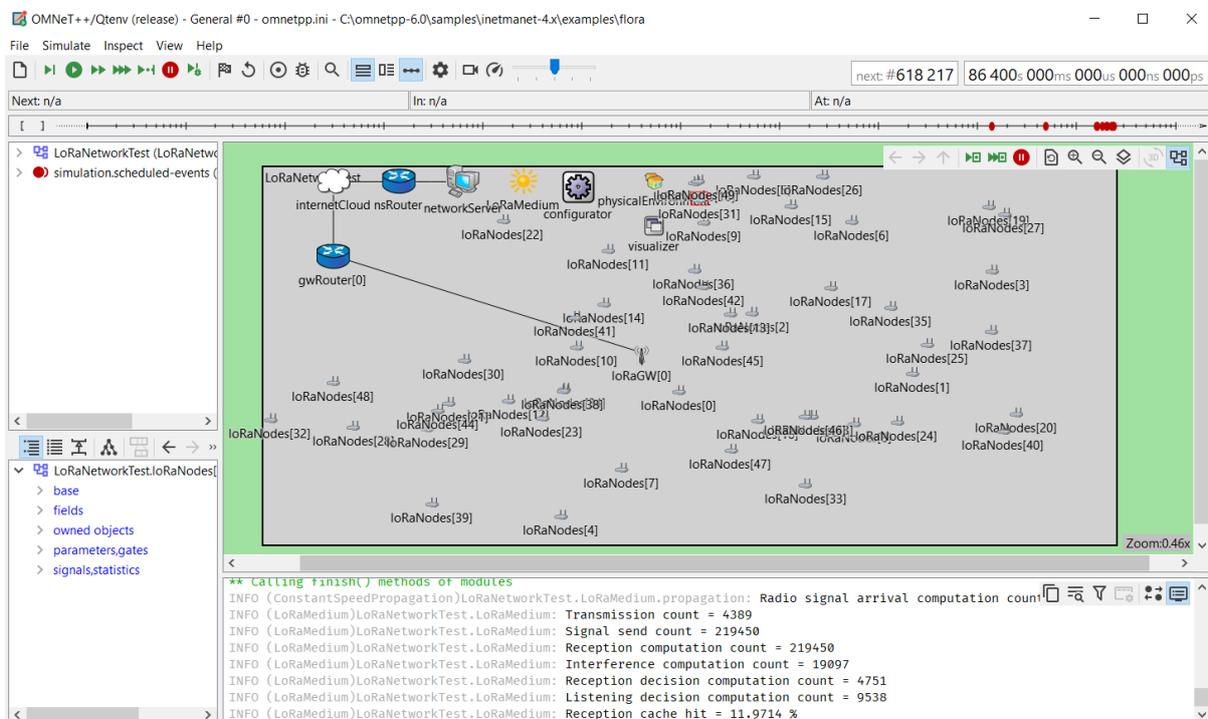
El período total de simulación de la red es de 1 días. Durante la simulación, los nodos transmitirán su primer paquete en los primeros 1000 segundos y, a partir de ese momento, transmitirán paquetes de forma aleatoria y variara de acuerdo con la distribución con una media de 1000 segundos, Además, se utiliza un parámetro llamado "sigma" para calcular la pérdida media de la señal, teniendo en cuenta la presencia de edificios altos en entornos urbanos y bajos en entornos suburbanos. Los valores de sigma son 3.57 para entornos urbanos y 7.0 para entornos rurales.

Se realizarán simulaciones para diferentes cantidades de nodos desplegados, desde 50 hasta 250 nodos con incrementos de 50, lo que resultará en un total de 4 simulaciones para cada modelo de red.

### 5.9.2. Resultados de simulación

Para realizar la simulación, se han configurado diferentes escenarios con diferentes cantidades de nodos (50, 100, 150, 200 y 250) y se han utilizado diferentes valores de SF (7, 8, 9, 10, 11 y 12) para cada nodo en la red. En la fig. 52, se puede apreciar el entorno de simulación para 100 nodos.

**Figura 52.**  
*Simulación con 100 nodos*



*Nota:* Entorno de simulación Qt con 100 nodos, con un área de 2000m x 1000m. Fuente: Autor

Cada combinación de nodos y SF se ha simulado para evaluar el rendimiento de la red. En la tabla 9 se muestran los resultados de la simulación.

**Tabla 9.**  
*Resultados tasa de extracción de datos (DER)*

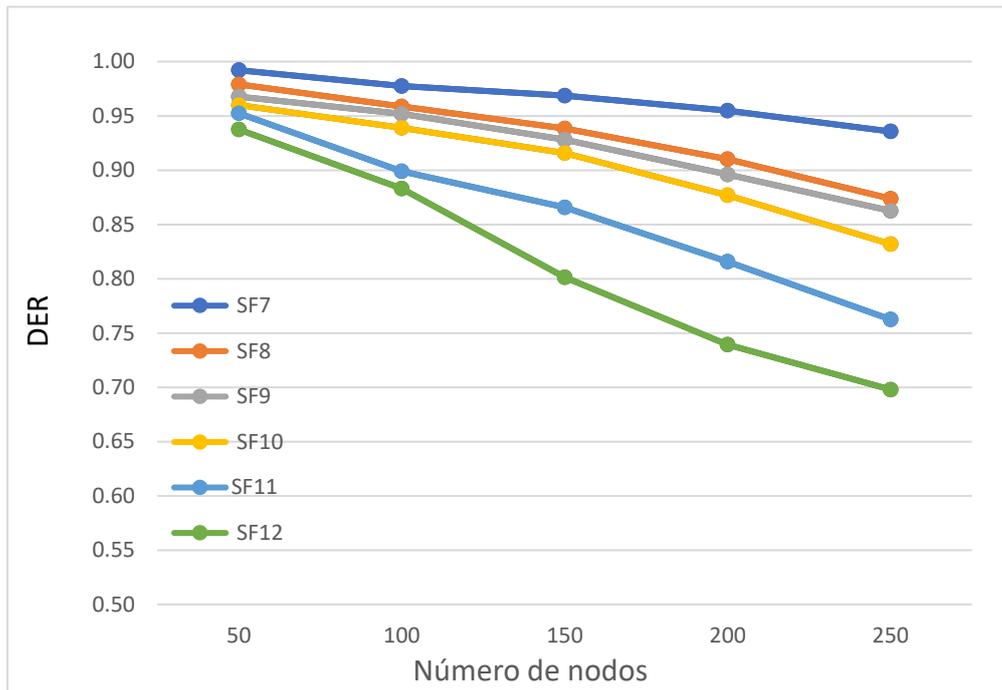
Número de Nodos	DER SF7	DER SF8	DER SF9	DER SF10	DER SF11	DER SF12
50	0.99%	0.98%	0.97%	0.96%	0.95%	0.94%
100	0.98%	0.96%	0.95%	0.94%	0.90%	0.88%
150	0.97%	0.94%	0.93%	0.92%	0.87%	0.80%
200	0.95%	0.91%	0.90%	0.88%	0.82%	0.74%
250	0.94%	0.87%	0.86%	0.83%	0.76%	0.70%

Fuente: Autor

Al analizar los resultados, se observa que el DER varía significativamente con la cantidad de nodos y el valor del SF. A medida que aumenta el número de nodos y se utilizan un SF de 12 el DER tiende a disminuir significativamente. Esto sugiere que la red puede experimentar mayores colisiones y congestiones, lo que resulta en una menor tasa de entrega de paquetes.

Por el contrario, cuando se utilizan SF de 7 y hay menos nodos, el DER tiende a mejorar, lo que indica una mejor calidad de la transmisión, en la fig. 52, se puede apreciar de mejor manera el comportamiento de la red con sus diferentes nodos.

**Figura 53.**  
*Comportamiento del DER*



*Nota:* DER dependiendo del tamaño de la red. Fuente: Autor.

En relación al consumo de energía de los nodos, la tabla siguiente presenta los valores promedio de consumo energético, y el número de colisiones de los nodos. Estos valores reflejan el consumo a lo largo de la simulación de 1 día.

**Tabla 10.**  
*Consumo de energía promedio y colisiones*

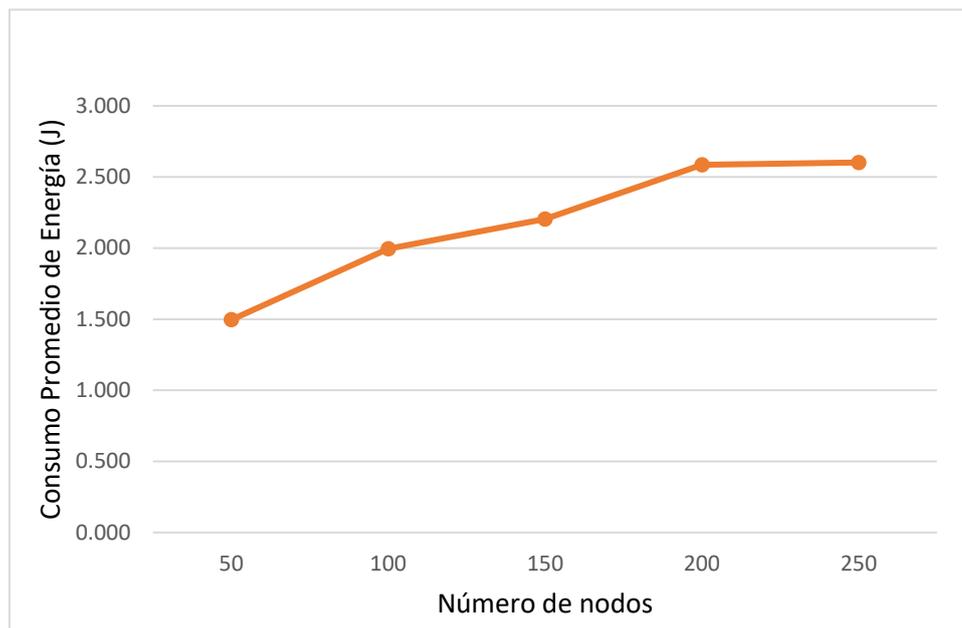
Número de nodos	Consumo promedio de energía (J)	Colisiones
50	1.50	70
100	2.00	287
150	2.20	650
200	2.58	1283
250	2.60	1972

Fuente: Autor

En la fig.54 se observa una tendencia clara de aumento en el consumo promedio de energía a medida que aumenta el número de nodos en la red. Esto sugiere que la cantidad de nodos en la red está directamente relacionada con el consumo de energía global. A medida que la red se vuelve más densa, la energía consumida por los nodos también tiende a aumentar.

**Figura 54.**

*Consumo promedio de energía*

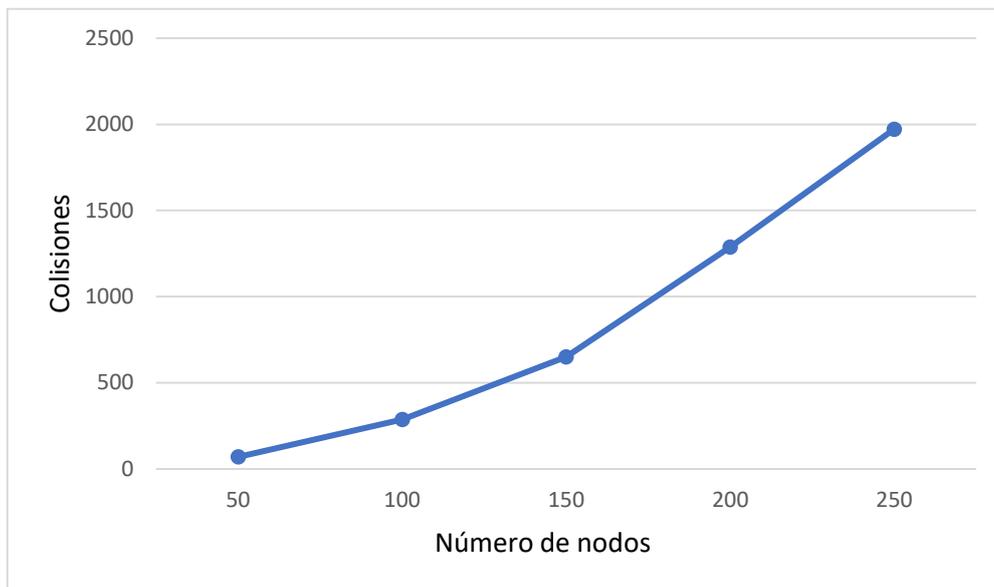


Fuente: Autor

De acuerdo a la fig. 55 se aprecia que a medida que aumenta el número de nodos en la red, la cantidad de colisiones también aumenta de manera significativa. Esto sugiere que la probabilidad de que los paquetes colisionen entre sí durante la transmisión se incrementa a medida que más nodos participan en la red.

A medida que se implementan más nodos en una red LoRaWAN, es esencial considerar tanto el rendimiento como la escalabilidad. Si la red se vuelve demasiado congestionada y las colisiones son frecuentes, esto podría comprometer la confiabilidad y eficiencia de la comunicación. Es importante evaluar cómo el sistema maneja el aumento en la carga de tráfico y qué estrategias se pueden aplicar para mantener una tasa de colisiones aceptable.

**Figura 55.**  
*Colisiones en relación con el número de nodos*



Fuente: Autor

### **5.10. Materiales**

Existen diferentes instrumentos de medición para cada una de las variables a continuación de mencionaran algunos de ellos con sus características generales.

#### **5.10.1. Equipo para la medición del nivel de agua**

Los Sensores ultrasónicos son una alternativa para poder obtener los datos de nivel de agua, el sensor emite pulsos ultrasónicos que se reflejan en la superficie del producto. Al medir el tiempo de propagación de la onda entre el envío y la recepción de la onda reflejada, se puede calcular la distancia entre el sensor y la superficie del producto y su nivel de llenado (Direct, 2019).

**Figura 56.**  
Sensor ultrasónico JSN-SR04T-V3.0



*Nota.* Es un dispositivo que mide la distancia hasta un objeto usando ondas ultrasónicas, con un alcance entre 25 y 450 cm. Fuente: (Naylamp, 2021b).

**Tabla 11.**  
Características del sensor JSN-SR04T-V30

Especificaciones técnicas	
Voltaje de Operación	5V DC
Corriente de trabajo	30 mA
Rango de detección	25 cm-450 cm
Precisión	3 mm o 0.3 cm
Frecuencia de emisión acústica	40 KHz
Duración mínima del pulso de disparo (nivel TTL)	10 $\mu$ S
Tiempo mínimo de espera entre una medida y el inicio de otra	20 mS
Temperatura de trabajo	-10°C hasta 70°C

*Nota.* Las especificaciones técnicas fueron tomadas de (Naylamp, 2021b).

### **5.10.2. Equipos para la medición de precipitaciones**

Un pluviómetro mide de forma exacta y fiable cualquier tipo de precipitaciones como llovizna, lluvia intensa, cellisca, granizo o nieve. Con el método de medición por pesaje se registra tanto la cantidad como la intensidad de todas las precipitaciones considerando factores marginales como la temperatura y el viento (Sánchez, 2011).

Un pluviómetro almacena agua en una bureta con un área transversal más pequeña que la boca del pluviómetro. El agua recolectada se lee una vez al día (Sánchez, 2011).

**Figura 57.**  
*Pluviómetro de balancín Misol WH-SP-RG*



*Nota.* Pluviómetro Misol. Fuente:(Impormel, 2020).

**Tabla 12.**  
*Características técnicas del pluviómetro Misol*

Especificaciones técnicas	
Rango de visualización de lluvia	0 a 0.278 mm
Material	Plástico ABS
Salida	Pulso
Dimensiones	19cm * 7cm * 12cm

*Nota.* Pluviómetro Misol. Fuente: (Impormel, 2020).

### **5.10.3. Equipo para la medición de humedad, temperatura y presión barométrica**

EL Sensor BME280 integra en un solo dispositivo sensores de presión atmosférica, temperatura y humedad relativa, con gran precisión, bajo consumo energético y un formato ultra compacto (Naylamp, 2021a).

**Figura 58.**

*Sensor BME/BMP280*



*Nota.* El sensor BME280 es capaz de medir presión atmosférica, temperatura y humedad relativa. Fuente: (Naylamp, 2021a).

**Tabla 13.**

*Características del sensor BME280*

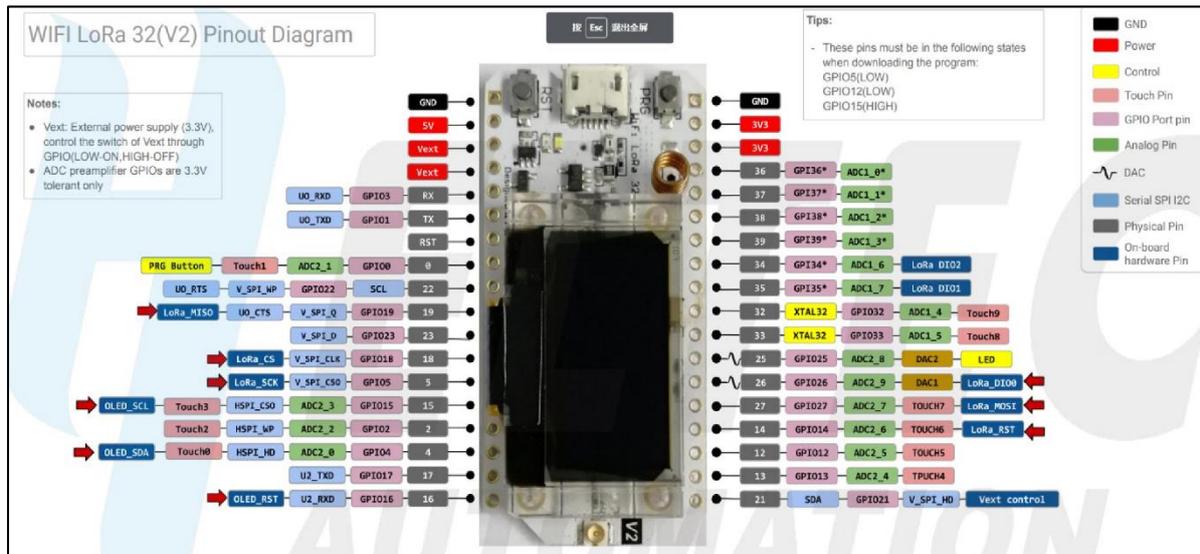
Especificaciones técnicas	
Voltaje de Operación	1.8V -3.3V DC
Interfaz de comunicación	I2C o SPI (3.3v)
Rango de presión	300 a 1100 hPa (0.3-1.1bar)
Resolución	0.16 Pa
Presión absoluta	1 hPa
Rango de Temperatura	-40°C a 85°C
Precisión de Temperatura	1°C
Rango de Humedad Relativa	0-100% RH
Precisión de HR	+3%
Rango de altura medible	0-9100 metros
Frecuencia de Muestreo	157 Hz (máx.)

*Nota.* Es un sensor de bajo consumo energético. Fuente: (Naylamp, 2021b).

#### **5.10.4. Nodo Heltec WIFI LoRA 32**

El dispositivo Wifi LoRa 32 es una placa de desarrollo de IoT clásica diseñada y producida por Heltec Automation (TM), es un producto altamente integrado basado en ESP32 SX1276, tiene Wi-Fi, funciones LoRa, también sistema de gestión de batería Li-Po, OLED de 0,96 " son También incluido. Es la mejor opción para ciudades inteligentes, granjas inteligentes, hogares inteligentes y muchos fabricantes.

**Figura 59.**  
Modulo Wifi Lora 32



*Nota.* El módulo incorpora un microprocesador ESP32 MCU de doble núcleo de 32 bits + núcleo ULP. Fuente: (EBYTE, 2019).

**Tabla 14.**  
Características del módulo Heltec Wifi Lora 32

Especificaciones técnicas	
LoRa Chipset	SX1276/SX1278
Frecuencia	863-923 MHz
Potencia máxima Tx	19dB ± 1dB
Sensibilidad de Recepción	-135 dBm
Memoria	8MB (64M-bits) SPI FLASH; 520KB internal SRAM
Pantalla	0.96-pulgadas OLED
Potencia de consumo	LoRa 20dB Salida
Ganancia de la antena	5dBi

*Nota.* El módulo está dentro de los microcontroladores de bajo consumo de potencia con múltiples módulos de comunicación múltiples integrados. Fuente: (Heltec, 2018)

### 5.10.5. Gateway wAP LR9

El equipo wAP LR9 es un Gateway creado por MikroTik, este dispositivo cuenta con reenvío de paquetes UDP preinstalado que es el encargado de la recolección de tráfico de la red, cuenta con punto de acceso inalámbrico y un puerto de Ethernet, incorpora una antena de interna de 2dBi con la opción de conectar una antena externa para una mayor ganancia. El dispositivo ofrece un rendimiento excelente y estable en casi cualquier condición climática.

**Figura 60.**  
Gateway wAP LR9



*Nota.* Cuenta con protección IP54 lo que le hace perfecto para exteriores. Fuente: (Mikrotik, 2020).

**Tabla 15.**  
Características wAP LR9

Especificaciones técnicas	
RAM	64MB
CPU	QCA9531
Frecuencia nominal de CPU	650MHZ
Sistema Operativo	RouterOS
Wifi	2.4 Ghz 802.11b/g/n
Ethernet	10/100 Ethernet ports
Potencia LoRa Tx	23 dBm
Sensibilidad Lora	-137dB
Temperatura de trabajo	-40°C a 60°C
Ganancia de la Antena	2dBi

Fuente: (Mikrotik, 2020)

## 6. Resultados

### 6.1. Simulación de radio enlace y cobertura

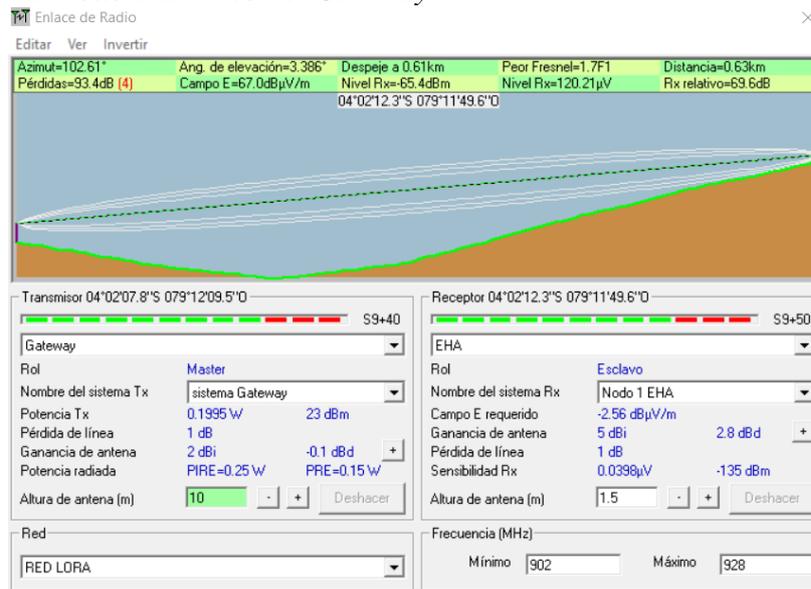
Para el análisis de cobertura se utilizó el software Radio Mobile y Google Earth En la red LoRaWAN, la tecnología LoRa permite que la transmisión entre el Gateway y el Nodo final sea robusta. Esto se debe a la modulación LoRa que atraviesa obstáculos eficientemente, al uso de frecuencias bajas que superan atenuaciones, a la tolerancia a la pérdida de señal y a la capacidad de ajustar la potencia de transmisión.

En la siguiente tabla se muestran los parámetros de configuración que se usaron para la simulación en Radio Mobile.

Parámetros	Gateway	Nodo 1 EHA	Nodo 2 EMA
Coordenadas	Latitud: 4° 2'7.8"S Longitud:79°12'9.5"O	Latitud: 4° 2'12.3"S Longitud:79°11'49.6"O	Latitud: 4° 2'17.6"S Longitud:79°11'18.00"O
Altura	10m	1.5m	1.5m
Ganancia de antena	2dBi	5dBi	5dBi
Potencia de transmisión máxima	23dBm	19dBm	19dBm
Sensibilidad de recepción máxima	-137dB	-135 dBm	-135 dBm

Los resultados de la simulación en RadioMobile indican niveles óptimos de señal. La distancia entre el Gateway y EHA es de 0.63 km como se muestra en la Fig. 61, con un azimut norte verdadero de 102.61°, con un ángulo de elevación es de 3.3861°. La simulación confirma que la propagación se realiza en línea de vista, con un despeje mínimo de 0.61 km a 1.7 F1 correspondiente a la zona de Fresnel. Estos valores indican una robusta y eficiente transmisión de señal entre los dispositivos.

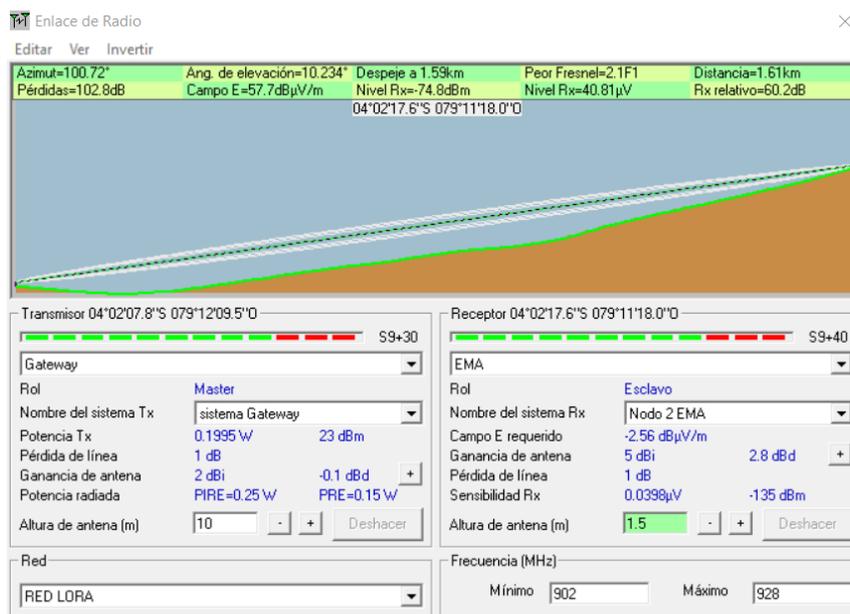
**Figura 61.**  
Radioenlace entre el nodo 1 EHA con el Gateway



Nota: se consideraron los datos técnicos de funcionamiento de los equipos, como potencia de transmisión y sensibilidad. Fuente: Autor.

La fig. 62, muestra el radioenlace entre el Gateway el Nodo EMA, la distancia entre ambos puntos es de 1.63 km, con un azimut norte verdadero de 100.72° con un ángulo de elevación se sitúa en 10.234°, la simulación confirma que la propagación sigue una línea de vista, con un despeje mínimo de 1.6 km y 2.1 F1 a la zona de Fresnel.

**Figura 62.**  
Radioenlace entre el Nodo 2 EMA con el Gateway



Nota: se consideraron los datos técnicos de funcionamiento de los equipos, como potencia de transmisión y sensibilidad. Fuente: Autor.

## 6.2. Cobertura

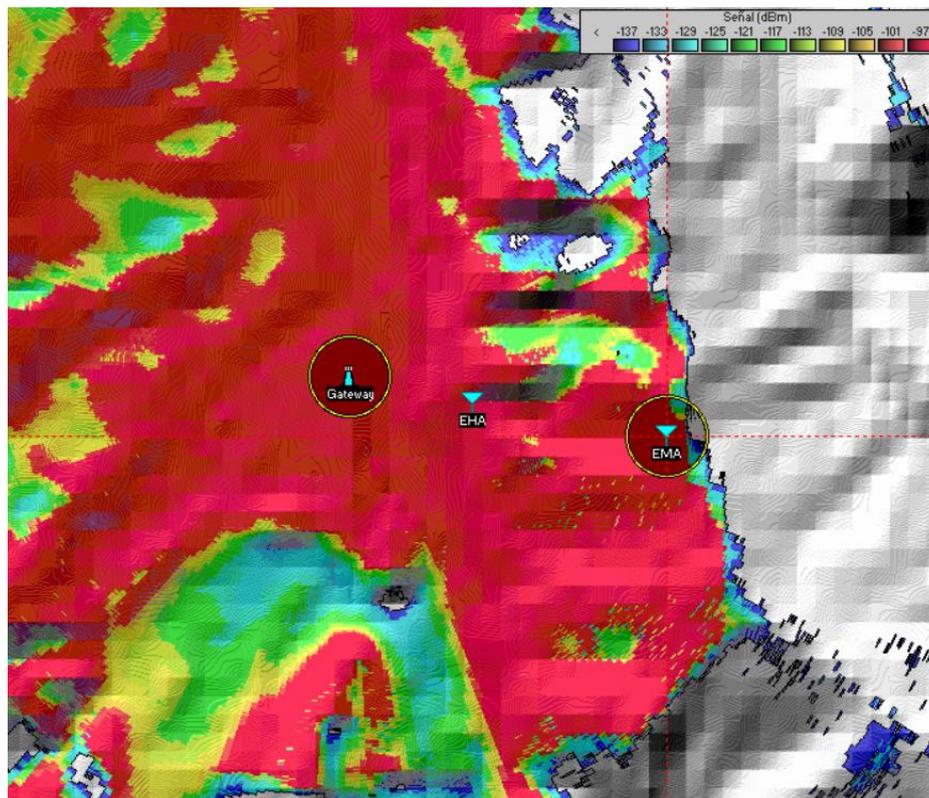
La siguiente tabla muestra los datos tomados en campo de los puntos en donde se van a instalar las los nodos.

Nodos	Distancia respecto al Gateway	RSSI en el Nodo	SNR en el nodo	RSSI Gateway	SNR Gateway	Factor de dispersión Uplink
Nodo 1 EHA	0.63 km	-97	5	-99	8.25	SF7
Nodo 2 EMA	1.62 km	-86	6	-85	7.75	SF7

La fig. 63, presenta una simulación de la cobertura del Gateway, realizada a través de Radio Mobile. En la gráfica se observa los diferentes niveles de potencia de la señal

**Figura 63.**

*Cobertura del Gateway*

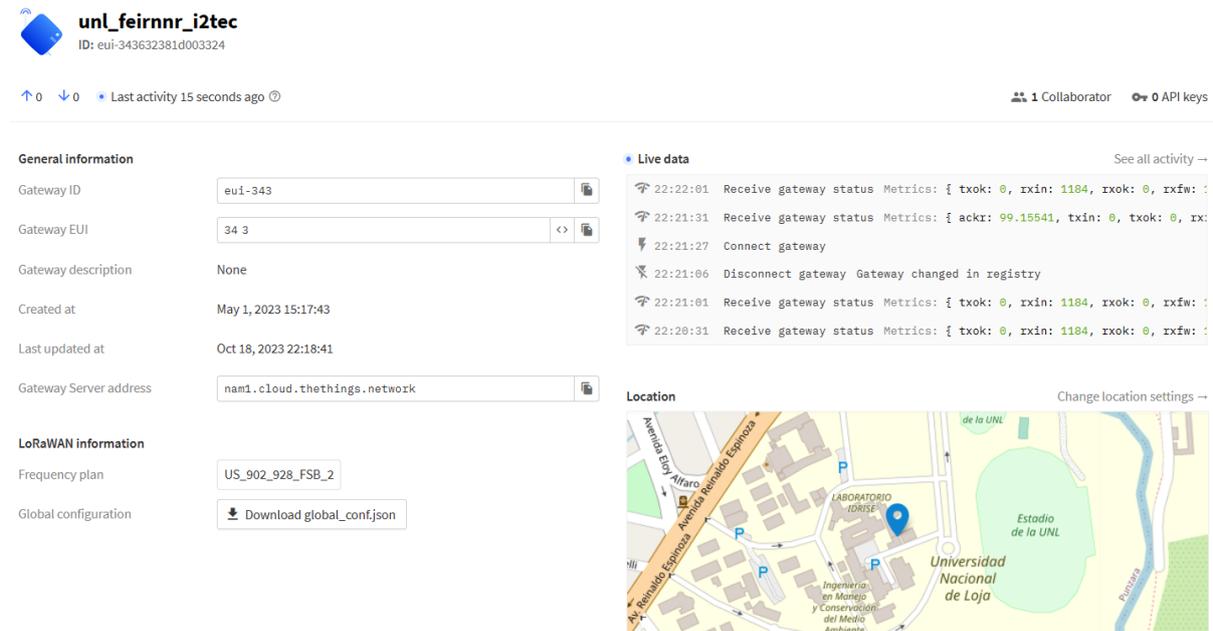


Fuente: Autor.

### 6.3. Integración de TTN con el Gateway y el Nodo Final

Se logro integrar de manera exitosa los dispositivos como el Gateway y el nodo final a TTN, siguiente los pasos y consideraciones mencionados en el apartado de la metodología. En la fig. 64 se puede apreciar la integración del Gateway, esto se puede verificar su conexión revisando el estado del Gateway y revisando los mensajes en tiempo real.

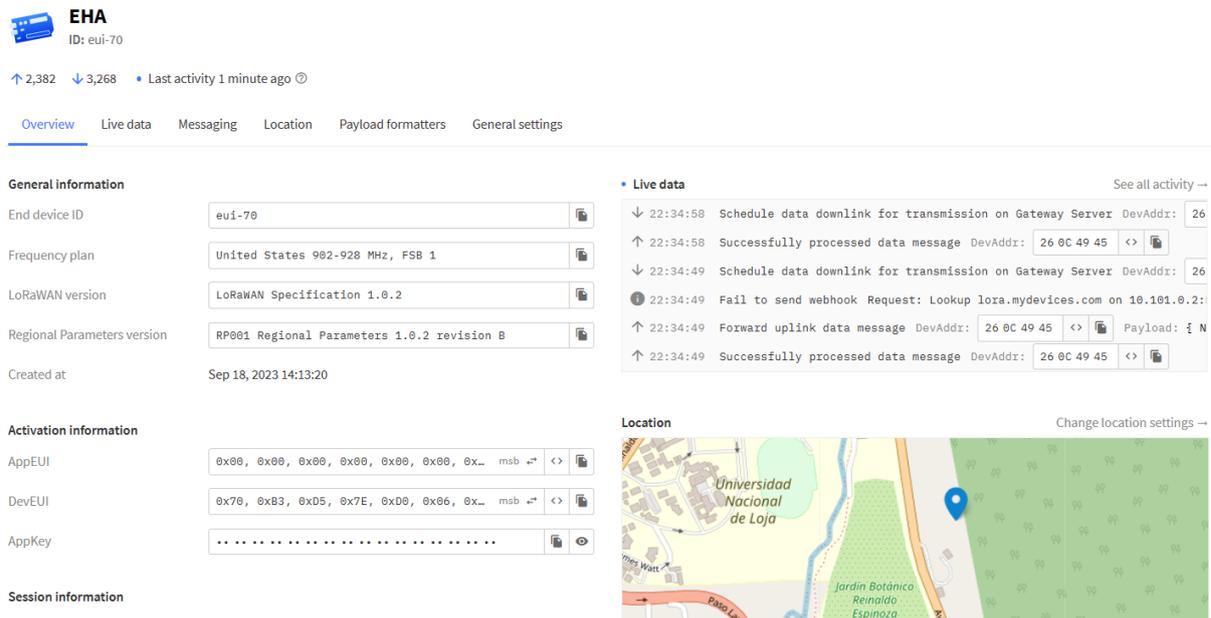
**Figura 64.**  
*Gateway conectado a TTN*



Fuente: Autor.

Una vez integrado el Gateway a TTN se procedió a la integración del nodo final el cual es el encargado de recopilar los datos y enviarlos al Gateway en TTN. En la fig. 65 se puede apreciar la integración exitosa del nodo final, se puede verificar el estado del dispositivo revisando los datos de transmisión en tiempo real en este caso revisando los mensajes de subida.

**Figura 65.**  
*Nodo final conectado a TTN*

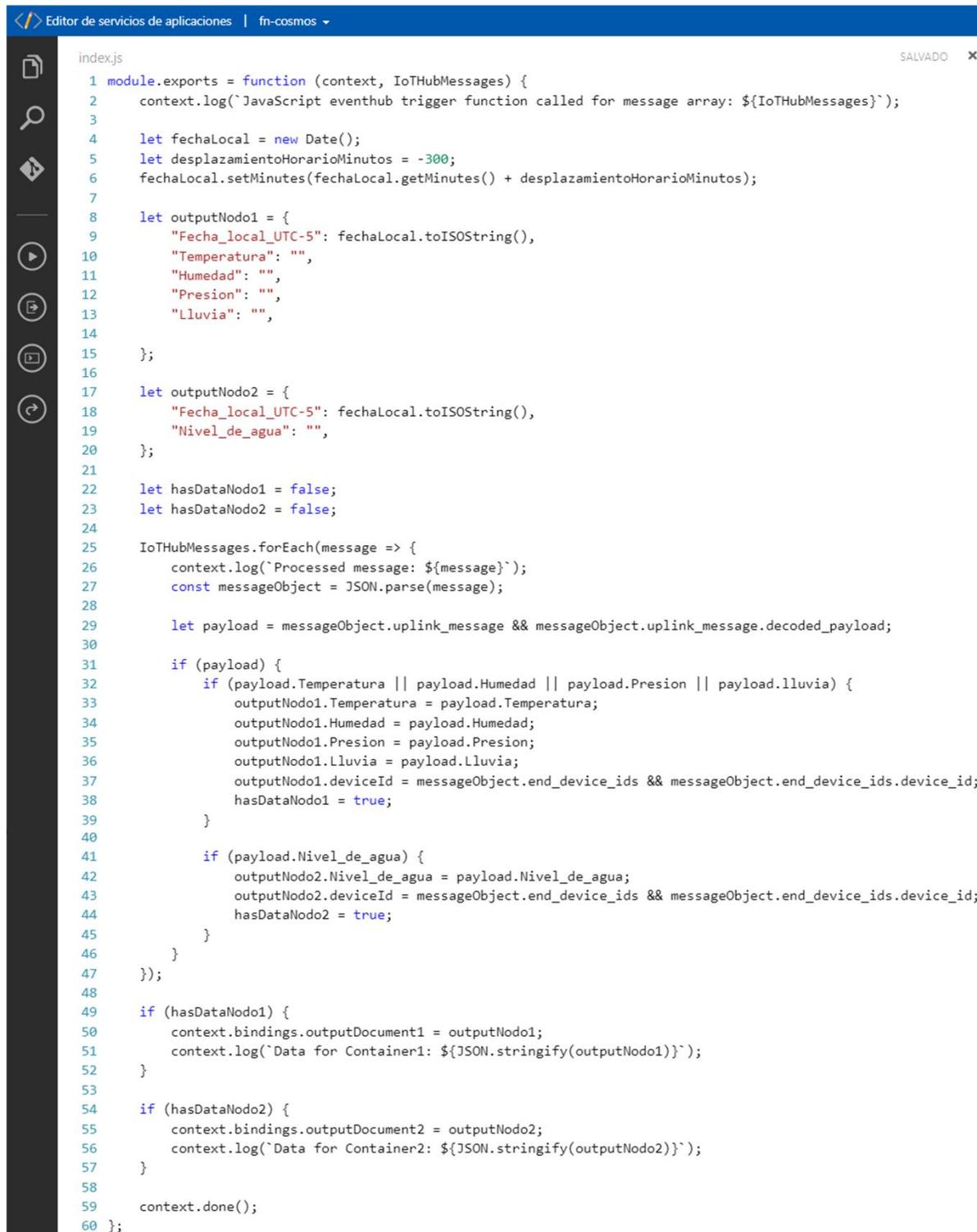


Fuente: Autor

#### 6.4. Integración de servicios de Azure

Una vez se ha establecido la conexión entre el servicio Azure IoT Hub y TTN, se inicia el siguiente paso de proceso de filtrado de datos a través de Azure Functions. Este proceso desencadena la extracción de las variables capturadas por los sensores, permitiendo así un almacenamiento eficiente y estructurado en la base de datos.

**Figura 66.**  
*Lectura y almacenamiento de variables*



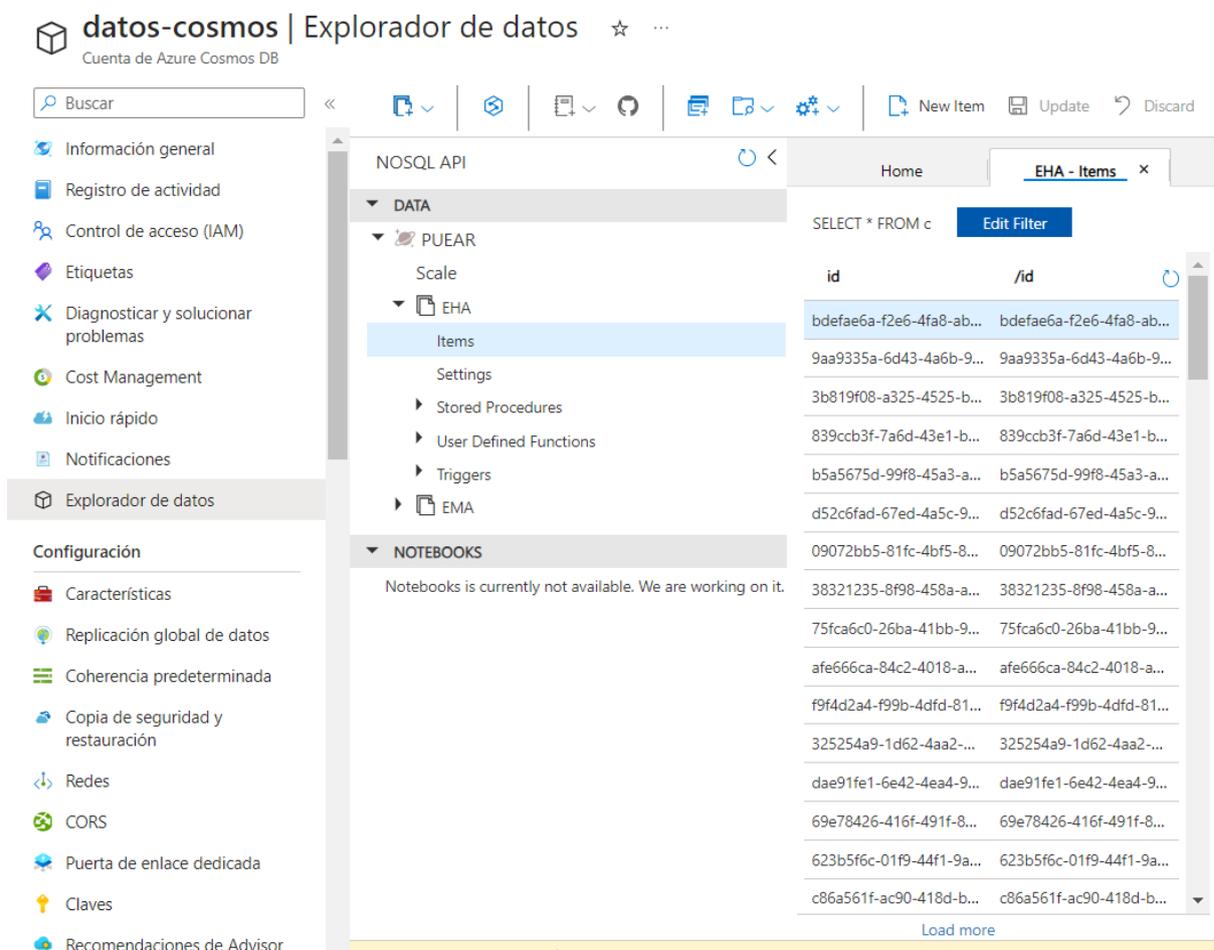
```
index.js SALVADO x
1 module.exports = function (context, IoTHubMessages) {
2   context.log(`JavaScript eventhub trigger function called for message array: ${IoTHubMessages}`);
3
4   let fechaLocal = new Date();
5   let desplazamientoHorarioMinutos = -300;
6   fechaLocal.setMinutes(fechaLocal.getMinutes() + desplazamientoHorarioMinutos);
7
8   let outputNodo1 = {
9     "Fecha_local_UTC-5": fechaLocal.toISOString(),
10    "Temperatura": "",
11    "Humedad": "",
12    "Presion": "",
13    "Lluvia": "",
14
15  };
16
17  let outputNodo2 = {
18    "Fecha_local_UTC-5": fechaLocal.toISOString(),
19    "Nivel_de_agua": "",
20  };
21
22  let hasDataNodo1 = false;
23  let hasDataNodo2 = false;
24
25  IoTHubMessages.forEach(message => {
26    context.log(`Processed message: ${message}`);
27    const messageObject = JSON.parse(message);
28
29    let payload = messageObject.uplink_message && messageObject.uplink_message.decoded_payload;
30
31    if (payload) {
32      if (payload.Temperatura || payload.Humedad || payload.Presion || payload.lluvia) {
33        outputNodo1.Temperatura = payload.Temperatura;
34        outputNodo1.Humedad = payload.Humedad;
35        outputNodo1.Presion = payload.Presion;
36        outputNodo1.Lluvia = payload.Lluvia;
37        outputNodo1.deviceId = messageObject.end_device_ids && messageObject.end_device_ids.device_id;
38        hasDataNodo1 = true;
39      }
40
41      if (payload.Nivel_de_agua) {
42        outputNodo2.Nivel_de_agua = payload.Nivel_de_agua;
43        outputNodo2.deviceId = messageObject.end_device_ids && messageObject.end_device_ids.device_id;
44        hasDataNodo2 = true;
45      }
46    }
47  });
48
49  if (hasDataNodo1) {
50    context.bindings.outputDocument1 = outputNodo1;
51    context.log(`Data for Container1: ${JSON.stringify(outputNodo1)}`);
52  }
53
54  if (hasDataNodo2) {
55    context.bindings.outputDocument2 = outputNodo2;
56    context.log(`Data for Container2: ${JSON.stringify(outputNodo2)}`);
57  }
58
59  context.done();
60 };
```

Fuente: Autor

Una vez completada la integración de los servicios de Azure, es crucial verificar la exactitud de los datos registrados en nuestra base de datos. Este procedimiento se puede llevar

a cabo desde la sección 'Explorador de Datos' en la plataforma de Cosmos DB. En primer lugar, es necesario seleccionar la base de datos correspondiente a través del nombre asignado durante su creación. Posteriormente, al acceder a la sección de 'Ítems', será posible visualizar una tabla que contiene los datos almacenados hasta el momento.

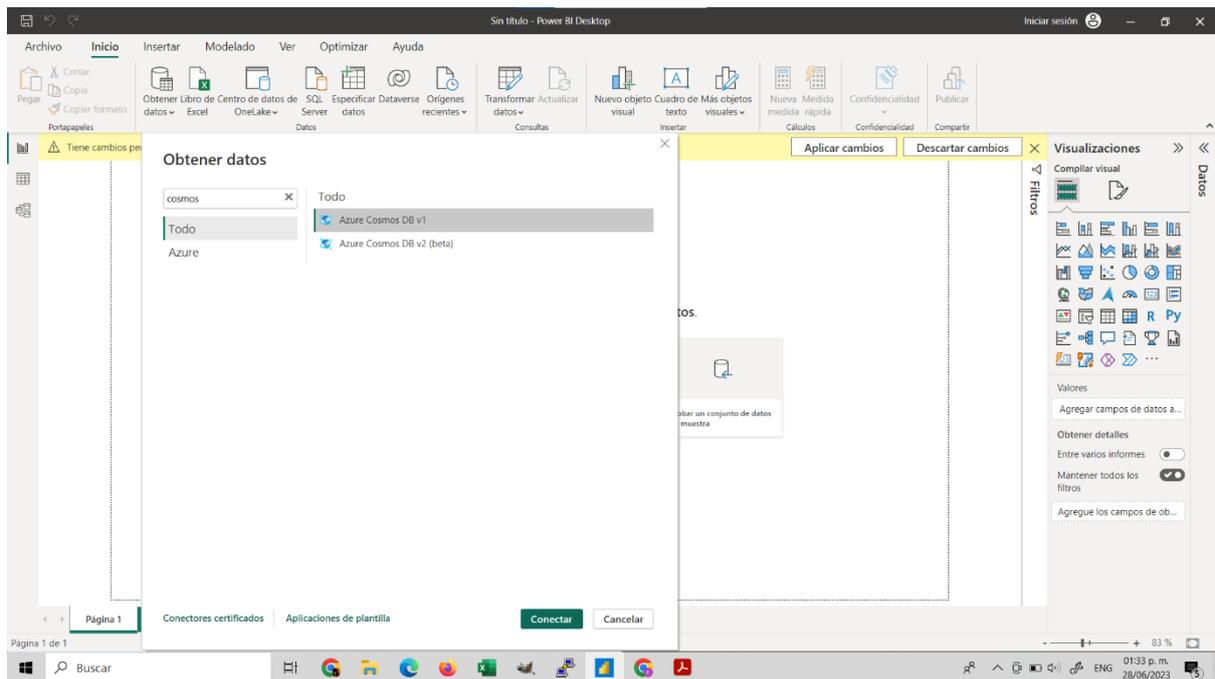
**Figura 67.**  
*Datos almacenados en Cosmos DB*



Fuente: Autor

Con el objetivo de visualizar los datos almacenados en la base de datos, se recurrió a la herramienta de escritorio Power BI. En la fig. 68 se ilustran los pasos fundamentales para importar la información alojada en Cosmos DB. Para llevar a cabo este proceso, se inicia seleccionando la opción "Obtener Datos", seguida de la búsqueda de la alternativa correspondiente a Cosmos DB. Por último, se procede a la conexión al seleccionar la opción "Conectar", lo que permite establecer un vínculo directo con los datos almacenados en la base de datos.

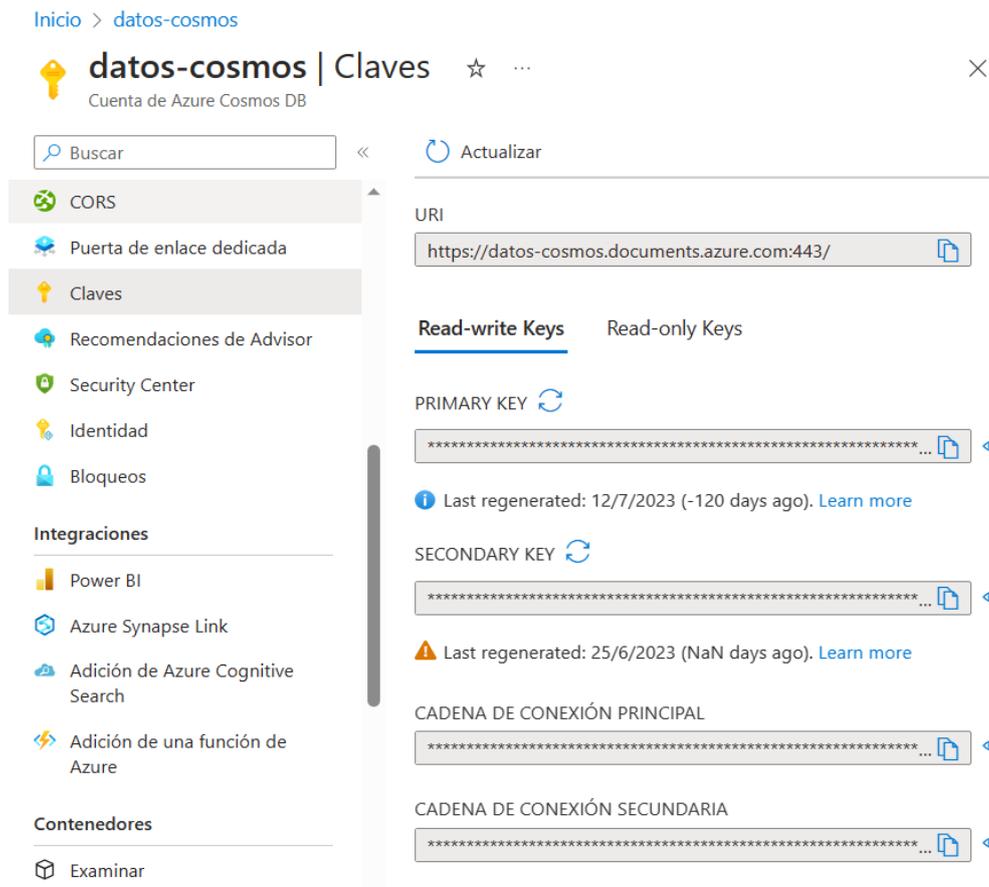
**Figura 68.**  
*Conexión de Power BI con Cosmos DB*



Fuente: Autor

Es fundamental disponer de la dirección “URI” y la “PRIMARY KEY” de Cosmos DB. Estas credenciales se encuentran disponibles en el portal de Cosmos DB, específicamente en la sección de exploración de datos bajo la opción "claves", tal como se ilustra en la fig. 69, estos datos son esenciales para establecer la conexión adecuada con la base de datos y garantizar un acceso seguro y controlado a la información almacenada.

**Figura 69.**  
*Datos de Conexión*



Fuente: Autor

Después de haber copiado los datos de conexión de Cosmos DB se debe pegarlos en Power BI para poder establecer dicha conexión, estas credenciales son necesarias únicamente cuando se conecte por primera vez. Luego de haber filtrado las columnas se genera una tabla como se observa en la fig. 70, con las cuales se va a proceder a realizar las gráficas para una mejor visualización,

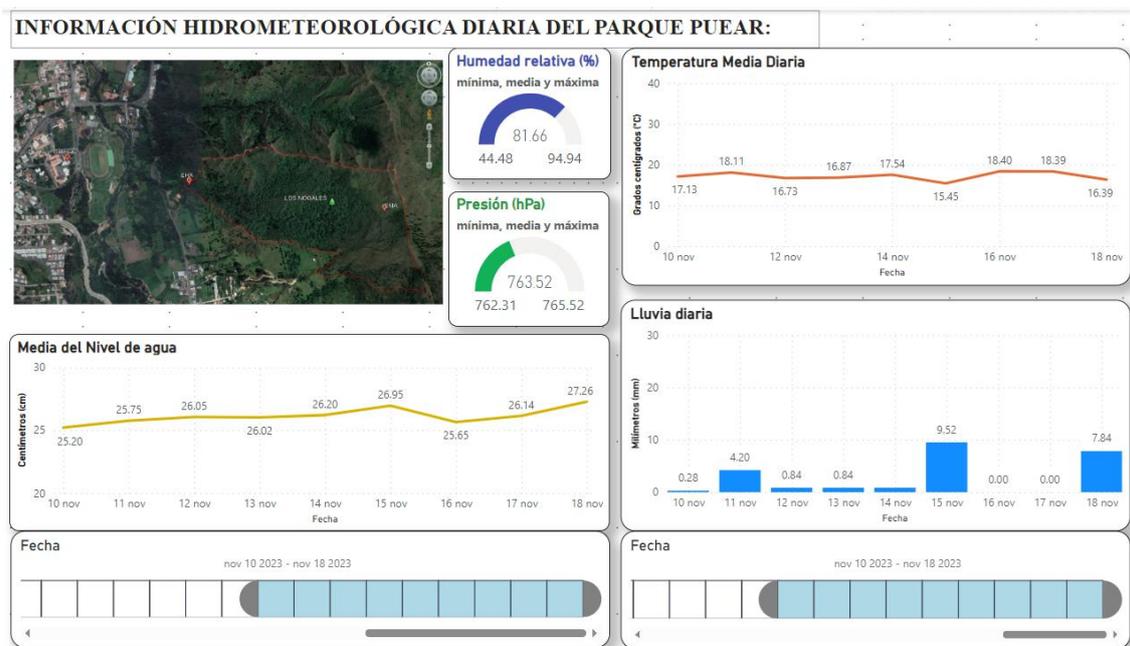
**Figura 70.**  
*Tabla de datos mostrada desde Power Bi*

	Document.Fecha_local_UTC-5	Document.Temperatura	Document.Humedad	Document.Presion	Document.Lluvia
1	21/09/2023 10:38:18 p. m. +00:00	35.41	28.76	874.46	0
2	21/09/2023 10:39:17 p. m. +00:00	35.51	28.67	874.52	0
3	21/09/2023 10:40:17 p. m. +00:00	35.53	28.64	874.48	0
4	21/09/2023 10:41:17 p. m. +00:00	35.27	29.43	874.49	13.97
5	21/09/2023 10:42:17 p. m. +00:00	35.41	28.77	874.45	0
6	21/09/2023 10:43:18 p. m. +00:00	35.48	28.69	874.46	0
7	21/09/2023 10:44:18 p. m. +00:00	35.62	28.39	874.48	0
8	21/09/2023 10:45:17 p. m. +00:00	35.48	28.63	874.42	0
9	21/09/2023 10:46:18 p. m. +00:00	35.5	28.64	874.36	0
10	21/09/2023 10:47:18 p. m. +00:00	35.51	28.52	874.44	0
11	21/09/2023 10:48:18 p. m. +00:00	35.45	28.83	874.43	0
12	21/09/2023 10:49:18 p. m. +00:00	35.47	28.68	874.44	0
13	21/09/2023 10:50:18 p. m. +00:00	35.46	28.99	874.46	0
14	21/09/2023 10:51:17 p. m. +00:00	35.58	28.5	874.49	0
15	21/09/2023 10:52:16 p. m. +00:00	35.49	28.71	874.47	0
16	21/09/2023 10:53:17 p. m. +00:00	35.56	28.45	874.47	0
17	21/09/2023 10:54:17 p. m. +00:00	35.54	28.85	874.47	0
18	21/09/2023 10:55:17 p. m. +00:00	35.5	28.91	874.48	0
19	21/09/2023 10:56:16 p. m. +00:00	35.36	28.84	874.48	0

Fuente: Autor

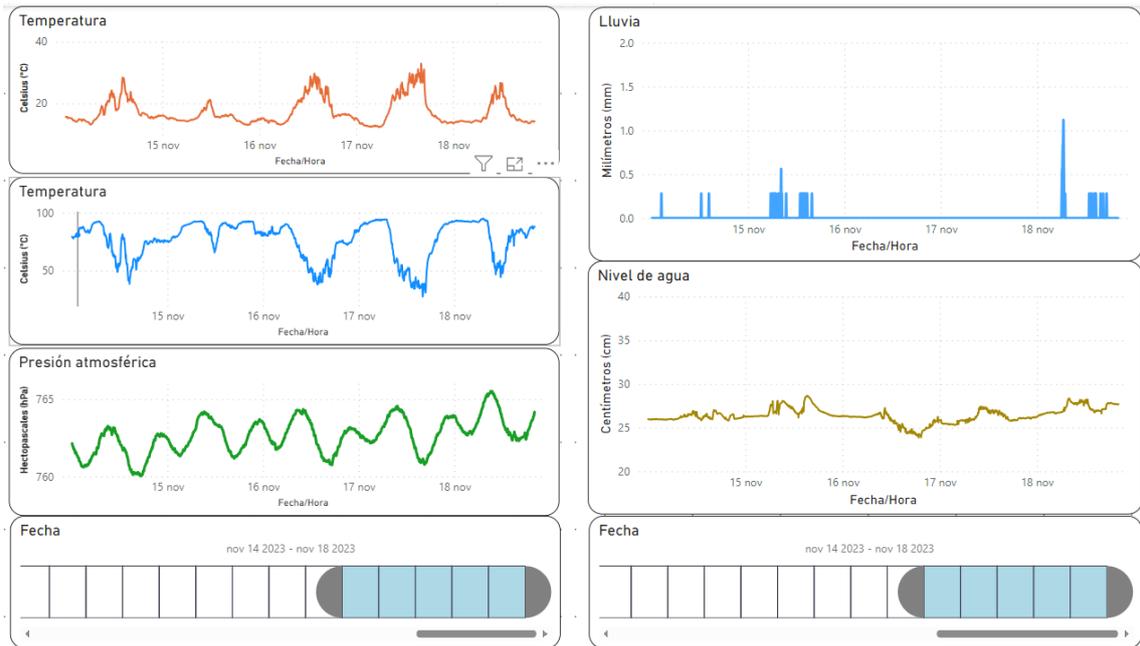
la fig. 71 incluye la información el promedio diario de las variables de nivel de agua, temperatura, humedad, presión y el total diario de lluvia. En la fig.71 se muestran los datos recopilados cada 5 minutos. Estos datos pueden visualizarse en intervalos de minutos, horas, días, semanas, meses o años.

**Figura 71.**  
 Visualización diaria de datos en Power BI



Fuente: Autor

**Figura 72.**  
*Visualización general de datos Power BI*



Fuente: Autor

## 7. Discusión

La presente tesis aborda el diseño y construcción de un prototipo de estación hidrológica autónoma, un proyecto que responde a la necesidad de mejorar la monitorización de variables hidrológicas en áreas remotas y de difícil acceso. Para alcanzar este objetivo, se ha empleado una combinación de tecnologías de vanguardia, destacando la red de área amplia de baja potencia (LoRaWAN) y la integración de servicios proporcionados por Microsoft Azure.

La implementación de LoRaWAN permitió la transmisión de datos de manera eficiente y económica a largas distancias, lo que resultó fundamental para la recolección de información hidrológica.

En paralelo, la integración de servicios de Microsoft Azure ha permitido la creación de una plataforma sólida, de fácil administración y altamente escalable para el almacenamiento de los datos recopilados. Es importante destacar que IoT Hub, dentro de esta plataforma, no solo facilita la conectividad con los dispositivos, sino que también habilita la visualización de los mensajes generados por estos dispositivos. Adicionalmente, Azure Functions juega un papel crucial al posibilitar la conexión entre los dispositivos finales registrados en IoT Hub y la aplicación de filtros para la selección y almacenamiento de datos en CosmosDB. Este proceso simplifica significativamente la accesibilidad a los datos en tiempo real y sienta las bases para la generación de informes personalizados en Power BI, los cuales son fundamentales para un análisis posterior.

En cuanto a los parámetros de medición, como el nivel de agua en la cuenca Los Nogales, es interesante observar que se registra un aumento en las horas de la mañana, seguido de una disminución en su nivel durante las tardes, especialmente en días en los cuales no se registra presencia de lluvia.

## 8. Conclusiones

Basándonos en los objetivos planteados y el contexto de la investigación, a continuación, se presentan las conclusiones de este estudio.

En el transcurso de esta investigación, se ha logrado con éxito el objetivo general de diseñar y construir una estación hidrológica autónoma (EHA) destinada a la captación de variables hidrológicas en los afluentes de agua del Parque Universitario de Educación Ambiental y Recreación (PUEAR) de la Universidad Nacional de Loja. Este logro representa un avance significativo en la capacidad de monitoreo y recolección de datos en el entorno del PUEAR.

La revisión bibliográfica ha permitido la identificación y elección de la tecnología LoraWAN siendo la más apropiada para la obtención y transmisión de datos, en conjunto con la plataforma Azure ha demostrado ser una excelente opción por ofrecer una amplia gama de herramientas y recursos específicamente diseñados para habilitar y gestionar soluciones de Internet de las cosas (IoT) de manera segura y escalable, los servicios de la plataforma, permite crear flujos de trabajo complejos y soluciones personalizadas de IoT. Esto incluye servicios como Azure Functions y Azure Cosmos DB, que ofrecen funcionalidades como cómputo sin servidor y almacenamiento de datos disponible.

Al analizar los resultados de simulación, se destaca una variación significativa en la tasa de extracción de datos (DER) en función de la cantidad de nodos y el valor del Factor de Dispersión (SF). Se observa que a medida que el número de nodos aumenta y se emplea un SF de 12, el DER experimenta una notable disminución. Este fenómeno sugiere la posibilidad de un incremento en colisiones y congestiones en la red, lo que, a su vez, resulta en una disminución en la tasa de entrega de paquetes. En contraste, cuando se utilizan SF de 7 en combinación con un menor número de nodos, se observa una mejora en el DER, indicando una mayor calidad de la transmisión.

Se ha logrado construir un prototipo de estación hidrológica que destaca por su autonomía energética y capacidad de transmisión de datos inalámbrica. Este prototipo se ha integrado con una plataforma web de IoT, lo que permitirá a docentes y estudiantes realizar análisis y estudios sobre el comportamiento de las variables hidrológicas en el PUEAR-UNL. La estación hidrológica se presenta como una herramienta valiosa para la investigación y el monitoreo continuo del entorno.

Como resultado de este trabajo, se ha elaborado un manual técnico detallado que servirá como guía para el montaje y mantenimiento de la estación hidrológica. Este recurso

proporciona información esencial sobre los materiales y elementos utilizados durante el montaje, lo que facilitará futuros proyectos similares y garantizará la consistencia en la implementación de estaciones hidrológicas autónomas.

## 9. Recomendaciones

Durante el desarrollo de este proyecto, se han identificado desafíos significativos que han llevado a la formulación de las siguientes recomendaciones

Se sugiere la incorporación de un sistema de carga solar en los sistemas de transmisión para garantizar un suministro continuo de energía. Esto es esencial para alimentar de manera constante los sensores y los nodos de transmisión, lo que contribuirá a la operación ininterrumpida de la estación hidrológica.

La revisión de la documentación oficial de los equipos, incluyendo su funcionamiento, protocolos de comunicación y la clasificación de los nodos finales, se ha demostrado como una etapa crítica en la configuración de los equipos. Esta práctica debe mantenerse como un estándar en futuros despliegues.

Se recomienda la utilización de una base de datos robusta y flexible para la administración y almacenamiento de datos. En este sentido, Microsoft Azure se destaca como una solución idónea. Su característica de servicio en la nube brinda accesibilidad desde cualquier lugar con conexión a internet, lo que facilita la gestión de datos.

Es crucial llevar a cabo una gestión eficiente de costos y recursos en Azure. Para ello, se sugiere explorar los tipos de servicios de costos por consumo y sus planes mensuales dentro de los servicios de suscripción de Azure. Esta medida permitirá un control adecuado tanto de los recursos económicos como del uso de datos.

## 10. Bibliografía

- Alliance, L. (2019). *RP002-1.0.0 LoRaWAN Regional Parameters*. [https://loralliance.org/resource\\_hub/rp002-1-0-0-lorawan-regional-parameters/](https://loralliance.org/resource_hub/rp002-1-0-0-lorawan-regional-parameters/)
- Altexsoft. (2020). *Making Sense of IoT Platforms: AWS vs Azure vs Google vs IBM vs Cisco*. <https://www.altexsoft.com/blog/iot-platforms/>
- Alvarado, E. (2017). *Manual de Medición de Caudales*. <https://icc.org.gt/wp-content/uploads/2023/03/064.pdf>
- Antonino, L. (2018). *Instalación de estaciones hidrométricas y meteorológicas*. Tesis de grado, Universidad Nacional de Cordoba.
- Araujo, M., Oliveira, H., Jan, M., Veira, M., & Wortche, H. (2023). Optimizing Resources and Increasing the Coverage of Internet-of-Things (IoT) Networks: An Approach Based on LoRaWAN. *Sensors 2023, Vol. 23, Page 1239, 23(3)*, 1239. <https://doi.org/10.3390/S23031239>
- Carrasco, G., & Elí, E. (2020). *Metodología para selección de tecnologías LPWAN para diversas aplicaciones de internet de las cosas*. <https://repositorio.uchile.cl/handle/2250/179720>
- Chamorro, G. (2008). *GUÍA DE HIDROMETRÍA DE AGUAS SUPERFICIALES GUÍA DE HIDROMETRÍA DE AGUAS SUPERFICIALES-Medición del Nivel de agua-SERVICIO NACIONAL DE METEOROLOGÍA E HIDROLOGÍA DIRECCIÓN GENERAL DE HIDROLOGÍA Y RECURSOS HÍDRICOS gchamorro@senamhi.gob.pe dgh@senamhi.gob.pe*
- Direct. (2019). *Qué sensor de nivel elegir* . <https://guide.directindustry.com/es/que-sensor-de-nivel-elegir/>
- EBYTE. (2019). *E32-433T30D User Manual*. <https://www.ebyte.com/en/product-view-news.html?id=108>
- Egli, P. (2015). *LPWAN OVERVIEW OF EMERGING TECHNOLOGIES FOR LOW POWER WIDE AREA NETWORKS IN INTERNET OF THINGS AND M2M SCENARIOS*. [https://indigoo.com/dox/itdp/12\\_MobileWireless/LPWAN.pdf?\\_ga=2.211](https://indigoo.com/dox/itdp/12_MobileWireless/LPWAN.pdf?_ga=2.211)
- Fernández, J. (2020). *Presión Atmosférica*. <https://www.fisicalab.com/apartado/presion-atmosferica>
- FloRa. (2022). *A framework for LoRa simulations with OMNeT++*. <https://flora.aalto.fi/>
- Flores, F., & Cossio, E. (2021). *Aplicaciones, Enfoques y Tendencias del Internet de las Cosas (IoT): Revisión Sistemática de la Literatura*.

- [https://ciateq.repositorioinstitucional.mx/jspui/bitstream/1020/543/1/Aplicaciones enfoques y tendencias del IoT.pdf](https://ciateq.repositorioinstitucional.mx/jspui/bitstream/1020/543/1/Aplicaciones%20enfoces%20y%20tendencias%20del%20IoT.pdf)
- Heltec. (2018). *Modulo Wifi LoRa 32*. <https://heltec.org/project/wifi-lora-32/>
- Hernández, S. (2020). *Estudio en detalle de LoRaWAN. Comparación con otras tecnologías LPWAN considerando diferentes patrones de tráfico*. <https://openaccess.uoc.edu/handle/10609/106369>
- Impormel. (2020). *Pluviometro Digital Acurite Medidor De Lluvia Meteorológico*. <http://impormel.com/equipos-de-medicion/1324-pluviometro-digital-acurite-medidor-de-lluvia-meteorologico.html>
- Instituto Nacional de Meteorología e Hidrología [INAMHI]. (2011). *Mapas Red Hidrológica*. [https://issuu.com/inamhi/docs/mapas\\_red\\_hidrol\\_gica/11](https://issuu.com/inamhi/docs/mapas_red_hidrol_gica/11)
- Instituto Nacional de Meteorología e Hidrología [INAMHI]. (2021). *Red de Estaciones Automáticas INAMHI*. <http://186.42.174.236/InamhiEmas/#>
- Kim, D. H., Lee, E. K., & Kim, J. (2019). Experiencing LoRa network establishment on a smart energy campus testbed. *Sustainability (Switzerland)*, 11(7). <https://doi.org/10.3390/SU11071917>
- Lopez, F., & Zurita, R. (2016). *Vertederos*. <https://instrumentacionuc.wixsite.com/facultad-ingenieria/copia-de-medidor-anemometro>
- Marie, M., & Carpi, A. (2003). *Temperatura*. <https://www.visionlearning.com/es/library/Ciencias-Generales/3/Temperatura/48>
- Martínez, A., Méndez, I., Rodríguez, N., & Zacarias, E. (2018). *La humedad en la atmósfera*. Universidad de Colima.
- Microsoft. (2022a). *Azure Cosmos DB*. [http://azure.microsoft.com/es-es/products/cosmos db/](http://azure.microsoft.com/es-es/products/cosmos%20db/)
- Microsoft. (2022b). *Azure functions*. <http://azure.microsoft.com/es-es/products/functions/>
- Microsoft. (2022c). *Azure IoT Hub*. <https://azure.microsoft.com/es-es/products/iot-hub/>
- Mikrotik. (2020). *wAP LR8 kit*. [https://mikrotik.com/product/wap\\_lr8\\_kit#fndtn-specifications](https://mikrotik.com/product/wap_lr8_kit#fndtn-specifications)
- Molina, L. (2022). *Análisis Del Comportamiento De La Precipitación Y Temperatura Según Registros Meteorológicos De La Estación Meteorológica Rumipamba En Los Últimos 30 Años Y El Primer Año De Pandemia. Cotopaxi – 2021*.
- Naylamp. (2021a). *Sensor de Presión, Temperatura y Humedad BME280*. <https://naylampmechatronics.com/sensores-posicion-inerciales-gps/357-sensor-de-presion-temperatura-y-humedad-bme280.html>

- Naylamp. (2021b). *Sensor ultrasonico JSN-SR04T-V3.0*.  
<https://naylampmechatronics.com/sensores-posicion-inerciales-gps/357-sensor-de-presion-temperatura-y-humedad-bme280.html>
- Organización Meteorológica Mundial [OMM]. (2020). *Guía de prácticas hidrológicas*.  
[https://library.wmo.int/doc\\_num.php?explnum\\_id=10456](https://library.wmo.int/doc_num.php?explnum_id=10456)
- Peña Queralta, J., Gia, T. N., Zou, Z., Tenhunen, H., & Westerlund, T. (2019a). Comparative study of LPWAN technologies on unlicensed bands for M2M communication in the IoT: Beyond Lora and Lorawan. *Procedia Computer Science*, 155, 343–350.  
<https://doi.org/10.1016/J.PROCS.2019.08.049>
- Peña Queralta, J., Gia, T. N., Zou, Z., Tenhunen, H., & Westerlund, T. (2019b). Comparative Study of LPWAN Technologies on Unlicensed Bands for M2M Communication in the IoT: beyond LoRa and LoRaWAN. *Procedia Computer Science*, 155, 343–350.  
<https://doi.org/10.1016/J.PROCS.2019.08.049>
- Sánchez, J. (2011). *Precipitaciones* . <https://hidrologia.usal.es/temas/Precipitaciones.pdf>
- Semtech. (2021). *LoRa and LoRaWAN*. <https://loradevelopers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>
- The Things Network. (2022). *LoRaWAN*. <https://www.thethingsnetwork.org/>

## 11. Anexos

### Anexo 1. Manual técnico de montaje y mantenimiento de las estaciones

#### Contenido

1. Descripción de las Estación Autónomas
2. Materiales
3. Montaje y mantenimiento de la estación EHA
4. Montaje y mantenimiento de la estación EMA

Este manual se ha diseñado con el propósito de proporcionar una guía para el montaje y mantenimiento de esta estación, que no solo asegura la captura precisa de datos hidrológicos, sino que también sienta las bases para proyectos similares en el futuro.

#### 1. Descripción de las Estación Autónomas

la Estación Hidrológica Autónoma (EHA) y la Estación Meteorológica Autónoma (EMA). Ambas estaciones están diseñadas para la captación de datos en tiempo real en el entorno del Parque Universitario de Educación Ambiental y Recreación (PUEAR) de la Universidad Nacional de Loja.

##### **EHA (Estación Hidrológica Autónoma):**

- Recopilación de Datos Hidrológicos: La EHA mide con precisión el nivel de agua en cuerpos de agua del área de interés.
- Transmisión de Datos: Utiliza tecnología LoRaWAN para transmitir estos datos a la plataforma en la nube, donde están disponibles para su posterior análisis.

##### **EMA (Estación Meteorológica Autónoma):**

- Recopilación de Datos Meteorológicos: La EMA mide variables meteorológicas, incluyendo temperatura, humedad, presión atmosférica y precipitaciones a través del pluviómetro.
- Transmisión de Datos: Utiliza tecnología LoRaWAN para transmitir estos datos a la plataforma en la nube, donde están disponibles para su posterior análisis.
- Alimentación Autónoma: Cada estación cuenta con su propia fuente de alimentación autónoma, que incluyen baterías de 12 v recargables con un sistema de energía solar el cual incluye un panel solar de 20 w y un controlador de carga, asegurando un funcionamiento ininterrumpido.

Conectividad con Plataforma en la Nube (Microsoft Azure): Los datos recopilados por ambas estaciones se transmiten a una plataforma en la nube, que permite su lectura y almacenamiento en una base de datos.

Aplicación en Estudios Hidrológicos y Ambientales: se han diseñado para proyectos de investigación y educación ambiental, permitiendo el monitoreo continuo de las condiciones hidrológicas en el PUEAR y áreas similares.

## 2. Materiales

Una parte fundamental para garantizar el montaje exitoso y el mantenimiento adecuado de las estaciones autónomas es contar con una lista completa de los materiales. A continuación, se detalla una lista detallada de los elementos esenciales para cada una de las estaciones.

### 2.1. Materiales para la EHA (Estación Hidrológica Autónoma)

Una parte fundamental para garantizar el montaje exitoso y el mantenimiento adecuado de las estaciones autónomas es contar con una lista completa de los materiales. A continuación, se detalla una lista detallada de los elementos esenciales para cada una de las estaciones:

**Tabla 16.**

*Materiales usados en el montaje de la estación EHA*

Descripción	Equipo
<p><b>Panel Solar monocristalino Soltek</b></p> <p>Potencia máxima nominal (pmax): 20W                      Rango de tolerancia de energía: 0-+3%                      Voltaje de circuito abierto (Voc): 21.96 v                      Voltaje de potencia máximo (Vmp): 18.30 v                      Corriente de cortocircuito (Isc): 1.7A                      Corriente de potencia máxima (Imp): 1.10A</p>	
<p><b>Controlador de carga solar modelo DX-10<sup>a</sup></b></p> <p>Voltaje de batería: 12v/24v                      Corriente de carga: 10<sup>a</sup>                      Entrada solar máxima: &lt;41v                      Carga Flotante: 14.4 v                      parada de descarga: 10.7 v                      Reconexión de descarga: 12.6v                      Autoconsumo: &lt; 10mA                      salida USB: 2 salidas,5v/2.5(Max)                      Temperatura de funcionamiento -35~+60 °C</p>	
<p><b>Batería de gel FirstPower (12V 5Ah/20Hr)</b></p> <p>Corrientes inicial máxima: 1.5A                      Uso en espera: 13.5 ~ 13.8v                      Ciclo de uso: 14.5 ~ 14.90v</p>	

<p><b>Regulador de voltaje y corriente xl4015e1</b></p> <p>Corriente de salida: 0A a 5A  - Voltaje de entrada: 6V a 38V  - Voltaje de salida: 1.25V a 36V  - Eficiencia máxima: 96%  - Potencia de salida: 75W</p>	
<p><b>Sensor ultrasónico JSN-SR04T-V3.0</b></p> <p>Voltaje de Operación: 5V DC  Corriente de trabajo: 30 mA  Rango de detección: 25 cm-450 cm  Precisión: 3 mm o 0.3 cm  Frecuencia de emisión acústica: 40 KHz  Duración mínima del pulso de disparo (nivel TTL): 10 μS  Tiempo mínimo de espera entre una medida y el inicio de otra: 20 mS  Temperatura de trabajo: -10°C hasta 70°C</p>	
<p><b>Modulo Heltec Lora 32.</b></p> <p>LoRa Chipset: SX1276/SX1278  Frecuencia: 863-923 MHz  Potencia máxima Tx: 19dB ± 1dB  Sensibilidad de Recepción: -135 dBm  Memoria: 8MB (64M-bits) SPI FLASH; 520KB internal SRAM  Pantalla: 0.96-pulgadas OLED  Potencia de consumo: 20dB  Ganancia de la antena: 5dBi</p>	

## 2.2. Materiales para la EMA (Estación Meteorológica Autónoma)

Nota: el sistema de carga solar de esta estación incluye los mismos modelos mencionados en la tabla anterior.

**Tabla 17.**

*Materiales usados en el montaje de la estación EMA*

Descripción	Equipo
<p><b>Pluviómetro de balancín Misol WH-SP-RG</b></p> <p>Rango de visualización de lluvia: 0 a 0.278 mm  Material: Plástico ABS  Salida: Pulso  Dimensiones: 19cm * 7cm * 12cm</p>	

<p><b>Sensor BME/BMP280</b></p> <p>Voltaje de Operación: 1.8V -3.3V DC  Interfaz de comunicación: 12C o SPI (3.3v)  Rango de presión: 300 a 1100 hPa (0.3-1.1bar)  Resolución: 0.16 Pa  Presión absoluta: 1 hPa  Rango de Temperatura: -40°C a 85°C  Precisión de Temperatura: 1°C  Rango de Humedad Relativa: 0-100% RH  Precisión de HR: +-3%  Rango de altura medible: 0-9100 metros  Frecuencia de Muestreo: 157 Hz (máx.)</p>	
<p><b>Modulo Heltec WiFi Lora 32.</b></p> <p>LoRa Chipset: SX1276/SX1278  Frecuencia: 863-923 MHz  Potencia máxima Tx: 19dB ± 1dB  Sensibilidad de Recepción: -135 dBm  Memoria: 8MB (64M-bits) SPI FLASH; 520KB internal SRAM  Pantalla: 0.96-pulgadas OLED  Potencia de consumo: 20dB  Ganancia de la antena: 5dBi</p>	

### 3. Montaje y mantenimiento de la estación EHA

#### Paso 1. Construcción del Vertedero

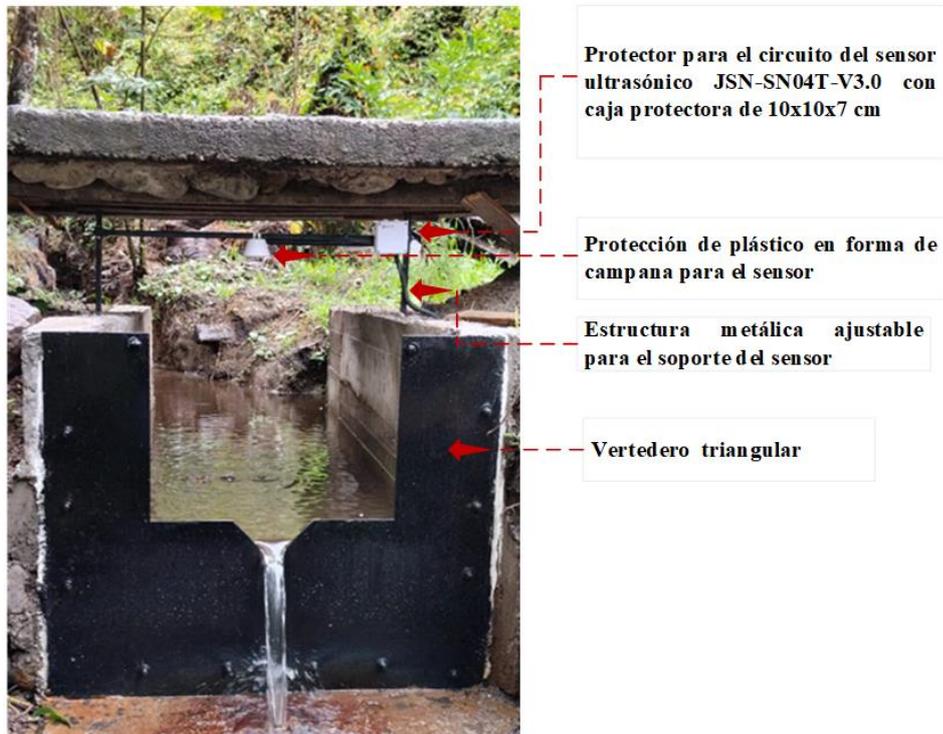
Como primera fase, se ejecutó la construcción de una canalización de concreto en la cuenca de Los Nogales. Posteriormente, se instaló una lámina de acero inoxidable siguiendo el diseño de un vertedero triangular, asegurándola con pernos. Se aplicó sellante en los bordes para prevenir filtraciones de agua.

#### Paso 2. Instalación del Sensor Ultrasónico

La estructura de soporte del sensor se ubicó a 1.20 metros desde la placa del vertedero triangular y a una altura de 1 metro desde el fondo del vertedero. Se diseñó e implementó un protector en forma de campana mediante impresión 3D para resguardar el sensor ultrasónico. Se instaló una caja protectora de plástico para el circuito del sensor, y se instaló un cable UTP de 15 metros para las conexiones de alimentación y señal del sensor.

### Figura 73.

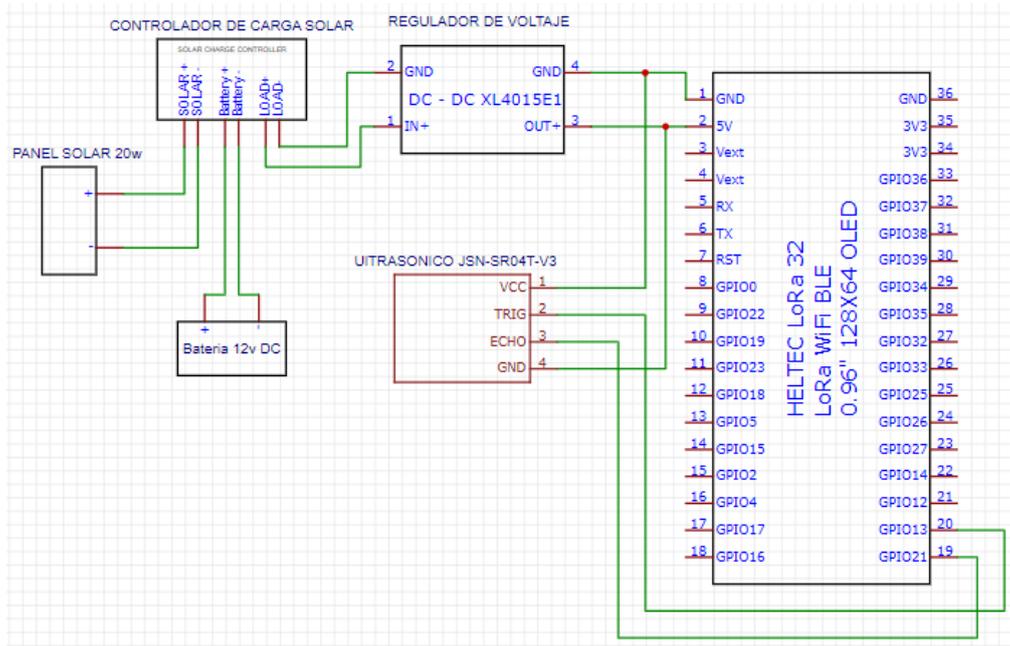
*Descripción de materiales usados en el vertedero*



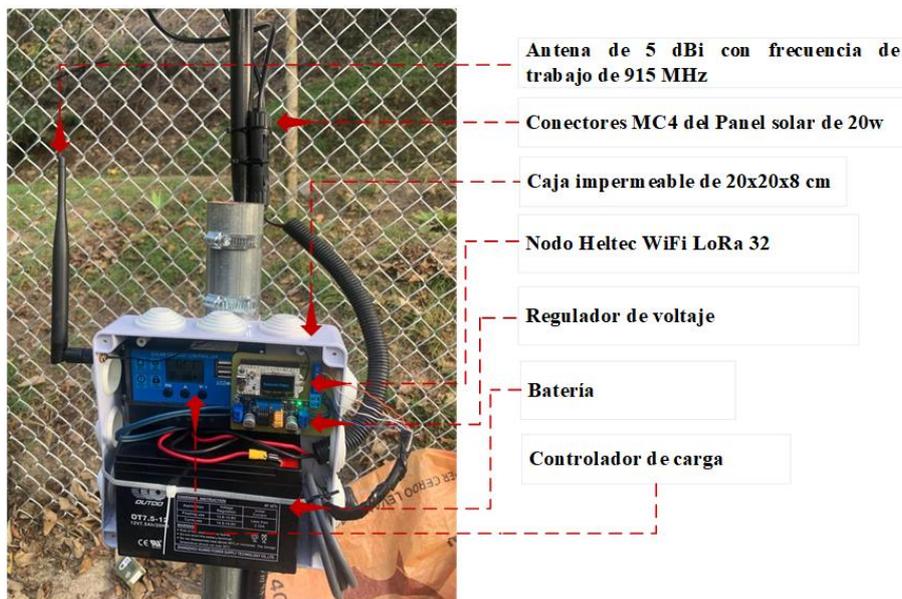
### Paso 3. Instalación del Nodo

Se soportó un soporte con tubo galvanizado de 1.5 metros de altura, seguido por la instalación del panel solar. Posteriormente, se fijó la caja protectora de plástico que albergaría los equipos. Se diseñó un circuito eléctrico que incluye la conexión del sistema de carga solar y la conexión de salida para la alimentación del módulo Lora.

**Figura 74.**  
*Diagrama electrónico*



**Figura 75.**  
*Descripción de materiales usados en el nodo EHA*



### 3.1. Mantenimiento

Realizar un mantenimiento regular y adecuado ayudará a garantizar la precisión y la confiabilidad del sistema de monitoreo a largo plazo.

#### Inspección regular

Realizar inspecciones visuales periódicas para asegurarse de que no haya obstrucciones en el vertedero o en el área circundante que puedan afectar la lectura del sensor ultrasónico.

Verificar la integridad de la lámina de acero del vertedero y la canalización de concreto para detectar posibles daños o desgastes.

### **Limpieza**

Limpiar cualquier residuo o sedimento que pueda acumularse en el vertedero y afectar la medición del nivel del agua. Asegurarse de que no haya obstrucciones en la campana protectora del sensor ultrasónico que puedan interferir con su capacidad para medir con precisión.

### **Revisión de conexiones**

Inspeccionar las conexiones eléctricas para asegurarse de que estén bien conectadas y sin daños, especialmente en la salida de carga del controlador y la salida del regulador de voltaje, en caso se presentarse irregularidades en el regulador de voltaje, el nodo se puede desconectarlo y conectarlo mediante USB al controlador de carga.

### **Calibración del sensor**

Realizar calibraciones periódicas del sensor ultrasónico para garantizar mediciones precisas. Ajustar la posición del sensor según sea necesario para mantener la precisión de las lecturas.

### **Revisión del sistema solar**

Inspeccionar el panel solar para asegurarse de que esté limpio y libre de obstrucciones que puedan afectar la carga de la batería, es recomendable limpiarlo con un paño de microfibra humedecido.

### **Registro de datos**

Revisar regularmente los registros de datos recopilados en por el sensor y la estación para identificar cualquier anomalía o tendencia significativa, la inspección se lo puede hacer en Power BI.

## **4. Montaje y mantenimiento de la estación EMA**

### **Paso 1. Instalación del Pluviómetro**

Se empleó un soporte metálico para la instalación del pluviómetro, asegurándose de que esté libre de obstáculos que puedan afectar la recolección precisa de la lluvia. La ubicación estratégica del pluviómetro es esencial para garantizar mediciones exactas.

### **Paso 2. Instalación del Sensor BME/BMP 280**

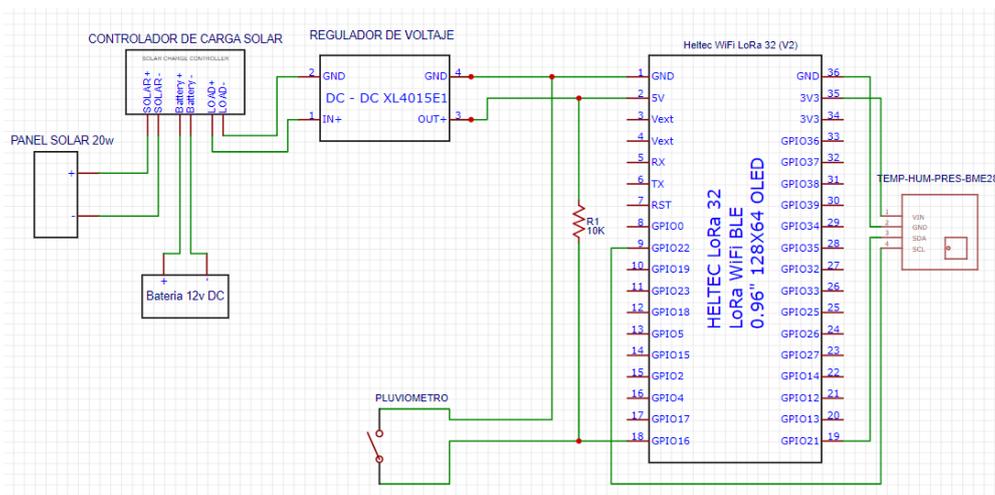
El sensor BME/BMP 280 se colocó dentro de una caja protectora diseñada específicamente para este propósito. Se realizaron recortes en la caja, y se aplicó una malla de plástico para asegurar la medición precisa de las variables ambientales. Esta medida garantiza que el sensor opere de manera eficiente y sin interferencias externas.

### Paso 3. Instalación del Nodo

Comenzamos con la instalación del nodo, asegurando un soporte robusto mediante un tubo galvanizado de 1.8 metros de altura. Acto seguido, se ubicó estratégicamente el panel solar para maximizar la captación de luz solar. La caja protectora de plástico se fijó para albergar los equipos de la estación. Es crucial ajustar el controlador de carga a los niveles óptimos para garantizar la carga máxima de la batería y el rendimiento continuo.

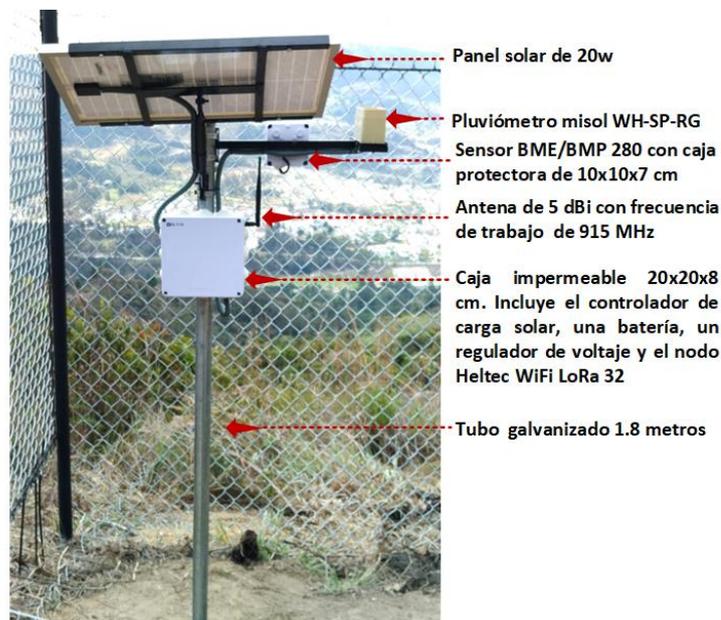
**Figura 76.**

*Diagrama electrónico*



**Figura 77.**

*Descripciones de materiales usados en el nodo EMA*



## **Mantenimiento**

### **Inspección Regular**

Realizar inspecciones visuales periódicas de todos los componentes para identificar posibles daños o desgastes especialmente el soporte y estructura.

### **Limpieza**

Limpiar el pluviómetro y la caja del sensor BME/BMP 280 para evitar obstrucciones que puedan afectar las mediciones, en especial el pluviómetro para evitar que se obstruya el paso de agua al balancín.

### **Revisión del sistema solar**

Inspeccionar el panel solar para asegurarse de que este posicionado correctamente y libre de obstrucciones que puedan afectar la carga de la batería, es recomendable limpiarlo con un paño de microfibra humedecido.

### **Verificación de Conexiones**

Inspeccionar las conexiones eléctricas para asegurarse de que estén bien conectadas y sin daños, especialmente en la salida de carga del controlador y la salida del regulador de voltaje, en caso se presentarse irregularidades en el regulador de voltaje, el nodo se puede desconectarlo y conectarlo mediante USB al controlador de carga.

### **Registro de Datos**

Revisar regularmente los registros de datos en Power Bi para detectar cualquier anomalía o tendencia significativa.

## Anexo 2. Código implementado para la conexión de la estación EHA con el equipo Heltec

```
#include "LoRaWan_APP.h" //Librería LoRaWAN proporcionado por el fabricante
#include "HT_SSD1306Wire.h" //Librería usada para escritura de texto en el display

extern SSD1306Wire display;
uint32_t license[4]={...}; // incluye licencia del equipo
/*Parámetros para la conexión con el Gateway*/
uint8_t devEui[] = {...};
uint8_t appEui[] = {...};
uint8_t appKey[] = {...};
/*ABP para*/
uint8_t nwkSKey[] = {...};
uint8_t appSKey[] = {...};
uint32_t devAddr = ( uint32_t )0x007e6ae1;

/*Máscara de canales LoraWan, canales predeterminados 0-7*/
uint16_t userChannelsMask[6]={ 0x00FF,0x0000,0x0000,0x0000,0x0000,0x0000 };

/*Región LoraWan, seleccione en las herramientas IDE de arduino*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;

/*Se admiten LoraWan Clase, Clase A y Clase C*/
DeviceClass_t loraWanClass = CLASS_A;

/*el ciclo de trabajo de transmisión de datos de la aplicación. valor en [ms]. */
uint32_t appTxDutyCycle = 5000; //15000

/*OTAA or ABP*/
bool overTheAirActivation = true;

/*ADR abilitado*/
bool loraWanAdr = true;

/* Indica si el nodo está enviando mensajes confirmados o no confirmados */
bool isTxConfirmed = true;

/* Puerto de aplicación */
uint8_t appPort = 2;
uint8_t confirmedNbTrials = 4;

/*variables para el intervalo de tiempo de subida de datos */
unsigned long tiempo_uplink = 0;
unsigned long tiempo_send = 1000*60*5; // unidades en milisegundos 1000*60*5 (5minutos)
int ciclos = 0;

/*Variables para el sensor ultrasónico JSN-SR04T-V3.0*/
const int trig = 20; //Pin TRIG
const int echo = 19; //Pin ECHO
float altura_sensor = 100; //Representa la altura del sensor instalado en el vertedero
float duracion; //Variable usada para medir el tiempo de los pulsos
float nivel_H2O; //Esta variable toma la distancia del sensor
float nivel_total; //Esta variable incluye el nivel de agua total

/* Prepara la carga útil de la trama */
int16_t Rssi;
```

```

void setup() {
  Serial.begin(115200);           //Comunicación serial
  Mcu.setlicense(license);       //Contiene la información de la autorización para el MCU
  Mcu.begin();                   //Incluye la inicialización de hardware
  display.init();                //Inicializa la pantalla display.
  pinMode(trig, OUTPUT);         //Pin destinado al trig del ultrasónico
  pinMode(echo, INPUT);         //Pin destinado al echo del ultrasónico
  delay(4000);                   //Se agrega 4 segundos por el tiempo de encendido del
sensor

  deviceState = DEVICE_STATE_INIT; //inicializa el proceso LoRaWan

}

/*con el void loop podemos subir los datos cada cierto tiempo*/
void loop(){

  if (ciclos < 1) {
    appTxDutyCycle = tiempo_send; // los datos se envían cada 5 minutos
    nodofinal();

  }

  /* La siguiente función permite la medición de la distancia con el sensor ultrasónico */
}
void nivel_de_agua() {
  digitalWrite(trig, LOW);       //Inicia el pulso en LOW por 2uS
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);     //Se genera el pulso HIGH por 10us
  delayMicroseconds(10);
  digitalWrite(trig,LOW);       //Se termina el pulso
  duracion = pulseIn(echo, HIGH); //Guarda la duración del pulso

  nivel_H2O = (duracion*0.0343)/(2); //Convierte la duración del pulso a distancia.
  nivel_total = altura_sensor - nivel_H2O; //Se obtiene la altura del nivel de agua por
diferencia
}

/*Se utiliza para preparar un paquete de datos para ser transmitido*/
static void prepareTxFrame( uint8_t port )
{
  unsigned char *send;
  appDataSize =4;               //La cantidad de datos que se transmitirán es de
4 bytes.
  send = (unsigned char *)&nivel_total; //Esto es útil para enviar la información de
nivel_total como bytes individuales.
  appData[0] = send[0];
  appData[1] = send[1];
  appData[2] = send[2];
  appData[3] = send[3];

  Serial.println("Nivel: " + String(nivel_total) + " cm"); //Imprime la información en
puerto serial
  void informacion(); //Esta función imprime en el display la información tomada por el
sensor
}

/*Controla el proceso de transmisión de datos*/

```

```

void nodofinal() {
    switch(deviceState)

    {
    case DEVICE_STATE_INIT: //Inicia el proceso de lectura del sensor
    {
        nivel_de_agua();           //Lee el sensor.
        informacion();           //Muestra información en el display.

        #if(LORAWAN_DEVEUI_AUTO)
            LoRaWAN.generateDeveuiByChipID();
        #endif
            LoRaWAN.init(loraWanClass,loraWanRegion);
        break;
    }
    case DEVICE_STATE_JOIN: //Conecta el dispositivo con el Gateway
    {
        conectando();
        LoRaWAN.join();
        break;

    }
    case DEVICE_STATE_SEND: // Se realiza el envío de datos
    {
        subiendo_datos();
        prepareTxFrame(appPort);
        LoRaWAN.send();
        deviceState = DEVICE_STATE_CYCLE; // entra al ciclo de transmisión
        break;
    }
    case DEVICE_STATE_CYCLE: //programa el próximo ciclo de transmisión y se cambia el
    estado del dispositivo
    {
        txDutyCycleTime = appTxDutyCycle + randr( -APP_TX_DUTYCYCLE_RND, APP_TX_DUTYCYCLE_RND
    );
        LoRaWAN.cycle(txDutyCycleTime);
        deviceState = DEVICE_STATE_SLEEP;
        break;
    }
    case DEVICE_STATE_SLEEP: //El dispositivo entra en un estado de suspensión
    {

        LoRaWAN.sleep(loraWanClass);
        break;
    }

    default: //restablece el estado del dispositivo a DEVICE_STATE_INIT.
        //Esto es una medida de seguridad para asegurarse de que el dispositivo siempre
        inicie desde un estado conocido.
        {
            deviceState = DEVICE_STATE_INIT;
            break;
        }
        if (deviceState == DEVICE_STATE_CYCLE) {
        }
    }
}
}

```

```

/*Imprime en la pantalla del dispositivo la información tomada por el sensor*/
void informacion() {
    display.clear();
    display.setFont(ArialMT_Plain_24);
    display.drawString(30, 0, "EHA");
    display.setFont(ArialMT_Plain_16);
    display.drawString(0,30, "Nivel: " + String(nivel_total) + "cm");
    display.display();
    delay (3000);
}
/*Imprime en pantalla cuando el equipo este conectándose*/
void conectando() {
    display.clear();
    display.setFont(ArialMT_Plain_16);
    display.drawString(0,30, "Conectando...");
    display.display();
    delay (1000);
}
/*Imprime en pantalla la subida de datos*/
void subiendo_datos() {
    display.clear();
    display.setFont(ArialMT_Plain_16);
    display.drawString(0,30, "Subeindo Datos...");
    display.display();
    delay (1000);
}

```

### Anexo 3. Código implementado para la conexión de la estación EMA con el equipo Heltec.

```
#include "LoRaWan_APP.h"           //Libreria LoRaWan proporcionado por el fabricante
#include <Adafruit_BME280.h>       // Libreria del Sensor BME280

#define sensor_1_address 0x76
Adafruit_BME280 bme1;           // I2C device 1

uint32_t license[4]={...}; // incluye licencia del equipo
/*Parametros para la cotexion con el Gateway*/
uint8_t devEui[] = {...};
uint8_t appEui[] = {...};
uint8_t appKey[] = {...};
/* ABP para*/
uint8_t nwkSKey[] = {... };
uint8_t appSKey[] = {...};
uint32_t devAddr = ( uint32_t )0x007e6ae1;

/*Máscara de canales LoraWan, canales predeterminados 0-7*/
uint16_t userChannelsMask[6]={ 0x00FF,0x0000,0x0000,0x0000,0x0000,0x0000 };

/*Región LoraWan, seleccione en las herramientas IDE de arduino*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;

/*Se admiten LoraWan Clase, Clase A y Clase C*/
DeviceClass_t loraWanClass = CLASS_C;

/*El ciclo de trabajo de transmisión de datos de la aplicación. valor en [ms].*/
uint32_t appTxDutyCycle = 5000; //15000

/*OTAA or ABP*/
bool overTheAirActivation = true;

/*ADR abilitado*/
bool loraWanAdr = true;

/* Indica si el nodo está enviando mensajes confirmados o no confirmados */
bool isTxConfirmed = true;

/* Puerto de aplicación */
uint8_t appPort = 2;
uint8_t confirmedNbTrials = 4;

/*variables para el intervalo de tiempo de subida de datos */
unsigned long tiempo_uplink = 0;
unsigned long tiempo_send = 1000*60*5; // unidades en milisegundos 1000*60*5 (5minutos)
int ciclos = 0;

/*Varaibles para el sensor ultrasonico BME/BMP280*/
float temperatura;
float humedad;
float presion;
float lluvia;
const unsigned long debounceTime = 150; //Evita rebotes en la lectura del pluviometro
volatile unsigned long lastInterruptTime = 0; //Se utiliza junto con la variable pulsosLluvia para medir el tiempo entre interrupciones
```

```

const int pluviometroPin = 16;           //Define el pin donde se conecta el
pluviómetro
volatile unsigned long pulsosLluvia = 0; //Se declara una variable pulsos Lluvia que
se utiliza para contar los pulsos generados por el pluviómetro

void setup() {

    Serial.begin(115200);                //Comunicación serial
    Mcu.setlicense(license);             //Contiene la información de la autorización
para el MCU
    Mcu.begin();                         //Incluye la inicialización de hardware
    pinMode(pluviometroPin, INPUT_PULLUP); //Establece el pin como entrada con
resistencia de pull-up
    attachInterrupt(digitalPinToInterrupt(pluviometroPin), contarPulsos, FALLING);
//Configura una interrupción en el pin pluviometroPin para que llame a la función
contarPulsos() cuando el nivel del pin cambie de alto a bajo (FALLING).
    bool device_status = bme1.begin(sensor_1_address); //inicializa el sensor BME/BMP280
    if (!device_status) Serial.println("No se pudo encontrar un sensor BME280, verifique el
cableado o la dirección del dispositivo-1");
    Serial.println("-- TEST SENSOR BME280 --");
    delay(4000);                         //Esto se utiliza para dar tiempo al sensor BME280 para
inicializarse y estabilizarse después de la configuración.
    deviceState = DEVICE_STATE_INIT; //inicializa el proceso LoRaWAN

}

/*con el void loop podemos subir los datos cada periodo de tiempo*/
void loop(){

    if (ciclos < 1) {
        appTxDutyCycle = tiempo_send; // los datos se envían cada 5 minutos
        nodofinal();
    }
}

/*Esta función contarPulsos() es una función de interrupción. Se encarga de contar los
pulsos generados por un dispositivo, como un pluviómetro, cuando se detecta un cambio de
nivel en el pin de interrupción*/
void contarPulsos() {
    unsigned long currentInterruptTime = millis();
    if (currentInterruptTime - lastInterruptTime > debounceTime) {
        pulsosLluvia++;
        lastInterruptTime = currentInterruptTime;
    }
}

/*La función obtenerLluvia() se utiliza para calcular la cantidad de lluvia acumulada en
milímetros (mm) en función de los pulsos registrados por el pluviómetro. La conversión de
pulsos a mm se realiza multiplicando por 0.3, y luego se restablece el contador de pulsos
(pulsosLluvia) a cero después de obtener la cantidad de lluvia*/
float obtenerLluvia() {
    float lluviaCalculada = pulsosLluvia * 0.3; // Convierte pulsos a mm
    pulsosLluvia = 0; // Restablece el contador después de enviar los datos
    return lluviaCalculada;
}

/*Se utiliza para preparar un paquete de datos para ser transmitido*/
static void prepareTxFrame( uint8_t port )
{

```

```

float lluvia = obtenerLluvia(); //Se declara una variable local lluvia y se le asigna
el valor de la lluvia actual, que se obtiene llamando a la función obtenerLluvia()
/*Estas líneas leen los valores actuales de temperatura, humedad y presión utilizando un
sensor BME280.*/
temperatura = bme1.readTemperature();
humedad= bme1.readHumidity();
presion= bme1.readPressure() / 100.00;
/*Se declara un puntero send para convertir las variables de tipo float (temperatura,
humedad, presion, lluvia) en bytes individuales y luego se establece el tamaño del paquete
de datos appDataSize en 16 bytes.*/
unsigned char *send;
appDataSize =16;
send = (unsigned char *)&temperatura;
appData[0] = send[0];
appData[1] = send[1];
appData[2] = send[2];
appData[3] = send[3];

send = (unsigned char *)&humedad;
appData[4] = send[0];
appData[5] = send[1];
appData[6] = send[2];
appData[7] = send[3];

send = (unsigned char *)&presion;
appData[8] = send[0];
appData[9] = send[1];
appData[10] = send[2];
appData[11] = send[3];

send = (unsigned char *)&lluvia;
appData[12] = send[0];
appData[13] = send[1];
appData[14] = send[2];
appData[15] = send[3];
/*Estas líneas imprimen en el puerto serie los valores actuales de temperatura, humedad,
presión y lluvia.*/
Serial.println("Temp : " + String(temperatura) + " °C\t\t");
Serial.println("Humi : " + String(humedad) + " %\t\t");
Serial.println("Pres : " + String(presion) + " hPa\t");
Serial.println("lluv : " + String(lluvia) + " mm^2\t\t");
}

/*Controla el proceso de transmisión de datos*/
void nodofinal() {
switch(deviceState)

{
case DEVICE_STATE_INIT: //Inicia el proceso de lectura del sensor
{
#ifdef LORAWAN_DEVEUI_AUTO
LoRaWAN.generateDeveuiByChipID();
#endif
LoRaWAN.init(loraWanClass,loraWanRegion);
break;
}
case DEVICE_STATE_JOIN: //Conecta el dispositivo con el Gateway

```

```

{
    LoRaWAN.join();
    break;
}
case DEVICE_STATE_SEND: //Se realiza el envio de datos
{
    // Desactiva la interrupción antes de transmitir
    detachInterrupt(digitalPinToInterrupt(pluviometroPin));

    prepareTxFrame(appPort);
    //informacion();
    LoRaWAN.send();
    deviceState = DEVICE_STATE_CYCLE; // entra al ciclo de transmisión
    break;
}
case DEVICE_STATE_CYCLE: //programa el próximo ciclo de transmisión y se
cambia el estado del dispositivo
{
    // Desactiva la interrupción antes de transmitir
    detachInterrupt(digitalPinToInterrupt(pluviometroPin));
    prepareTxFrame(appPort);
    txDutyCycleTime = appTxDutyCycle + randr( -APP_TX_DUTYCYCLE_RND, APP_TX_DUTYCYCLE_RND
);
    LoRaWAN.cycle(txDutyCycleTime);
    deviceState = DEVICE_STATE_SLEEP;
    // Vuelve a activar la interrupción después de la transmisión
    attachInterrupt(digitalPinToInterrupt(pluviometroPin), contarPulsos, FALLING);

    break;
}

case DEVICE_STATE_SLEEP: //El dispositivo entra en un estado de suspensión
{
    LoRaWAN.sleep(loraWanClass);
    break;
}

default: //Restablece el estado del dispositivo a DEVICE_STATE_INIT.
//Esto es una medida de seguridad para asegurarse de que el dispositivo
siempre inicie desde un estado conocido.
{
    deviceState = DEVICE_STATE_INIT;
    break;
}
if (deviceState == DEVICE_STATE_CYCLE) {
}
}
}

```

#### Anexo 4. Código “Payload” usado en TTN para poder interpretar los datos

```
/*
La función bytesToInt(by): convierte un array de 4 bytes en un número entero. La variable by
representa el array de bytes que se quiere convertir. Utiliza desplazamientos de bits para
combinar los bytes y formar el número entero. Retorna el número entero resultante.*/
function bytesToInt(by) {
    f = by[0] | by[1]<<8 | by[2]<<16 | by[3]<<24;
    return f;
}

/*La función bytesToFloat(by): Esta función convierte un array de 4 bytes en un número de
punto flotante (float). Nuevamente, se utilizan desplazamientos de bits para combinar los
bytes y formar un número entero. Luego, se descomponen los bits para extraer el signo, el
exponente y la mantisa del número en notación de punto flotante. Utilizando estas partes, se
calcula el valor del número de punto flotante. Retorna el número de punto flotante
resultante.*/

function bytesToFloat(by) {
    var bits = by[3]<<24 | by[2]<<16 | by[1]<<8 | by[0];
    var sign = (bits>>>31 === 0) ? 1.0 : -1.0;
    var e = bits>>>23 & 0xff;
    var m = (e === 0) ? (bits & 0x7fffffff)<<1 : (bits & 0x7fffffff) | 0x800000;
    var f = sign * m * Math.pow(2, e - 150);
    return f;
}

/*La función Decoder(bytes, port):Esta es la función principal de decodificación que
procesará los datos recibidos. Finalmente, se crea un objeto decoded que contiene las
mediciones convertidas, y se retorna este objeto.*/
function Decoder(bytes, port) {
    i = 0;
    centigrados = bytesToFloat(bytes.slice(i,i+=4)).toFixed(2); // Redondea a 2 decimales
    porcentaje = bytesToFloat(bytes.slice(i,i+=4)).toFixed(2); // Redondea a 2 decimales
    pascales = bytesToFloat(bytes.slice(i,i+=4)).toFixed(2);
    cubicos = bytesToFloat(bytes.slice(i,i+=4)).toFixed(2);
    var decoded = {};
    decoded.Temperatura = parseFloat(centigrados); // Convierte a número de nuevo (opcional)
    decoded.Humedad = parseFloat(porcentaje); // Convierte a número de nuevo (opcional)
    decoded.Presion = parseFloat(pascales);
    decoded.Lluvia = parseFloat(cubicos) ;
    return decoded;
}
```

## Anexo 5. Código implementado Azure Funciones para filtrar y almacenar los datos

```
/* ("log") indica que la función ha sido llamada y recibe un arreglo de mensajes
IoTHubMessages.*/

module.exports = function (context, IoTHubMessages) {
    context.log("JavaScript eventhub trigger function called for message array:
    ${IoTHubMessages}");
    /* El código ajusta el objeto de fecha Local restando 300 minutos, Lo que podría ser un
    desplazamiento horario específico (UTC-5 en este caso) para obtener la hora Local.*/
    let fechaLocal = new Date();
    let desplazamientoHorarioMinutos = -300;
    fechaLocal.setMinutes(fechaLocal.getMinutes() + desplazamientoHorarioMinutos);
    /*Se crean dos objetos, outputNodo1 y outputNodo2, para almacenar datos de dos tipos
    diferentes de nodos.*/
    let outputNodo1 = {
        "Fecha_local_UTC-5": fechaLocal.toISOString(),
        "Temperatura": "",
        "Humedad": "",
        "Presion": "",
        "Lluvia": "",
    };
    let outputNodo2 = {
        "Fecha_local_UTC-5": fechaLocal.toISOString(),
        "Nivel_de_agua": "",
    };
    /*Se establecen hasDataNodo1 y hasDataNodo2 como falsas inicialmente, para determinar si
    se han recibido datos para cada uno de los tipos de nodos*/
    let hasDataNodo1 = false;
    let hasDataNodo2 = false;
    /*El código itera a través de los mensajes IoTHubMessages utilizando un bucle forEach. Para
    cada mensaje, se analiza su contenido en formato JSON y se verifica si contiene información
    relacionada con los datos de payload.*/
    /*hasDataNodo1 y hasDataNodo2 se establecen en true si se han procesado datos válidos para
    los nodos correspondientes.*/
    IoTHubMessages.forEach(message => {
        context.log("Processed message: ${message}");
        const messageObject = JSON.parse(message);

        let payload = messageObject.uplink_message &&
        messageObject.uplink_message.decoded_payload;

        if (payload) {
            if (payload.Temperatura || payload.Humedad || payload.Presion ||
            payload.lluvia) {
                outputNodo1.Temperatura = payload.Temperatura;
                outputNodo1.Humedad = payload.Humedad;
                outputNodo1.Presion = payload.Presion;
                outputNodo1.Lluvia = payload.Lluvia;
                outputNodo1.deviceId = messageObject.end_device_ids &&
                messageObject.end_device_ids.device_id;
                hasDataNodo1 = true;
            }

            if (payload.Nivel_de_agua) {
                outputNodo2.Nivel_de_agua = payload.Nivel_de_agua;
            }
        }
    });
}
```

```

        outputNodo2.deviceId = messageObject.end_device_ids &&
messageObject.end_device_ids.device_id;
        hasDataNodo2 = true;
    }
}
});
/*Finalmente, si se han recibido datos para outputNodo1, se almacenan en
context.bindings.outputDocument1, y si se han recibido datos para outputNodo2, se almacenan
en context.bindings.outputDocument2.*/
    if (hasDataNodo1) {
        context.bindings.outputDocument1 = outputNodo1;
        context.log("Data for Container1: ${JSON.stringify(outputNodo1)}");
    }
    if (hasDataNodo2) {
        context.bindings.outputDocument2 = outputNodo2;
        context.log("Data for Container2: ${JSON.stringify(outputNodo2)}");
    }
    context.done();
};

```

## Anexo 6. Configuración inicial de simulación de la red LoraWAN en OMNeT++

### [General]

```
network = LoRaNetworkTest      #Establece el nombre de La red simulada como
"LoRaNetworkTest".
**.maxTransmissionDuration = 5s #Establece La duración máxima de transmisión.
**.vector-recording = false    #Desactiva La grabación de vectores de resultados.

rng-class = "cMersenneTwister" # Configura el generador de números aleatorios como
"cMersenneTwister".
**.loRaGW[*].numUdpApps = 1    #Configura cada gateway LoRa para tener una aplicación UDP.
**.loRaGW[0].packetForwarder.localPort = 2000 #Establece el puerto local del gateway LoRa
en 2000.
**.loRaGW[0].packetForwarder.destPort = 1000
**.loRaGW[0].packetForwarder.destAddresses = "networkServer"
**.loRaGW[0].packetForwarder.indexNumber = 0

#Activa La evaluación de tasa de adaptación automática (ADR) en Los nodos y en el servidor.
**.evaluateADRinNode = true
**.evaluateADRinServer = true

**.networkServer.numApps = 1 #Configura el servidor de red LoRa para tener una aplicación.
**.networkServer.**.evaluateADRinServer = true
**.networkServer.app[0].typename = "NetworkServerApp" #Especifica el tipo de aplicación en
el servidor de red como "NetworkServerApp".
**.networkServer.app[0].destAddresses = "loRaGW[0]" #se establece que La aplicación envía
datos al gateway LoRa [0]
**.networkServer.app[0].destPort = 2000 #se establece que La aplicación envía
datos al gateway LoRa [0]
**.networkServer.app[0].localPort = 1000 #se establece que La aplicación envía
datos al gateway LoRa [0]
**.networkServer.app[0].adrMethod = "${"avg"}" #Define el método de adaptación de La
tasa (ADR) utilizado por esta aplicación

**.numberOfPacketsToSend = 0 #${numberOfPackets = 200..5000 step 200} #100 #obviously 0
means infinite number of packets
sim-time-limit = 1d #Establece el límite de tiempo de simulación en 1 día
simtime-resolution = -11 #El valor -11 se refiere a La potencia negativa de 10 que se
utiliza para definir La resolución
repeat = 1 #Configura La repetición de La simulación

**.timeToFirstPacket = exponential(1000s) #Configura el tiempo que tarda un nodo en enviar
el primer paquete como una distribución exponencial con una tasa de Llegada de 1000
segundos
**.timeToNextPacket = exponential(1000s) #Configura el tiempo entre el envío de dos
paquetes consecutivos para un nodo como una distribución exponencial con una tasa de
Llegada de 1000 segundos.
**.alohaChannelModel = false

**.numberOfNodes = 100 #Configura el número de nodos en La simulación en
100
#Establece Las dimensiones máximas del área de restricción (área de simulación)
**.constraintAreaMaxX = 4000.0m
**.constraintAreaMaxY = 2000.0m
**.initFromDisplayString = false #Desactiva La inicialización desde una cadena de
visualización para poder ejecutarla en La ventana

**.loRaNodes[*]**.initialX = uniform(0m, 2000m)
**.loRaNodes[*]**.initialY = uniform(0m, 1000m)
**.loRaNodes[*]**.initialLoRaSF = intuniform(7,12) #facto de dispersión
**.loRaNodes[*]**.initialLoRaBW = 125 kHz # ancho de banda
**.loRaNodes[*]**.initialLoRaCR = 4 # factor de corrección
**.loRaNodes[*]**.evaluateADRinNode = true
```

```

#gateway features
**.numberOfGateways = 1           #Establece el número de gateways LoRa
#Configura la ubicación inicial del primer gateway LoRa en las coordenadas (1000 metros,
500 metros).#
**.loRaGW[0]**.initialX = 1000.00m
**.loRaGW[0]**.initialY = 500.00m
**.LoRaGWNic.radio.iAmGateway = true

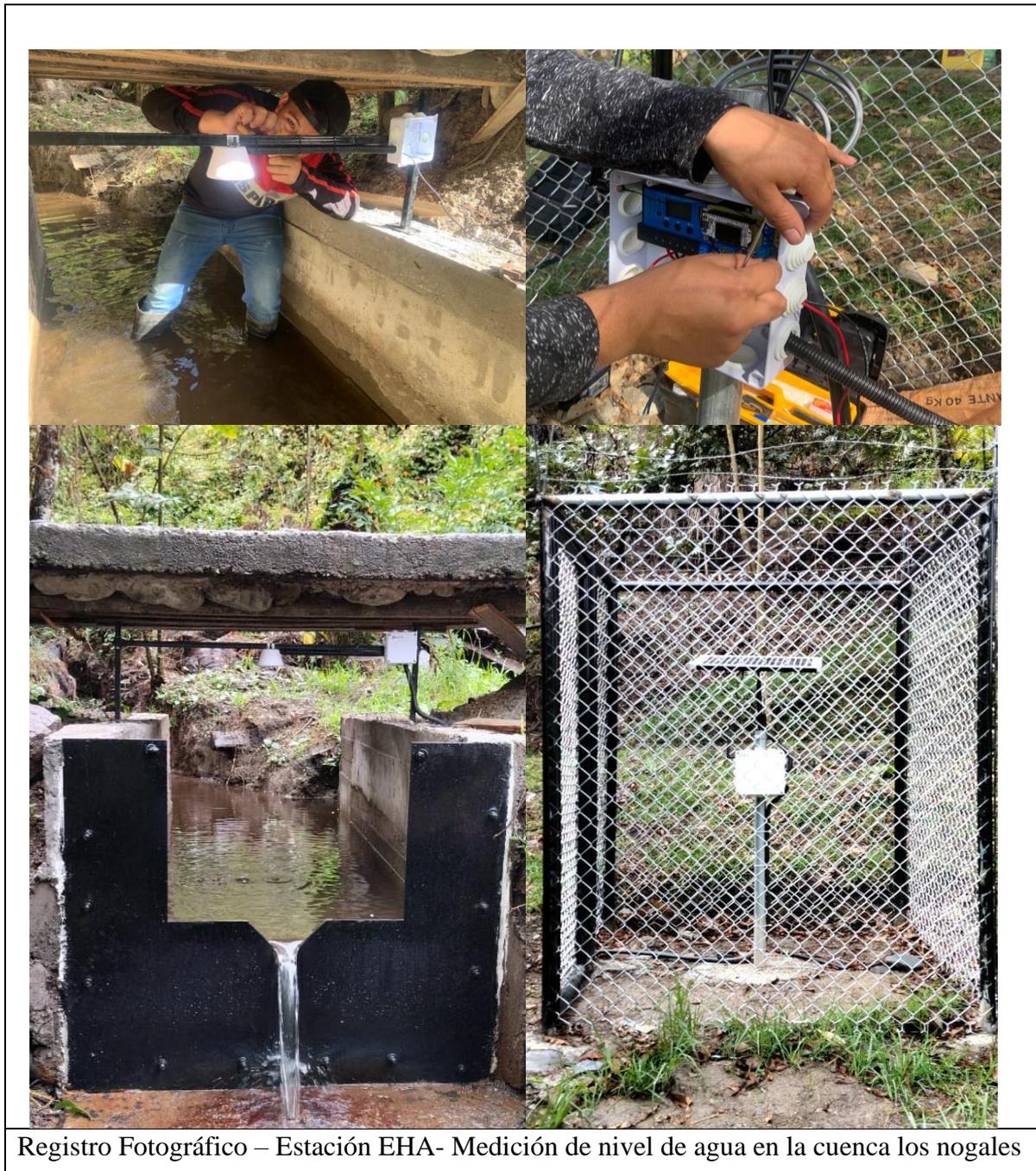
#características de consumo de energía de los dispositivos LoRa
**.loRaNodes[*].LoRaNic.radio.energyConsumer.typename = "LoRaEnergyConsumer"      #modela
el consumo de energía específico de los dispositivos LoRa.
**.loRaNodes[*]**.energySourceModule = "^.IdealEpEnergyStorage"                  #Esto
controla cómo se gestiona y almacena la energía en los nodos.
**.loRaNodes[*].LoRaNic.radio.energyConsumer.configFile =
xmldoc("energyConsumptionParameters.xml") #Configura un archivo XML
("energyConsumptionParameters.xml") para especificar parámetros relacionados con el consumo
de energía de la radio de los nodos LoRa.

**.sigma = 3.57 # entornos urbanos
**.constraintAreaMinX = 0m
**.constraintAreaMinY = 0m
**.constraintAreaMinZ = 0m
**.constraintAreaMaxZ = 0m
#Estas líneas desactivan la separación de las partes de transmisión y recepción en los
módulos de radio de la red LoRa.
LoRaNetworkTest.**.radio.separateTransmissionParts = false
LoRaNetworkTest.**.radio.separateReceptionParts = false

**.ipv4Delayer.config = xmldoc("cloudDelays.xml") #Configura un archivo XML
("cloudDelays.xml") para definir retrasos específicos relacionados con IPv4.
**.radio.radioMediumModule = "LoRaMedium"           #Especifica el módulo de medio de radio
utilizado en la simulación
**.LoRaMedium.pathLoss.typename = "LoRaLogNormalShadowing" #Define el tipo de modelo de
pérdida de propagación utilizado en el medio de radio LoRa como "LoRaLogNormalShadowing".
**.minInterferenceTime = 0s           #Establece el tiempo mínimo de interferencia en 0 segundos.
**.displayAddresses = true           #Habilita la visualización de direcciones en la simulación.

```

**Anexo 7.** Registro de la instalación de la red LoraWAN



Registro Fotográfico – Estación EHA- Medición de agua en la cuenca los nogales



Registro Fotográfico – Estación EMA- Medición de temperatura, humedad, presión atmosférica y precipitaciones



Registro Fotográfico – Gateway – instalación de Gateway LoRaWAN

## **Anexo 8.** Certificado de traducción del resumen al idioma inglés



Loja, 23 de noviembre de 2023

Lic. Karina Yajaira Martínez Luzuriaga

LICENCIADA EN CIENCIAS DE LA EDUCACIÓN MENCIÓN INGLÉS

### **CERTIFICO:**

Yo, Karina Yajaira Martínez Luzuriaga con cédula de identidad Nro. 1104902679, Licenciada en Ciencias de la Educación Mención Inglés por la Universidad Técnica Particular de Loja, con número de registro 1031-2022-2574017 en la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación, señalo que el presente documento es fiel traducción del idioma español al idioma inglés del resumen del Trabajo de Titulación denominado "DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE ESTACIÓN HIDROLÓGICA AUTÓNOMA PARA EL CENTRO DE INVESTIGACIÓN E INNOVACIÓN TECNOLÓGICA I<sup>2</sup>TEC DE LA UNIVERSIDAD NACIONAL DE LOJA" elaborado por el Sr. Edison Stalin Salinas Nalvay, con cédula de identidad Nro. 1150816989, estudiante egresado de la carrera de Ingeniería en Electrónica y Telecomunicaciones de la Universidad Nacional de Loja.



Lic. Karina Yajaira Martínez Luzuriaga

C.I. 1104902679

REGISTRO SENESCYT N°: 1031-2022-2574017