

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**IMPLEMENTACIÓN DE UN PROTOTIPO DE SENSADO PARA EL
PROCESO DE ADMINISTRACIÓN DE BASURA UTILIZANDO
LORAWAN EN EL CONTEXTO DE CIUDADES INTELIGENTES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
MAGISTER EN TECNOLOGÍAS DE INFORMACIÓN CON MENCIÓN EN
SISTEMAS DE INTERNET DE LAS COSAS**

KATHERINE JEANNETH ECHEVERRÍA AMORES

DIRECTOR: PhD. SORAYA LUCÍA SINCHE MAITA

Quito, septiembre 2023

AVAL

Certifico que el presente trabajo fue desarrollado por Katherine Jeanneth Echeverría Amores, bajo mi supervisión.



PhD. Soraya Lucía Sinche Maita
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Katherine Jeanneth Echeverría Amores, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.



Katherine Jeanneth Echeverría Amores

DEDICATORIA

A mi familia.

AGRADECIMIENTO

A mis padres por su apoyo incondicional en cada una de las etapas de mi vida, gracias por su compañía y por su amor desmedido.

A mis hermanos por ser una fuente de inspiración que me impulsa a seguir invirtiendo tiempo en mí.

A mi sobrino Andrés Salvador que se ha convertido en el motor de mi vida, te amo.

A Marco, por siempre estar a mi lado, por su apoyo y consejos.

A mis amigos, por siempre estar presentes.

A mi directora, la PhD. Soraya Sinche por su predisposición y ayuda en la realización de este proyecto de titulación. Muchas gracias por guiarme y ayudarme a cumplir una meta más en mi vida.

Y un agradecimiento eterno a la Escuela Politécnica Nacional que me ha permitido crecer en mi vida profesional y personal. Siempre estaré orgullosa de ser una politécnica.

ÍNDICE DE CONTENIDO

AVAL.....	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1. INTRODUCCIÓN.....	1
1.1 Pregunta de investigación.....	1
1.2 Objetivo General	2
1.3 Objetivos Específicos.....	2
1.4 Alcance	2
1.5 Marco Teórico	2
1.5.1. Sensores	4
1.5.1.1. Sensor de infrarrojo	4
1.5.1.2. Sensor de ultrasonido.....	5
1.5.2. Sistemas embebidos.....	6
1.5.3. Especificación LoRaWAN	7
1.5.3.1. LoRa	7
1.5.3.2. LoRaWAN.....	8
1.5.3.2.1. Arquitectura LoRaWAN	9
1.5.3.2.2. Tipos de mensajes	10
1.5.3.2.3. Seguridad LoRaWAN	11
2. METODOLOGÍA.....	13
2.1. Diseño del Prototipo	13
2.1.1. Adquisición de Datos	13
2.1.1.1 Cálculo de la distancia con el sensor de ultrasonido HC-SR04.....	14
2.1.1.2 Cálculo del porcentaje de llenado	15
2.1.2. Red de comunicaciones LoRaWAN.....	15

2.1.2.1. Dispositivo final LoRaWAN.....	15
2.1.2.2. Gateway LoRaWAN RAK 7243	17
2.1.3. Servidores de Red y Aplicación	18
2.2. Implementación del Prototipo	19
2.2.1. Configuración para el sensor de ultrasonido.....	19
2.2.2. Configuración módulo MKR 1310 WAN.....	22
2.2.2.1. Registro del nodo	24
2.2.2.2. Activación OTAA	26
2.2.2.3. Cambio de tipo de variable	28
2.2.3. Registro del Gateway.....	29
2.2.4. Integración con Things Board	31
3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	37
3.1. Resultados.....	37
3.2. Conclusiones	43
3.3. Recomendaciones	44
4. REFERENCIAS BIBLIOGRÁFICAS	45

RESUMEN

Los problemas de basura en el contexto de las ciudades inteligentes son una preocupación importante. Aunque las ciudades inteligentes están diseñadas para mejorar la calidad de vida de los residentes al integrar tecnología y sistemas avanzados, también presentan desafíos relacionados con la gestión de residuos.

La acumulación de basura en las calles y áreas públicas puede llevar a problemas de contaminación y tener un impacto negativo en el medio ambiente y la calidad de vida de los residentes. La basura mal gestionada puede atraer plagas, generar malos olores y contaminar el suelo y el agua.

El presente trabajo de titulación tiene como objetivo implementar un prototipo de sensado para el proceso de administración de basura utilizando LoRaWAN en el contexto de ciudades inteligentes.

El desarrollo del prototipo incluye cuatro etapas. En la primera etapa se realiza la configuración del módulo de Arduino LoRaWAN para que reciba los valores de distancia enviados por el sensor de ultrasonido.

La segunda etapa se realiza el establecimiento de la comunicación LoRaWAN entre el módulo de Arduino y la nube de The Things Network mediante el Gateway RAK 7243, que se encuentra configurado como una puerta de enlace pública.

En la tercera etapa del desarrollo del prototipo se realizan las configuraciones en The Things Stack para el registro del Gateway y de la aplicación, así como la integración con un dashboard para presentar los datos enviados.

Finalmente, se presentan los datos, recopilados en tiempo real, mediante la plataforma Things Board.

PALABRAS CLAVE: Ciudades Inteligentes, IoT, LoRaWAN, LoRa, The Things Network, Things Board.

ABSTRACT

Garbage management issues in the context of smart cities are a major concern. Although smart cities are designed to improve the quality of life for residents by integrating advanced technology and systems, they also present challenges related to waste management.

The accumulation of garbage in the streets and public areas can lead to pollution problems and have a negative impact on the environment and the quality of life of residents. Poorly managed garbage can attract pests, generate bad odors, and contaminate the soil and water.

The objective of this degree work is to implement a sensing prototype for the garbage management process using LoRaWAN in the context of smart cities.

The development of the prototype includes four stages. In the first stage, the configuration of the Arduino LoRaWAN module is carried out so that it receives the distance values sent by the ultrasound sensor.

The second stage is the establishment of LoRaWAN communication between the Arduino module and The Things Network cloud through the RAK 7243 Gateway, which is configured as a public gateway.

In the third stage of the development of the prototype, the configurations are made in The Things Stack for the registration of the Gateway and the application, as well as the integration with a dashboard to present the data sent.

Finally, the data is presented, collected in real time, through the Things Board platform.

KEYWORDS: Smart Cities, IoT, LoRaWAN, LoRa, The Things Network, Things Board

1. INTRODUCCIÓN

El uso de sistemas IoT (*Internet of Things*) en el contexto de Ciudades Inteligentes brinda mejoras en movilidad, seguridad, energía, servicios militares, medicina, gestión de la ciudad, etc., ya que permite la obtención de gran cantidad de datos, los cuales se recopilan y analizan para generar información relevante.

La administración de residuos sólidos representa un gran aporte al desarrollo de las ciudades acercándolas al concepto de *Smart Cities*, ya que mediante una red de sensores distribuidos por sector se puede optimizar recursos y realizar un trabajo más efectivo. La sostenibilidad ambiental es uno de los puntos más críticos en la actualidad; además los cambios climáticos han sido cada vez más devastadores para diferentes partes del mundo por lo que un sistema de manejo de desechos se considera importante.

A pesar de que existen varias tecnologías que pueden dar soporte al transporte de información en este campo, LoRaWAN es una de las más utilizadas, ya que permite la conexión a TTN (*The Things Network*), la cual ofrece herramientas y una red global de comunicación para el desarrollo de aplicaciones IoT. Además, cuenta con *The Things Stack* que actúa como un servidor de red LoRaWAN, componente crítico para cualquier solución LoRaWAN, que gestiona de forma segura las aplicaciones, los dispositivos finales y los gateways.

LoRaWAN es una tecnología que se basa, a nivel de capas inferiores, en LoRa (*Long Range*) permitiendo comunicaciones de largo alcance entre 15 y 20 kilómetros en una zona rural y entre 2 y 8 kilómetros en zona urbana [1]. LoRa posibilita una comunicación segura de extremo a extremo con velocidades de datos bajas y con bajo consumo de energía.

En este contexto se plantea implementar un prototipo para sensar el llenado de los contenedores de residuos sólidos, que permita alertar al sistema de administración de basura que se debe realizar el proceso de recolección, antes de que los contenedores lleguen a desbordarse, donde se utilizará como tecnología de comunicaciones LoRaWAN.

1.1 Pregunta de investigación

Un prototipo de gestión de residuos sólidos instalado en contenedores, ¿Ayudaría en el modelamiento de procesos de optimización de recursos, durante la recolección de basura, dentro de municipios?

1.2 Objetivo General

Implementar un prototipo para el proceso de administración de basura utilizando LoRaWAN en el contexto de ciudades inteligentes.

1.3 Objetivos Específicos

- Revisar los fundamentos de sistemas IoT y de la tecnología LoRaWAN.
- Diseñar los módulos que intervendrán en el prototipo.
- Implementar los módulos siguiendo el diseño planteado.
- Validar el correcto funcionamiento del prototipo implementado.

1.4 Alcance

En este trabajo de titulación se plantea la implementación de un prototipo para sensar el llenado de contenedores de basura, en base a sensores de ultrasonido. La información se envía a un servidor en la nube TTN, utilizando la tecnología LoRaWAN, un servidor de red que se encontrará alojado en la nube.

Se implementa un prototipo donde, para el proceso de sensado, se utiliza un sensor de ultrasonido que es el encargado de la detección del nivel de llenado del contenedor, el cual está conectado a un sistema embebido (ARDUINO). Estos datos son enviados a través de la tecnología LoRaWAN. Se utiliza una placa Arduino MKR WAN 1310 que combina la funcionalidad de la conectividad MKR Zero y LoRa. Esta placa se basa en Atmel SAMD21 y un módulo Murata CMWX1ZZABZ Lo-Ra. Además, posee una potencia computacional de 32 bits similar a la placa MKR ZERO [2].

En el caso de la tecnología LoRaWAN se dispone de acceso gratuito a la nube TTN. Para este proyecto se ha considerado utilizar un gateway propio debido a que en los existentes en la ciudad de Quito no garantiza una conexión continua durante el desarrollo del prototipo. El servidor de red se configura en la nube de TTN.

Este proyecto tendrá un producto final demostrable.

1.5 Marco Teórico

IoT en el contexto de Ciudad Inteligente brinda nuevas alternativas para implementar servicios que proporcionen una convivencia cómoda y con un eficiente uso de recursos.

Uno de los problemas más desafiantes en el desarrollo de ciudades inteligentes es manejo de residuos sólidos. Alrededor del 30% del presupuesto total de un municipio se consume para la gestión de residuos, ya que cuenta con pocos recursos y un manejo inadecuado de estos residuos [3]. Mediante la implementación las tecnologías IoT, estos recursos podrían ser utilizados con mayor eficiencia, generando un ahorro significativo a los gobiernos.

Los sistemas IoT y la tecnología de computación en la nube ofrecen una posibilidad de automatizar procesos mediante sistemas controlados por algoritmos que cambiarán la forma en que se realiza la administración de residuos sólidos. En varios municipios alrededor del mundo ya se ha considerado la implementación de sistemas de sensado de basura.

En la Universidad de Keio, Japón se desarrolló *Cruisers*, una plataforma asociada de detección de automóviles para ciudades inteligentes, que se desarrolla con el apoyo de varios sectores como [4]:

- Los camiones de recolección de basura se usan como vehículos anfitriones para colocar los sensores.
- Se emplea una red de comunicación celular 3G para entregar inalámbricamente los datos detectados a los servidores.
- Se utilizan servidores proxy para convertir el formato de los datos detectados a un formato común según sus necesidades. Esta plataforma se ha implementado en veinticuatro camiones de recolección de basura en la ciudad de Fujisawa, es decir, casi 1/4 de la gama completa de dichos camiones dentro de la ciudad. La aplicación asociada de iOS también se desarrolla para demostrar el método de detección y, por lo tanto, el espacio ocupado.

Otro trabajo presentado en el AJAST (Asian Journal of Applied Science and Technology) [5] se centra en los algoritmos de optimización para la gestión de recolección de residuos municipales. Se considera la infraestructura IoT y las redes de sensores existentes, ya implementadas en Francia, donde la comunicación se basa en la tecnología Sig Fox [5]. Estos sensores están conectados a la infraestructura principal construida en la ciudad, para que puedan proporcionar el conjunto de datos para la gestión de residuos. Esta red puede utilizar tecnologías LPWAN como Sig Fox, LoRa WAN, Weightless, Link Labs, Nwave o diferentes tecnologías que crean una infraestructura de comunicación de bajo consumo y bajo costo.

Actualmente, el principal problema con respecto a la recolección de desechos es que los camiones de basura funcionan de manera ineficaz. Las ciudades están estructuradas de formas diferentes, lo que provoca que algunos contenedores lleguen a llenarse a mayor velocidad que otros, y por lo cual deben ser vaciados con mayor frecuencia. LoRaWAN ofrece una gran variedad de aplicaciones en las que se utilizan dispositivos finales de bajo consumo de energía y largo alcance, lo que facilita la implementación de sensores de ultrasonido conectados con esta tecnología alrededor de la ciudad que puede dar soporte a un sistema de administración de residuos sólidos.

1.5.1. Sensores

En la construcción de ciudades inteligentes el uso de dispositivos que ayuden en la detección, sensado y reelección de datos es primordial para crear aplicaciones que ayuden a mejorar la forma de vida de las personas. Mediante los sensores se puede adquirir información en cualquier momento y en cualquier lugar. Así, la red participa activamente en la creación de un entorno inteligente.

Las redes de sensores prometen revolucionar la detección en una amplia gama de dominios de aplicación. Esto se debe a su confiabilidad, precisión, flexibilidad, rentabilidad y facilidad de implementación [6].

Existen varios tipos de sensores que recopilan información y ofrecen un monitoreo continuo, en el mercado se encuentran los sensores ópticos, magnéticos, de luz, humedad, posición, distancia, entre otros.

Los sensores de distancia, que son los sensores evaluados para el desarrollo del prototipo, proporcionan información de la distancia exacta entre dos puntos. Entre los más conocidos se tiene el infrarrojo, que determina la distancia mediante la emisión y recepción de radiación, y el sensor de ultrasonido, que se basa su sistema en la emisión y recepción de ondas ultrasónicas.

1.5.1.1. Sensor de infrarrojo

El sensor de infrarrojo está diseñado para detectar formas, colores y diferencias de superficies; a más de clasificar y posicionar objetos [7]. Se basa en el método de triangulación, que consiste en determinar el ángulo de incidencia del haz de luz infrarroja reflejado para calcular la distancia, sin la interferencia de la luz del ambiente.

En la Figura 1.1 se muestra como el sensor está formado por un LED IR que emite una señal analógica, que cambia dependiendo de la distancia entre el sensor, el objeto y un

detector de luz o PSD (*Position Sensing Device*). Cuando el haz de luz es reflejado por un objeto, alcanzará el detector de luz y se formará un punto óptico en la PSD [8].

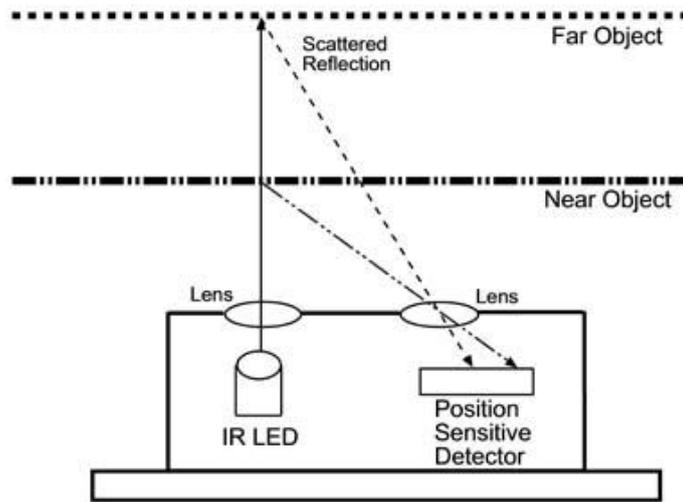


Figura 1.1. Diagrama de funcionamiento del sensor infrarrojo [8]

1.5.1.2. Sensor de ultrasonido

Este sensor se utiliza para determinar la distancia de un objeto en el rango de 2 a 400 cm., consta de dos transductores, un emisor y un receptor piezoeléctricos. El emisor emite 8 pulsos de ultrasonido (40KHz) luego de recibir la orden en el pin TRIGGER como se presenta en la Figura 1.2; el pulso de sonido ultrasónico se propaga en todas las direcciones desde el sensor. Estas ondas viajan a una velocidad constante, generalmente en el aire a aproximadamente 343 metros por segundo. Cuando las ondas ultrasónicas encuentran un objeto en su camino, parte del sonido es reflejado o dispersado por el objeto. El sensor de ultrasonido detecta el eco de vuelta de estas ondas reflejadas, es decir el ECHO cambia a alto [9].

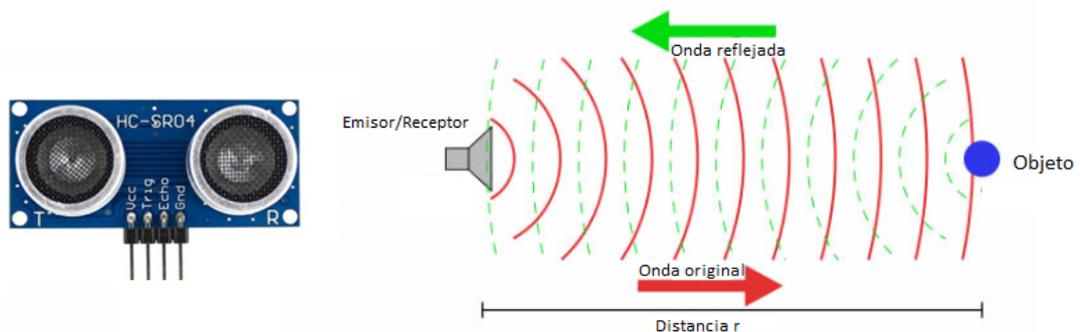


Figura 1.2. Funcionamiento del sensor de ultrasonido [9]

1.5.2. Sistemas embebidos

Los sistemas embebidos están desarrollados bajo un hardware y software que le permitirán desempeñar funciones determinadas. Sus componentes se encuentran integrados en una placa como se muestran en la Tabla 1.1. El procesamiento central del sistema consta de un microprocesador que incluye interfaces de entrada/salida, así como una memoria en el mismo chip [10].

Tabla 1.1. Componentes básicos de un sistema embebido [10]

Componente	Descripción
Procesador	Cerebro del sistema embebido, realiza las funciones de procesamiento y control. Puede ser un microcontrolador de propósito general o un microprocesador más potente dependiendo de los requisitos de la aplicación.
Fuente de alimentación	Los sistemas embebidos requieren una fuente de energía para funcionar. Puede ser una batería, una fuente de alimentación externa o una combinación de ambas. Es importante considerar los requisitos de consumo de energía y la eficiencia energética en el diseño del sistema.
Memoria	Los sistemas embebidos suelen incluir diferentes tipos de memoria para almacenar programas, datos y variables temporales. Esto puede incluir memoria de programa (como Flash ROM), memoria de datos (como RAM) y memoria no volátil (como EEPROM).
Entradas/Salidas	Permiten la interacción entre el sistema embebido y el entorno externo. Algunos ejemplos comunes de periféricos de E/S son puertos GPIO (Entrada/Salida de Propósito General), UART (<i>Universal Asynchronous Receiver/Transmitter</i>), I2C (Inter-Integrated Circuit), ADC (Convertidor Analógico-Digital), DAC (Convertidor Digital-Analógico) y SPI (<i>Serial Peripheral Interface</i>).

Las capacidades de comunicación inalámbrica han ampliado enormemente el espacio de aplicaciones con sistemas embebidos. Las funciones de estos sistemas como la adquisición, el procesamiento y la entrega de datos son fundamentales para nuevas propuestas de ciudades inteligentes; por ejemplo, en la recolección de residuos sólidos se

puede utilizar módulos de Arduino LoRAWAN para monitorizar los datos en tiempo real, lo que garantizaría una toma de decisiones basados en la evidencia presentada.

1.5.3. Especificación LoRaWAN

LoRaWAN es un protocolo de red LPWA (*Low Power Wide-Area*), diseñado para conectar a Internet de forma inalámbrica dispositivos finales que funcionan con baterías. Se centra en los requisitos de IoT como servicios de comunicación bidireccional, plataformas de gestión de datos, seguridad, movilidad y localización [11].

La especificación define los estándares y protocolos que rigen la comunicación entre los dispositivos IoT y su infraestructura de red. Como se presenta en la Figura 1.3, toma en consideración aspectos importantes como: capas del protocolo, tipos de dispositivos, tipos de mensajes, arquitectura, entre otros.

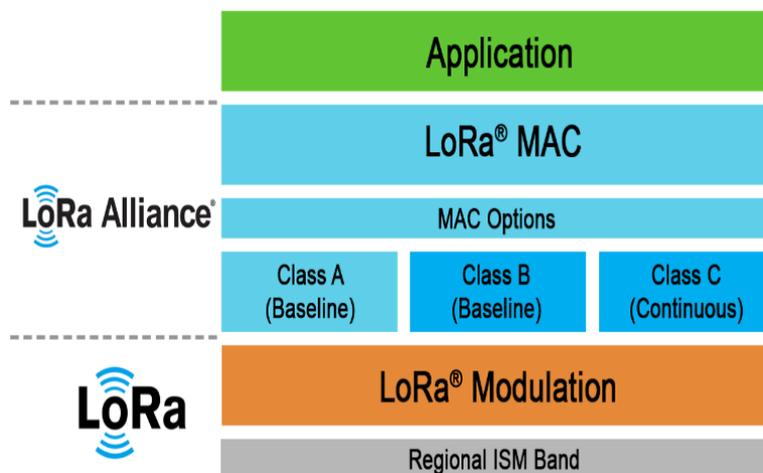


Figura 1.3. Stack de protocolos de la especificación LoRaWAN [11]

1.5.3.1. LoRa

LoRa es una tecnología de capa física, el resto de capas de los protocolos de red se deben cubrir por otras especificaciones como es el caso de LoRaWAN [11].

LoRa se basa en una técnica capaz de modular redes LPWA basado en la técnica CSS (*Chirp Spread Spectrum*), permitiendo la comunicación de largo alcance. Opera en diferentes bandas de frecuencia como EU686 MHz, US915 MHz, CN470 MHz. Incluye un ancho de banda escalable (125kHz, 250kHz o 500kHz). Según el ancho de banda y la SNR (*signal-to-noise ratio*), se elige un factor de dispersión variable para expandir la señal y hacerla más resistente al ruido [12].

El módulo LoRa Murata CMWX1ZZABZ opera con esta tecnología, cuya especificación se detalla en el Anexo A, el cual forma parte del nodo MKR WAN 1310, lo que permite aprovechar las ventajas de LoRa en aplicaciones IoT.

1.5.3.2. LoRaWAN

Es una tecnología de red de largo alcance, maneja bajos volúmenes de datos, ofrece comunicaciones bidireccionales y está pensada para dispositivos con bajo consumo de energía.

LoRaWAN se implementa en una topología de estrella, como se presenta en la Figura 1.4, en la que los nodos se comunican directamente con una estación base centralizada, también llamada *Gateway*. Los *Gateways* están conectados al servidor a través de conexiones IP y actúan como un puente transparente, simplemente convirtiendo paquetes de RF en paquetes de IP y viceversa [12].

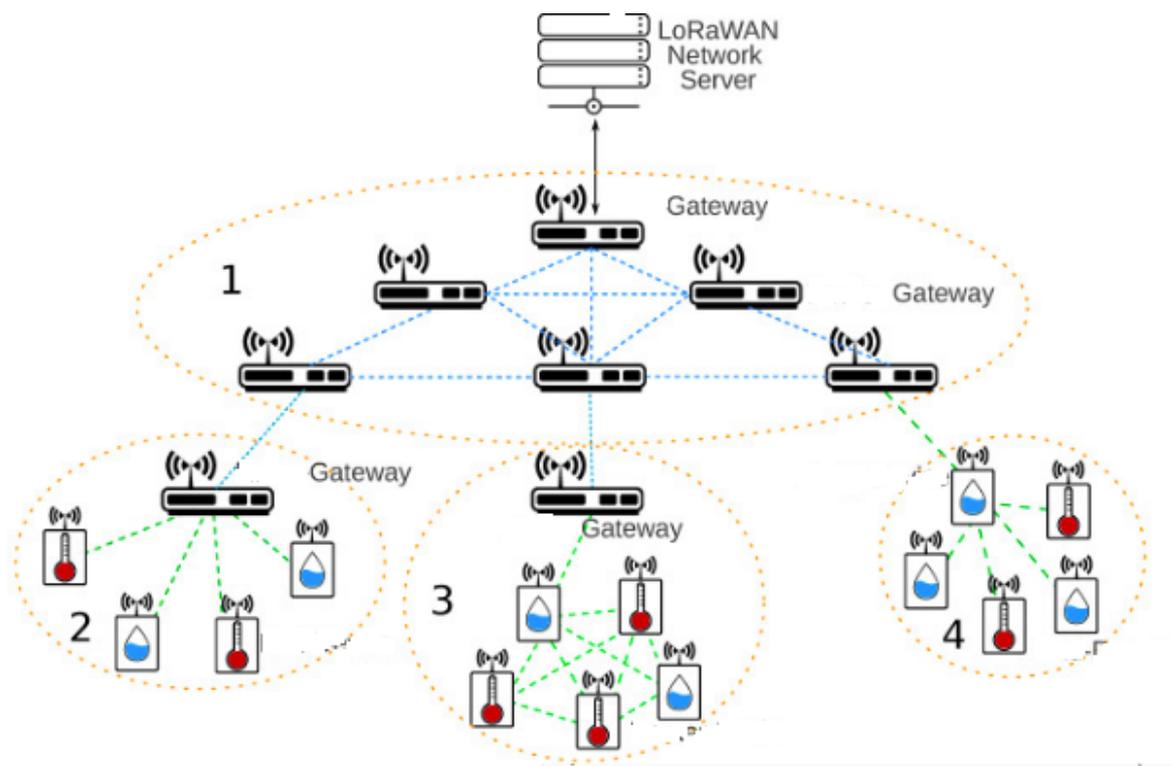


Figura 1.4. Topología LoRaWAN [13]

LoRaWAN ofrece una gran variedad de aplicaciones, las cuales se definen de acuerdo con el tipo de dispositivo final que utilizan. Estos dispositivos finales pueden ser de tres clases:

- **Clase A:** Dispositivos de menor potencia

El dispositivo final es quien inicia la comunicación enviando datos hacia el Gateway, puede entrar en modo de suspensión dependiendo del requerimiento de las aplicaciones, lo que hace que los dispositivos tengan el más bajo consumo de energía.

- **Clase B:** dispositivos finales con latencia determinista

En los dispositivos clase B, además de la comunicación iniciada como en clase A, se pueden programar ventanas de recepción, es decir en enlace descendente. Esta ventana tiene 128 segundos en donde el servidor sabrá que el dispositivo final está escuchando.

- **Clase C:** Latencia más baja

Mientras los dispositivos están transmitiendo datos la ventana de enlace descendente se encuentra cerrada, caso contrario la ventana permanece abierta, por lo que, permite que el servidor inicie la transmisión en cualquier momento, lo que proporciona una baja latencia.

1.5.3.2.1. Arquitectura LoRaWAN

La arquitectura LoRaWAN cuenta con dispositivos finales, gateways, servidor de red y el servidor de aplicación, como se muestra en la Figura 1.5.

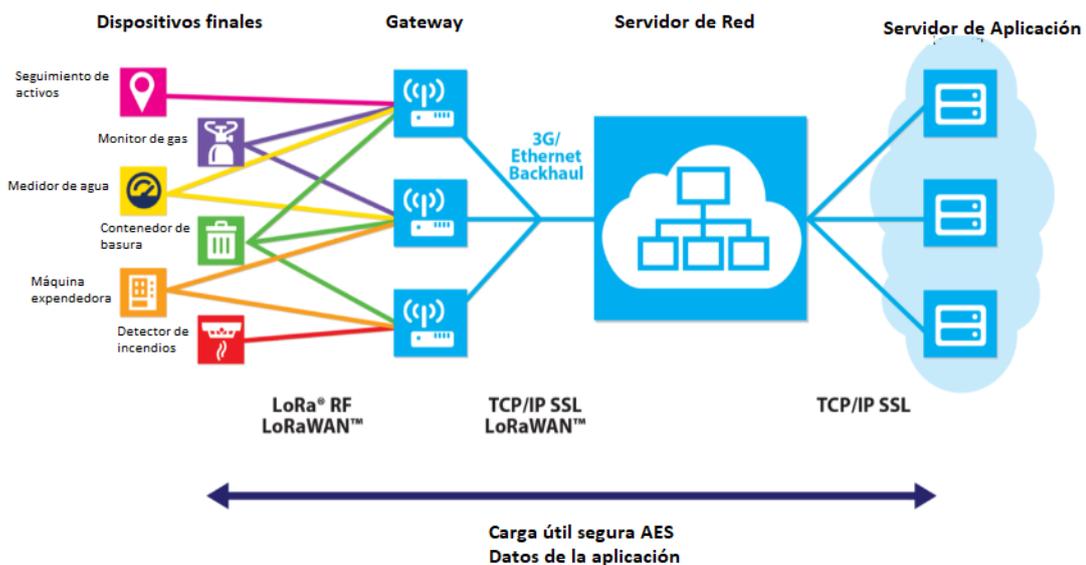


Figura 1.5. Arquitectura LoRaWAN [12]

- **Dispositivos finales:** sensores o actuadores, que detecta la información y la envía inalámbricamente al Gateway utilizando la tecnología LoRa y siguiendo el protocolo LoRaWAN. Pueden ser dispositivos de bajo consumo de energía y están diseñados para transmitir datos de manera eficiente y confiable.
- **Gateways:** son equipos intermediarios entre los nodos y la infraestructura de red. Reciben los mensajes enviados por los dispositivos finales y los retransmiten al servidor de red central. Las puertas de enlace están equipadas con antenas LoRa para comunicarse con los dispositivos finales y pueden cubrir un área geográfica determinada.
- **Servidor de red:** es el componente central de la red LoRaWAN. Recibe los mensajes enviados por los dispositivos finales a través de los gateways y los procesa. El servidor de red administra la asignación de canales de comunicación, autentica los dispositivos, gestiona el enrutamiento de mensajes y realiza la configuración de la red.
- **Servidor de aplicación:** es responsable de recibir y procesar los datos provenientes de los dispositivos finales. Puede realizar diversas tareas, como almacenar datos, analizarlos, generar informes, activar acciones o interactuar con otros sistemas. El servidor de aplicaciones proporciona la interfaz de usuario o la interfaz de programación de aplicaciones para interactuar con los datos del dispositivo final.

1.5.3.2.2. Tipos de mensajes

La especificación LoRaWAN define varios tipos de mensajes, incluyendo mensajes de solicitud de activación, mensajes de datos *uplink* y *downlink*, mensajes de confirmación y mensajes de solicitud de *downlink*.

En la Tabla 1.2 se observan los tipos de mensajes que se intercambian en la comunicación LoRaWAN.

Tabla 1.2. Tipos de mensajes LoRaWAN [14]

Tipo de mensaje	Descripción
<i>Join-request</i>	Mensaje de enlace ascendente, utilizado por el procedimiento de la activación OTAA.
<i>Unconfirmed Data Up</i>	Mensaje de enlace ascendente, no requiere confirmación.

Tipo de mensaje	Descripción
Join-accept	Mensaje de enlace descendente, utilizado por el procedimiento de la activación OTAA.
Unconfirmed Data Down	Mensaje de enlace descendente, no requiere confirmación.
Confirmed Data Up	Mensaje de enlace ascendente, requiere confirmación.
Confirmed Data Down	Mensaje de enlace descendente, requiere confirmación.
RFU	Reservado para uso futuro.
Proprietary	Se utiliza para implementar formatos de mensajes no estándar.

1.5.3.2.3. Seguridad LoRaWAN

Los dispositivos LoRaWAN utilizan cifrado AES de 128 bits y un identificador único global (DevEUI) basado en EUI (*Extended Unique Identifier*). Estos se utilizan en la autenticación del nodo. Además, especifica claves de seguridad: NwkSKey (*Network Session Key*), AppSKey (*Application Session Key*) y AppKey (*Application Key*).

Las claves se pueden activar por personalización (ABP) en la línea de producción o durante la puesta en marcha, o se pueden activar por aire (OTAA) en el campo. OTAA permite que los dispositivos se vuelvan a registrar si es necesario [16].

- **ABP (*Activation by Personalization*):** en el Gateway se debe incluir los parámetros DevAddr, NwkSKey y AppSKey.
- **OTAA (*Over-the-Air activation*):** En la activación OTAA el dispositivo final y la red tienen la clave AppKey, mediante esta clave se derivan las claves NwkSKey y AppSKey. En el Gateway se debe colocar los parámetros DevEui, AppEui, AppKey.
 - DevEui: ID global del dispositivo, lo asigna el fabricante.
 - AppEui: identificador global de la aplicación, asociado al servidor de aplicaciones. AppKey: clave de cifrado compartida entre los nodos finales y el servidor de red. Se utiliza para proteger los mensajes de aplicación enviados entre el nodo y el servidor de aplicaciones. La AppKey se utiliza para cifrar y descifrar los datos transmitidos, asegurando que solo el

dispositivo final y el servidor de aplicaciones puedan leer y procesar los datos.

2. METODOLOGÍA

En este capítulo se muestra los componentes utilizados para desarrollar el prototipo para el sensado de llenado de contenedores de basura, desde el sensado hasta la observación en tiempo real de los datos obtenidos.

2.1. Diseño del Prototipo

Para el diseño del prototipo se ha considerado el diagrama de bloques que se presenta en la Figura 2.1, donde se incluye:

- Adquisición de Datos.
- Red de Comunicaciones LoRaWAN.
- Servidores de Red y Aplicación.
- Presentación de datos.

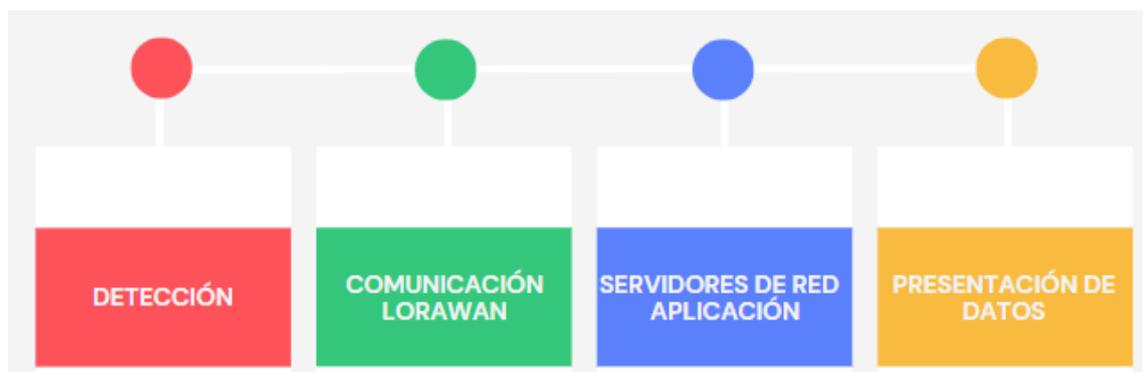


Figura 2.1. Diagrama de bloques del prototipo de sensado de basura

2.1.1. Adquisición de Datos

En el proceso de adquisición de datos de llenado del contenedor de basura se lo realiza en base al cálculo de la distancia.

Dentro de la gama de sensores de distancia, los más comunes son los sensores ópticos y de ultrasonido. Para la elección del tipo de sensor a utilizar, se han analizado cuáles son las características que podría ayudar a la detección de residuos sólidos dentro de los contenedores de basura.

Tabla 2.1. Características de sensores infrarrojo y de ultrasonido

Características	Sensores Infrarrojos	Sensores de ultrasonido
Detección	Con línea de visión directa	Sin línea de visión directa
Sensibilidad	Con altos niveles de movimiento	Con nivel de movimiento muy bajo
Influencia por elementos duros (paredes, humo, polvo, neblina, luz del día)	Afectado	Inmune
Rango de medición	Pequeño	Grande
Objeto detectable	Detección afectada por materiales/colores de objeto	Detección no afectada por materiales/colores del objeto
Distancia de detección	Corta	Larga

Con base a la Tabla 2.1 se concluye que, los sensores que se afecten por el tipo de material y el color del objeto no es una opción viable para el prototipo, ya que, al ser un contenedor de basura no se podría realizar esta clasificación antes de que el usuario deposite los residuos. Por esta razón se utilizará un sensor de ultrasonido.

2.1.1.1 Cálculo de la distancia con el sensor de ultrasonido HC-SR04

Para el cálculo de la distancia desde el sensor a una superficie, se considera que el sensor tiene un ángulo de detección de hasta 15°, ángulo de medición de 30° y un rango de distancia de 2 a 400 cm [17].

Por otro lado, se conoce que la velocidad del sonido es 343 m/s en condiciones de temperatura 20 °C, 50% de humedad, y presión atmosférica a nivel del mar. Transformando unidades se obtiene la ecuación 2.1.

$$\text{Speed of Ultrasonics Wave} = 343 \frac{m}{s} \times 100 \frac{cm}{m} \times \frac{1}{1000000} \frac{s}{\mu s} = \frac{1}{29.2} \frac{cm}{\mu s} \quad (2.1)$$

De donde se deduce que el sonido tarda 29,2 microsegundos en recorrer un centímetro. La distancia a partir del tiempo se mide entre la emisión y recepción del pulso mediante la ecuación 2.2.

$$Distance (cm) = \frac{Tiempo (\mu s)}{29.2} \quad (2.2)$$

Este cálculo se puede realizar mediante la librería NewPing.h que ha mejorado la precisión en la medida de distancia, ya que tiene nuevas bibliotecas ultrasónicas y de ping disponibles.

2.1.1.2 Cálculo del porcentaje de llenado

Para el cálculo del porcentaje de llenado, se considera la distancia máxima entre el sensor de ultrasonido y la base del contenedor, como en la ecuación (2.3).

$$porcentaje = 100\% - \left(\frac{distancia\ entre\ sensor\ y\ objeto\ (cm) \times 100\%}{distancia\ total\ del\ contenedor\ (60cm)} \right) \quad (2.3)$$

2.1.2. Red de comunicaciones LoRaWAN

En un ecosistema IoT se tienen varios dispositivos finales que se necesitan interconectar para la detección, almacenamiento y análisis de datos. En el caso de LoRaWAN para el proceso de comunicación es necesario disponer de dispositivos finales ubicados para ser ubicados en los contenedores de basura y el gateway que permitirá enviar la información hacia la nube TTN.

2.1.2.1. Dispositivo final LoRaWAN

Entre las características que debe incluir el dispositivo final que es parte del prototipo se tiene:

- **Conectividad:** se debe utilizar la tecnología de comunicación LoRa para la transmisión de datos a través de redes LPWAN. La frecuencia de operación debe estar en la banda libre de 915 MHz definida para el Ecuador.
- **Alcance:** El módulo MKR WAN 1310 proporciona un alcance de hasta varios kilómetros en áreas abiertas y puede penetrar paredes y obstáculos con facilidad,

debido a que es un sistema de administración de basura, es importante considerar que las ciudades cuentan con varios objetos de obstrucción.

- Consumo de energía: debe estar diseñado para ser eficiente en términos de consumo de energía, lo que permite una mayor duración de la batería para aplicaciones alimentadas por energía.
- Interfaces: la placa tiene que incluir interfaces seriales tales como I2C, SPI y UART, que permiten la conexión y comunicación con otros dispositivos y sensores.

Existe una amplia gama de productos en el mercado que permiten la comunicación de los sensores con la tecnología LoRaWAN, como se observa en la Tabla 2.2.

Tabla 2.2. Placas para trabajar con LoRaWAN [18]

Módulos	Características
Arduino MKR WAN 1300	<ul style="list-style-type: none"> • Este módulo está basado en el microcontrolador de Arduino MKR. • Utiliza la tecnología LoRaWAN para la comunicación. Es compatible con la red LoRaWAN y puede ser utilizado para desarrollar aplicaciones IoT de bajo consumo energético. • Opera en las frecuencias de banda libre de 868 MHz (Europa) o 915 MHz (América). • Proporciona un alcance de hasta varios kilómetros en áreas abiertas y puede penetrar paredes y obstáculos con facilidad.
Dragino LoRa Shield	<ul style="list-style-type: none"> • Este módulo se conecta directamente a una placa Arduino (como Arduino Uno) a través de sus pines. • Proporciona conectividad LoRaWAN y puede ser utilizado para enviar y recibir datos a través de redes LoRa. • Opera en las frecuencias de banda libre de 868 MHz (Europa) o 915 MHz (América). • Proporciona un alcance de hasta varios kilómetros en áreas abiertas y puede penetrar paredes y obstáculos con facilidad.
Heltec ESP32 LoRaWAN	<ul style="list-style-type: none"> • Este módulo combina la potencia del microcontrolador ESP32 y la conectividad LoRaWAN.

Módulos	Características
<p align="center">Heltec ESP32 LoRaWAN</p>	<ul style="list-style-type: none"> • Incluye una pantalla OLED integrada que facilita la visualización de datos en el dispositivo. • Opera en las frecuencias de banda libre de 868 MHz (Europa) o 915 MHz (América).
<p align="center">MKR WAN 1310</p>	<ul style="list-style-type: none"> • Está equipada con un módulo de comunicación que permite la conectividad a través de redes LPWAN, como LoRaWAN y Sigfox. • Estas redes proporcionan comunicación de larga distancia y bajo consumo de energía, lo que las hace ideales para aplicaciones de IoT. • El MKR WAN 1310 utiliza un microcontrolador Arm Cortex-M0+ SAMD21 de 32 bits de baja potencia de Atmel. • Opera en las frecuencias de banda libre de 868 MHz (Europa) o 915 MHz (América). • Proporciona un alcance de hasta varios kilómetros en áreas abiertas y puede penetrar paredes y obstáculos con facilidad.

El dispositivo final LoRaWAN que se ha elegido para la implementación del prototipo es la placa basada en Arduino MKR WAN 1310, al ser un módulo que se encuentra disponible en el mercado y se ajusta a los requerimientos del prototipo.

2.1.2.2. Gateway LoRaWAN RAK 7243

Para el caso del gateway LoRAWAN como se planteó en el plan aprobado se definió utilizar el equipo RAK7243C, que es una puerta de enlace LoRaWAN de 8 canales completa que cumple con las normas Clase A y C, en su núcleo está el Raspberry Pi 3B+, que integra todos los demás componentes de hardware y ejecuta el sistema operativo.

Incluye un módulo concentrador RAK2245, que desempeña el papel de un transceptor LoRa. También incluye el RAK2013 que proporciona un backhaul celular; además de conectividad Wi-Fi y Ethernet de Raspberry Pi 3B+. Soporta bandas de frecuencia de EU433 / EU868 / US915 / AU915 / AS923 [19].

Este gateway se encuentra encendido de forma continua y está configurado como un gateway público por lo que se puede acceder a la nube TTN mediante este dispositivo.

TTN permite conocer la distribución de gateways alrededor del mundo, como se observa en la Figura 2.2.

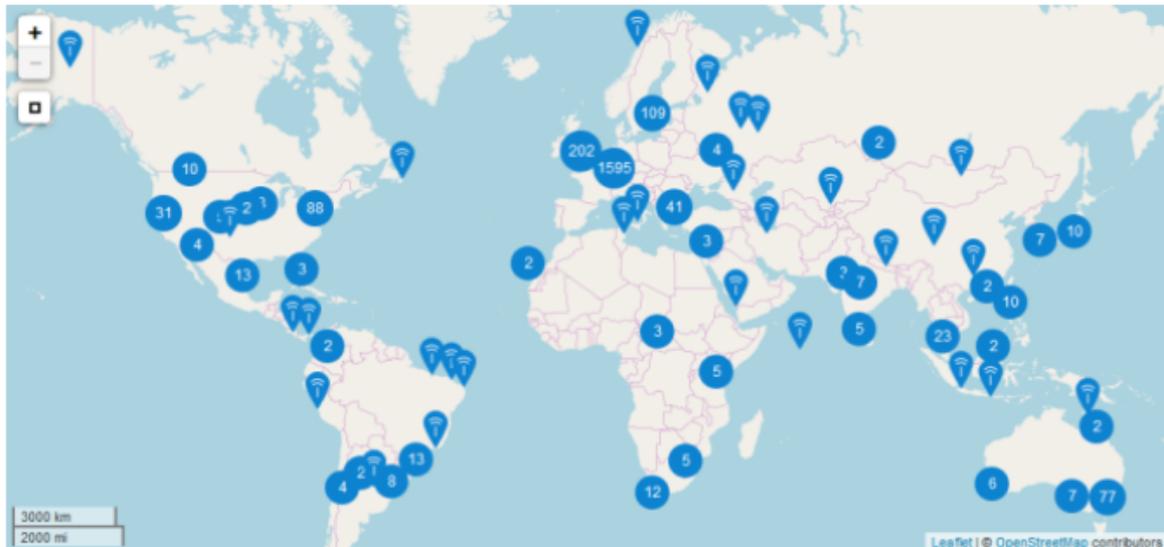


Figura 2.2. Distribución de gateways alrededor de mundo [20]

2.1.3. Servidores de Red y Aplicación

The Things Network es una red pública segura que admite un cifrado de extremo a extremo, reducción de riesgos a ataques de intermediarios y soporte para diferentes claves de cifrado de 128 bits para cada dispositivo final [21].

TTN proporciona un conjunto de herramientas abiertas, y una red global abierta para crear aplicaciones IoT a bajo costo, con máxima seguridad y altamente escalable. Ha desarrollado el *backend* de la red y da soporte a los gateways distribuidos por el mundo.

Además, cuenta con el servidor de red *The Things Stack* adecuado para implementaciones públicas y privadas distribuidas geográficamente, así como para redes locales. La arquitectura sigue el modelo de referencia de red LoRaWAN para el cumplimiento de estándares e interoperabilidad.

Por otra parte, para presentar datos del porcentaje de llenado del contenedor de basura, se utiliza la plataforma IoT ThingsBoard, de código abierto que permite la gestión y la recopilación de datos en tiempo real.

ThingsBoard proporciona un conjunto de herramientas y características para la creación de aplicaciones y soluciones IoT, permitiendo la integración de múltiples dispositivos y sistemas.

Ofrece una arquitectura escalable y modular, lo que significa que se puede adaptar a diferentes casos de uso y requerimientos específicos. La plataforma es flexible y permite al usuario crear aplicaciones y paneles de control personalizados para visualizar y analizar los datos recopilados.

2.2. Implementación del Prototipo

En la Figura 2.3 se presenta el diagrama del prototipo a ser implementado para el sensado del llenado de contenedores de residuos sólidos, así como del monitoreo de estos datos, el cual permitirá dar soporte a futuro a un sistema de administración de residuos sólidos.

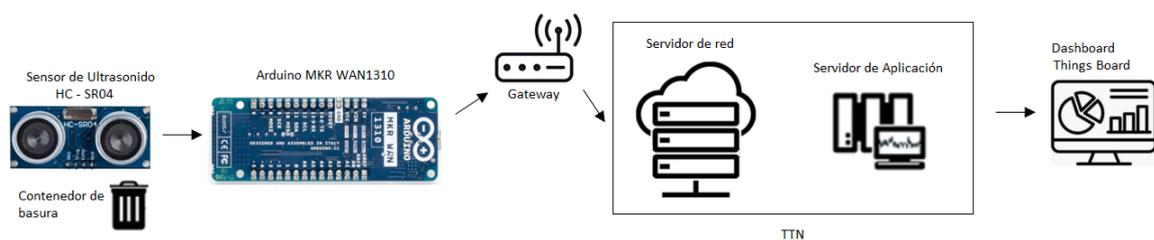


Figura 2.3. Prototipo para el sensado de distancia de residuos sólidos

2.2.1. Configuración para el sensor de ultrasonido

El sensor de Ultrasonido se conecta al módulo MKR WAN 1310 en base al diagrama de conexión presentado en la Figura 2.4.

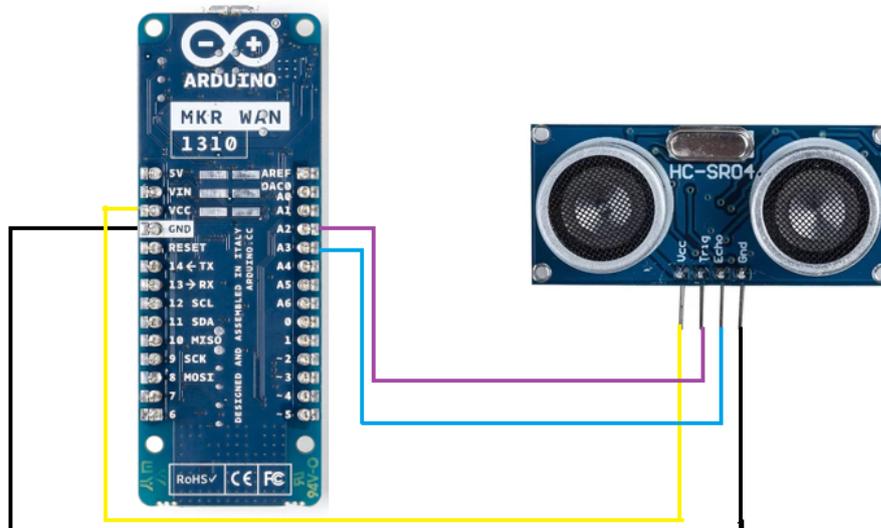


Figura 2.4. Diagrama de conexión del sensor de ultrasonido y el módulo

En base a este diagrama se realiza la conexión el módulo MKRWAN 1310 con el sensor de ultrasonido HC-SR04 como se muestra en la Figura 2.5

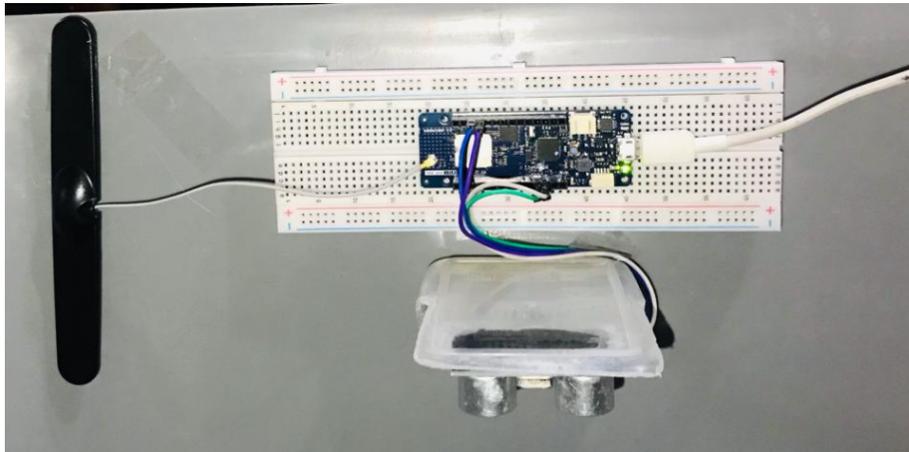


Figura 2.5. Conexión módulo MKRWAN 1310 con el sensor de ultrasonido HC-SR04

Finalmente, se realiza el montaje de la placa en el contenedor de basura como se presenta en la Figura 2.6.



Figura 2.6. Prototipo de contenedor de basura

Para el programa a ser cargado en la placa MKRWAN 1310 se ha seleccionado el pin 2 como el pin Trigger, el cuál envía los pulsos de ultrasonido y el pin 3, pin ECHO, cambiará a alto el tiempo que tarde la onda desde que fue emitida hasta ser detectada. En la Figura 2.7 se muestra el código a ser cargado para obtener la distancia mediante el sensor de ultrasonido en cm.

```
sketch_feb06a $
#include <NewPing.h>
#include <SPI.h>
#include <Wire.h>

// //////////////////////////////////////
#define TRIGGER_PIN 2
#define ECHO_PIN 3
#define MAX_DISTANCE 200
// //////////////////////////////////////

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

void setup() {
  Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:

  Serial.println(".....");

  delay(30000);
  Serial.print("Distancia: ");
  Serial.print(sonar.ping_cm());
  Serial.println("cm");

  float distancia=sonar.ping_cm();
}
```

Figura 2.7. Código para obtener la medida de distancia con el sensor HC-SR04.

En la Figura 2.8 se observa el Monitor Serie con los valores de distancia obtenidos en la prueba.



Figura 2.8. Monitor Serie con medidas de distancia

2.2.2. Configuración módulo MKR 1310 WAN

Se instala el Arduino SAMD Boards para agregar la tarjeta Arduino MKR WAN 1310 al IDE de Arduino como se muestra en la Figura 2.9.



Figura 2.9. Instalación de Arduino SAMD Boards

Luego de esto se agrega la tarjeta como se muestra en la Figura 2.10.

Herramientas > Placa: "Arduino MKR WAN 1310" > Arduino SAMD (32-bits ARM Cortex-M0+) Boards

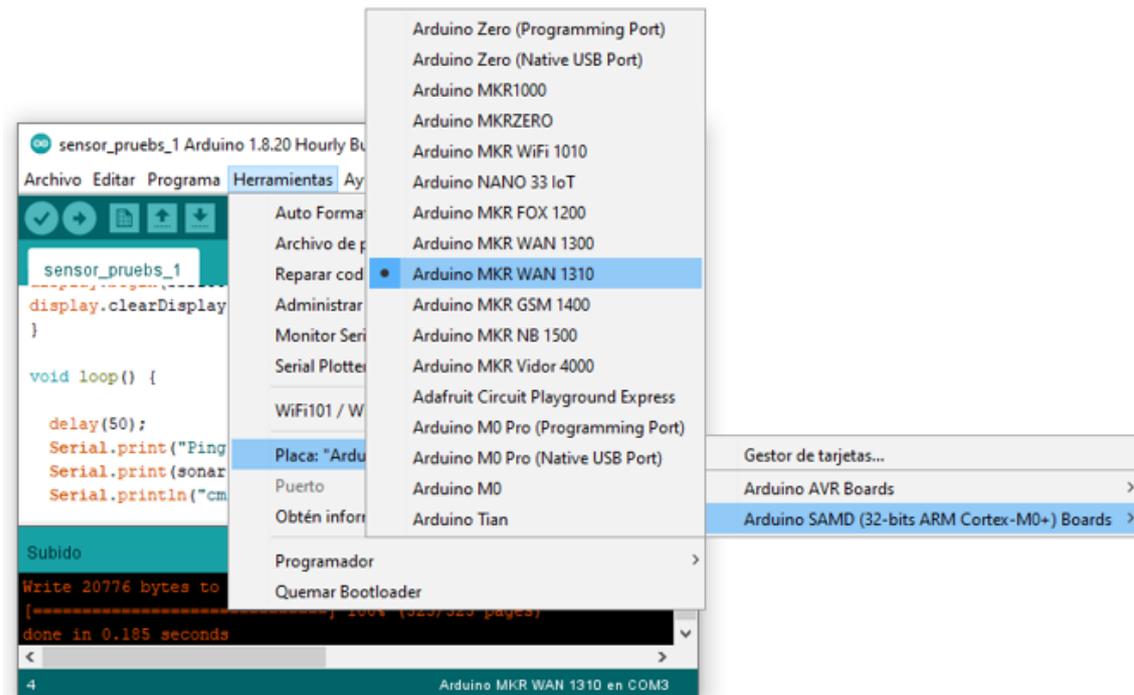


Figura 2.10. Agregar la tarjeta Arduino SAMD (32-bits ARM Cortex-M0+) Boards

Mediante el código de ejemplo "First Configuration" que ofrece Arduino (Figura 2.11), se obtiene el DevEUI del módulo. En la Figura 2.12 se observa el resultado.

Archivo > Ejemplos > MKRWAN > First Configuration

```
FirstConfiguration $
#include <MKRWAN.h>
LoRaModem modem;
// Uncomment if using the Murata chip as a module
// LoRaModem modem(Serial1);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  while (!Serial);

  // change this to your regional band (eg. US915, AS923, ...)
  if (!modem.begin(US915)) {
    Serial.println("Failed to start module");
    while (1) {}
  };
  Serial.print("Your module version is: ");
  Serial.println(modem.version());
  if (modem.version() != ARDUINO_FW_VERSION) {
    Serial.println("Please make sure that the latest modem firmware is installed");
    Serial.println("To update the firmware upload the 'MKRWANFWUpdate.ino'");
  }
  Serial.print("Your device EUI is: ");
  Serial.println(modem.deviceEUI());
}
void loop() {
```

Figura 2.11. Código para obtención de DevEUI

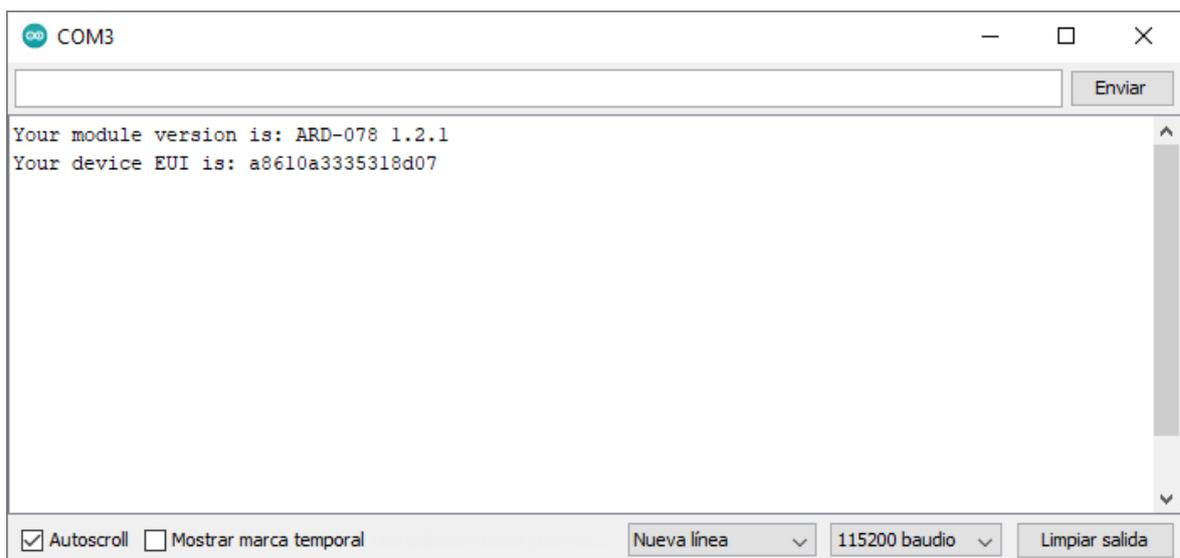


Figura 2.12. DevEUI del módulo MKR WAN 1310

2.2.2.1. Registro del nodo

En la plataforma *The Things Stack* se debe registrar el nodo MKR WAN 1310 identificado con el EUI “A8610A3335318D07”, que es el identificador del dispositivo en la red LoRaWAN.

En primer lugar, se crea la aplicación donde se encontrará el nodo. Para ello, se ingresa en Applications > Create Application. Se define el ID y el nombre de la aplicación y se crea la aplicación. Ver Figura 2.13.

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 *

test-waste

Application name

modulo sensor distancia

Description

Description for my new application

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

Create application

Figura 2.13. Creación de la aplicación

Una vez dentro de la aplicación se registra el nodo. En las Figuras 2.14 y 2.15 se observan los datos ingresados para la configuración de este módulo.

Figura 2.14. Registro del nodo, tipo de dispositivo

El AppKey se genera una vez que se ingrese el DevEUI, cada vez que se de click en generar el AppKey, este será diferente.

Figura 2.15. Registro del nodo, aprovisionamiento del módulo

Como se presenta en la Figura 2.16, la información con la que se ha configurado el nodo se puede observar en Applications > End devices > General Settings. El AppEUI y el AppKey son las claves que deben ser utilizadas en el código del módulo para que este se enlace a The Things Stack.

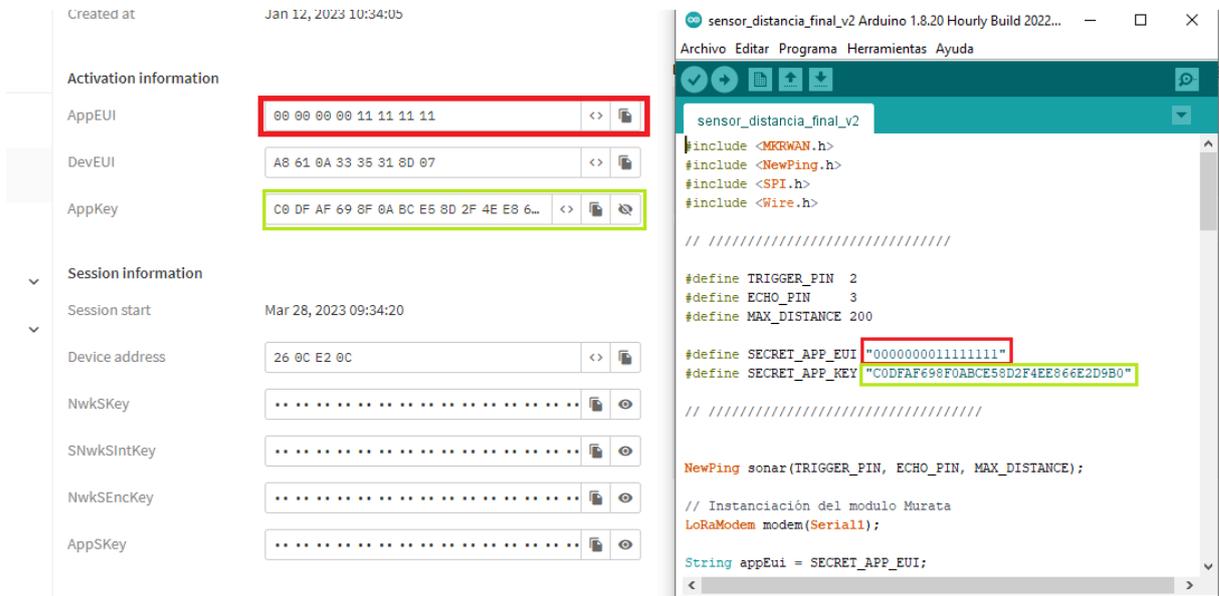


Figura 2.16. Configuración general del registro del nodo

2.2.2.2. Activación OTAA

Para realizar el proceso de activación OTAA se tienen los siguientes pasos:

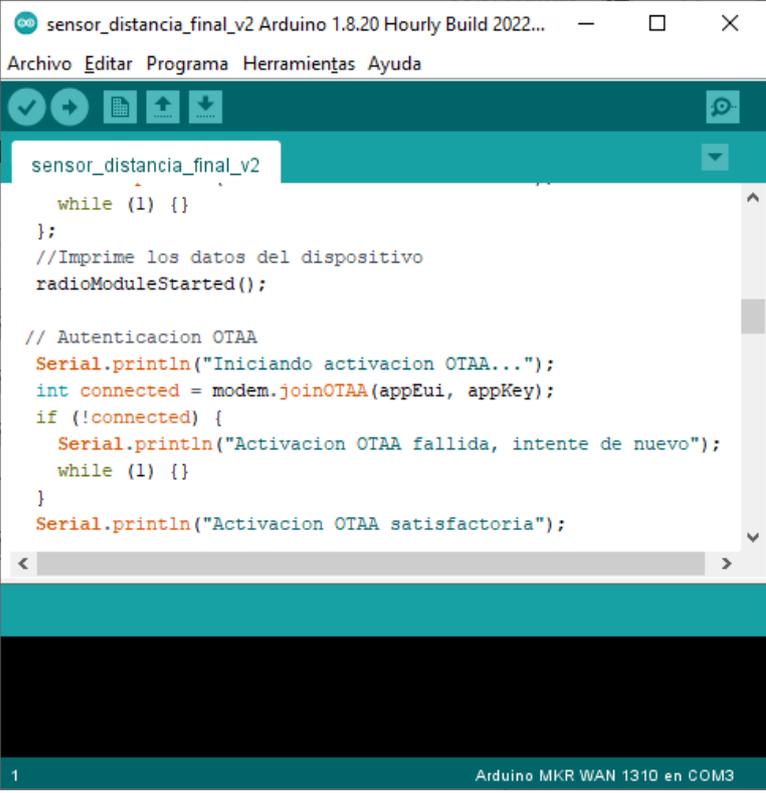
- **Registro del dispositivo:** se registra el módulo en la plataforma TTN, proporcionando el DevEUI, AppEUI y AppKey. Estos valores se utilizan para identificar y autenticar el dispositivo en la red.
- **Join Request:** El dispositivo envía una solicitud de unirse a la red (*Join Request*) a través del *gateway* LoRaWAN. Esta solicitud incluye el DevEUI, AppEUI y AppKey previamente registrados.
- **Join Accept:** La red TTN recibe la solicitud de unirse y realiza la validación de la autenticidad del dispositivo. Si la autenticación es exitosa, la red envía un *Join Accept* como respuesta de unión al dispositivo a través del *gateway*.
- **Claves de sesión:** El *Join Accept* contiene información importante, como la clave de sesión de red (*NwkSKey*) y la clave de sesión de aplicación (*AppSKey*).

Estas claves se utilizan para cifrar y descifrar los mensajes enviados entre el dispositivo y la red, garantizando la confidencialidad y la integridad de los datos.

- **Confirmación de unión:** El dispositivo recibe el *Join Accept*, extrae las claves de sesión y las almacena de forma segura. Luego, confirma con un *Join Confirmation* su unión a la red, lo que indica que está listo para comenzar a enviar y recibir datos a través de la red LoRaWAN.

Una vez completado el proceso de activación OTAA, el dispositivo puede comenzar a comunicarse con la red TTN utilizando las claves de sesión asignadas. Esto permite el intercambio bidireccional de datos entre el dispositivo y las aplicaciones o servicios conectados a la red.

En la Figura 2.17 se observa el código implementado en el nodo para la activación OTAA, para esto se utiliza el *AppEUI* y *AppKey* ingresados anteriormente. En el Anexo B se encuentra el código completo del módulo MKRWAN1310.



```
sensor_distancia_final_v2
while (1) {}
};
//Imprime los datos del dispositivo
radioModuleStarted();

// Autenticacion OTAA
Serial.println("Iniciando activacion OTAA...");
int connected = modem.joinOTAA(appEui, appKey);
if (!connected) {
  Serial.println("Activacion OTAA fallida, intente de nuevo");
  while (1) {}
}
Serial.println("Activacion OTAA satisfactoria");
```

Figura 2.17. Activación OTAA

Se puede observar en TTN el intercambio de mensajes como se presenta en la Figura 2.18.

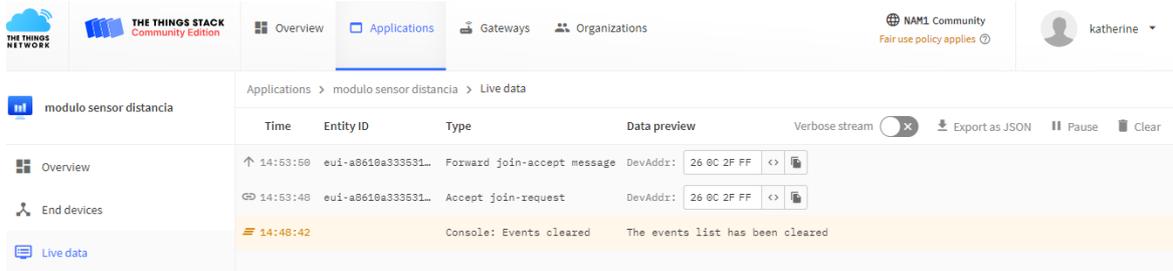


Figura 2.18. Intercambio de mensajes de activación OTAA en TTN

2.2.2.3. Cambio de tipo de variable

Se realiza un arreglo de 4 bytes para cambiar el tipo de variable que recibe desde el sensor, para el caso del prototipo corresponde al valor de distancia, el mismo se presenta en la Figura 2.19. La variable será colocada bytes.

```

sensor_distancia_final_v2 Arduino 1.8.20 Hourly Build 2022...
Archivo Editar Programa Herramientas Ayuda

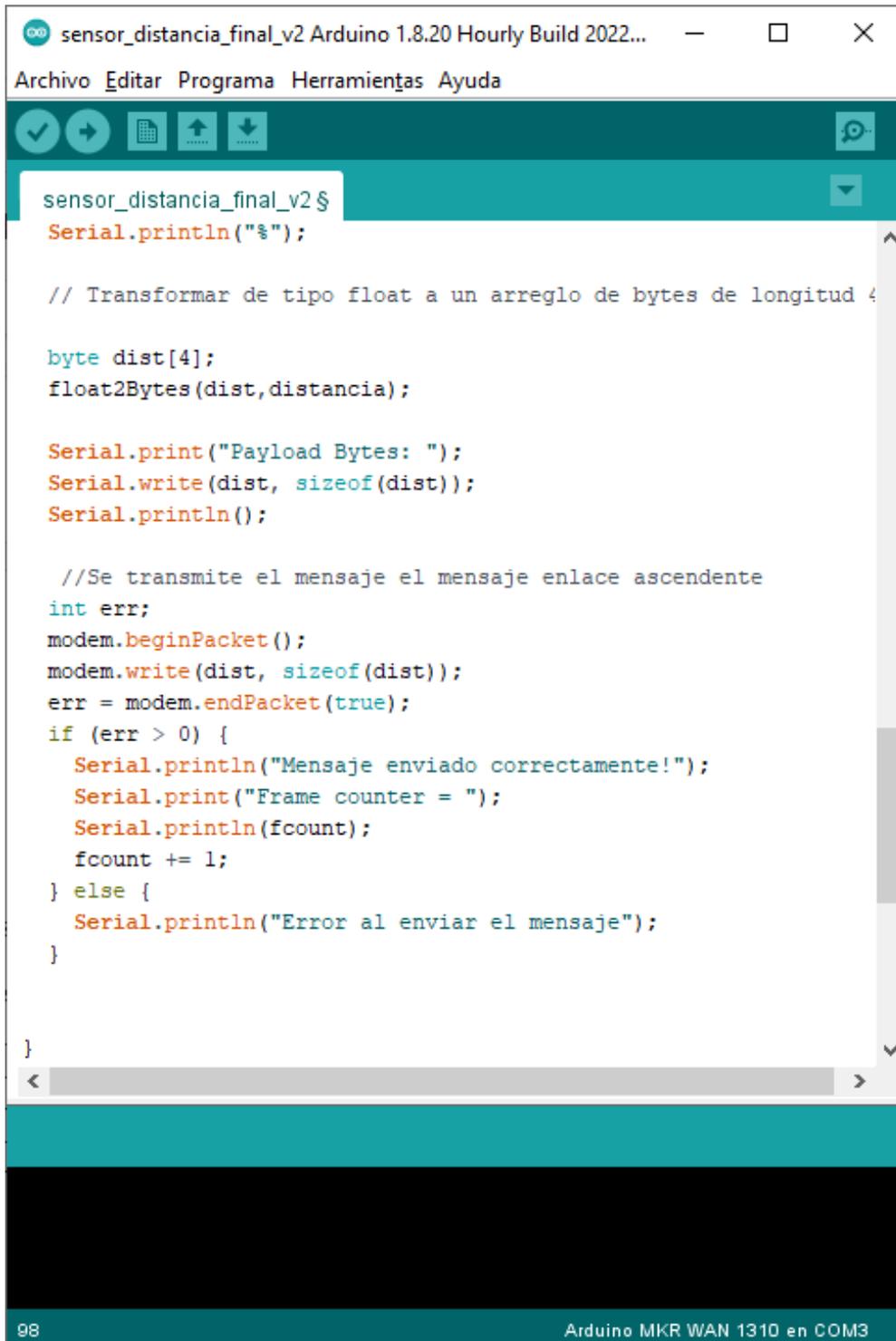
sensor_distancia_final_v2 $
}

// funcion para convertir de float a bytes
void float2Bytes(byte bytes_sensor[4],float float_variable){
  union {
    float a;
    unsigned char bytes[4];
  } thing;
  thing.a = float_variable;
  memcpy(bytes_sensor, thing.bytes, 4);
}

```

Figura 2.19. Código de cambio de tipo de variable

En la Figura 2.20 se muestra el código donde se inicia el módulo Murata para enviar los datos de distancia a la nube TTN.



```
sensor_distancia_final_v2 $
Serial.println("%");

// Transformar de tipo float a un arreglo de bytes de longitud 4

byte dist[4];
float2Bytes(dist,distancia);

Serial.print("Payload Bytes: ");
Serial.write(dist, sizeof(dist));
Serial.println();

//Se transmite el mensaje el mensaje enlace ascendente
int err;
modem.beginPacket();
modem.write(dist, sizeof(dist));
err = modem.endPacket(true);
if (err > 0) {
    Serial.println("Mensaje enviado correctamente!");
    Serial.print("Frame counter = ");
    Serial.println(fcount);
    fcount += 1;
} else {
    Serial.println("Error al enviar el mensaje");
}

}

98 Arduino MKR WAN 1310 en COM3
```

Figura 2.20. Código de iniciación de módulo Murata

2.2.3. Registro del Gateway

Para realizar el registro del Gateway en The Things Network, primero se obtiene el Gateway EUI, ver Figura 2.21.

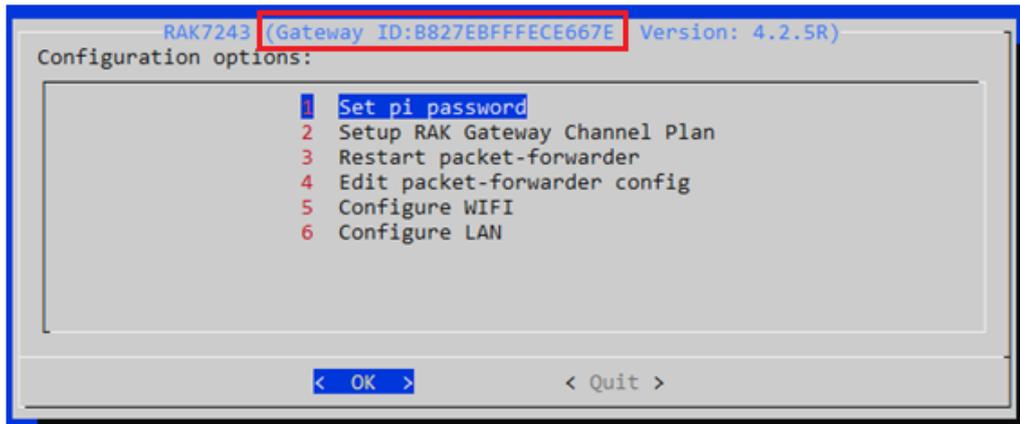


Figura 2.21. Gateway EUI

Luego, en *The Things Network* se realiza el registro del *Gateway* ingresando a TTN > *Console* > Elegir la opción *Register a Gateway* (Figura 2.22).

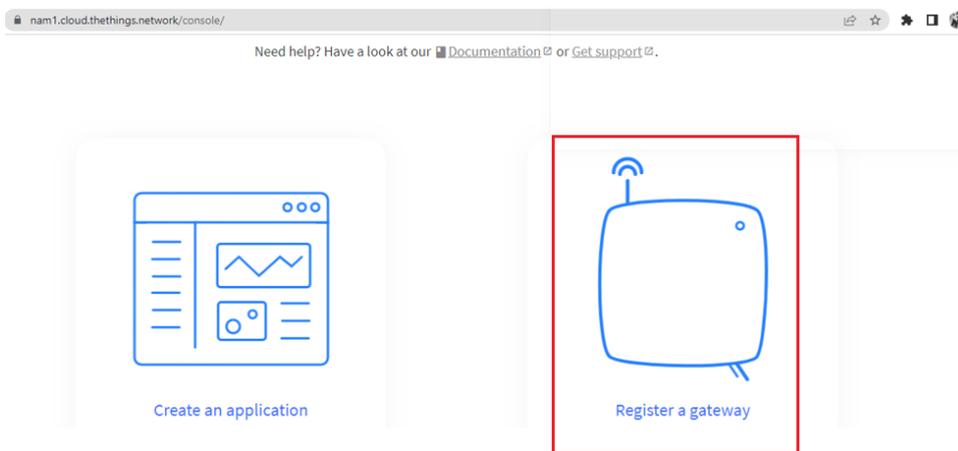


Figura 2.22. Selección de Registro de Gateway

Una vez seleccionado el registro, se ingresa el Gateway ID en el Gateway EUI y se confirma el registro, como se observa en la Figura 2.23.

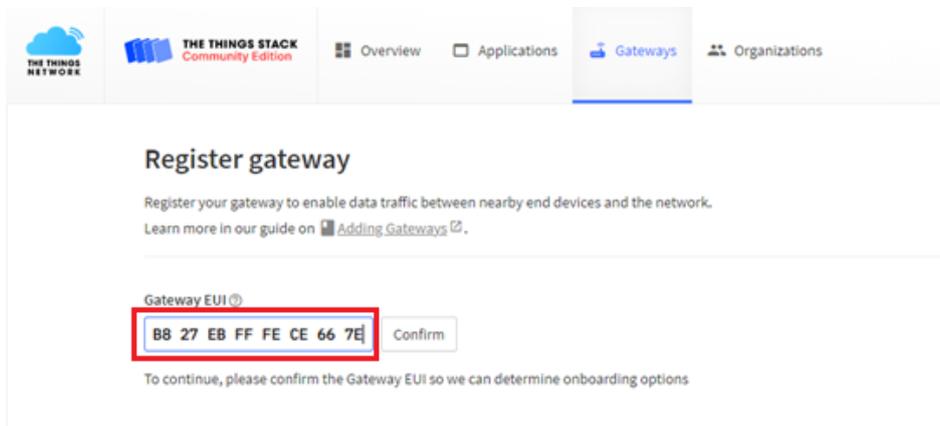


Figura 2.23. Ingreso de Gateway ID en TTN

2.2.4. Integración con Things Board

Para la presentación del porcentaje de llenado de los contenedores de basura se utiliza el dashboard “*The Things board*”, para lo cual es necesario realizar la integración de TTN con esta plataforma.

Para esta integración se debe generar un *Uplink Data Converter* y *Downlink Data Converter*.

Integrations center > *Data converters* > *Add Data Converter*

- **Uplink Data Converter**

Analiza el *payload* del mensaje entrante y lo transforma al formato que usa *ThingsBoard*. En la Figura 2.24 se indica los parámetros de configuración y en la Figura 2.25 la función utilizada. En el Anexo C se encuentra el código de Uplink data converter.

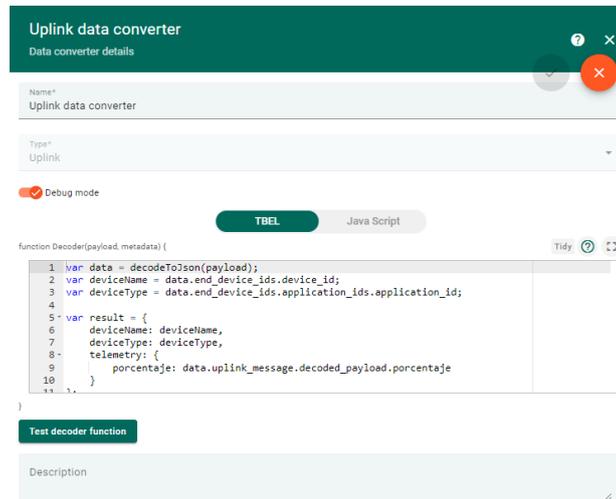


Figura 2.24. Generación de *Uplink Data Converter*

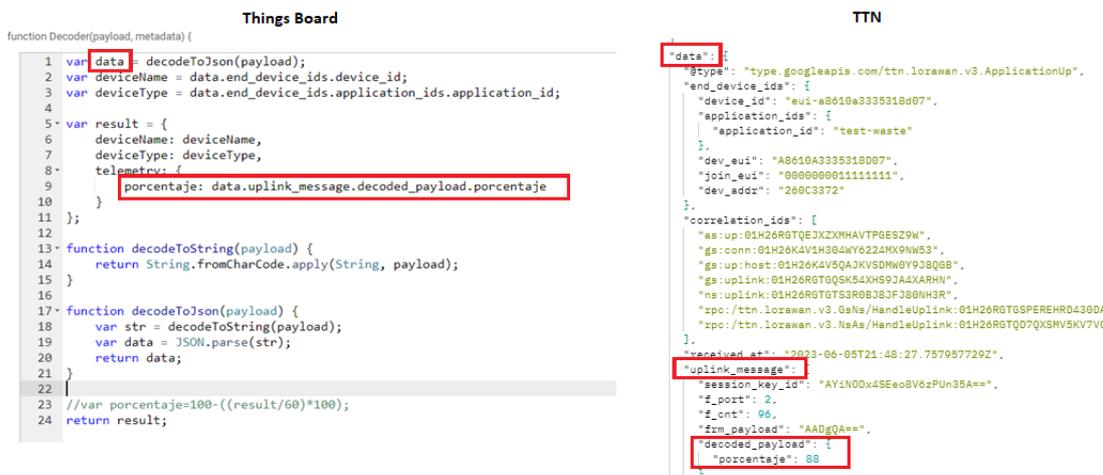


Figura 2.25. Configuración de la función

- **Downlink Data Converter**

Transforma el mensaje del motor de reglas entrante y sus metadatos al formato que utiliza la integración correspondiente.

En la Figura 2.26 se indica los parámetros de configuración y en la Figura 2.27 la función utilizada. En el Anexo D se encuentra el código del Downlink data converter.

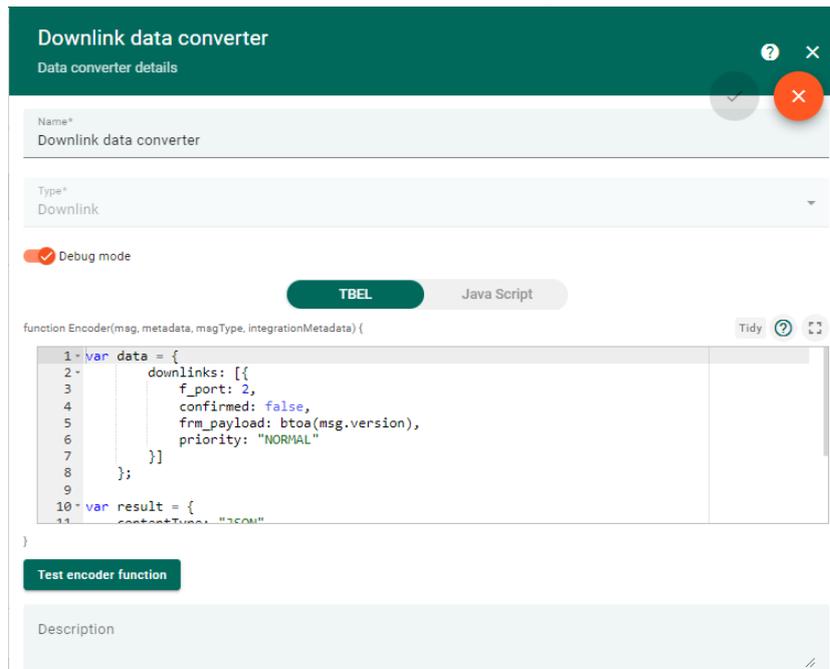


Figura 2.26. Generación de *Downlink Data Converter*



Figura 2.27. Configuración de la función

Una vez generado el *Uplink* y el *Downlink Data Converters*, se genera el *API Key*. Figuras 2.28 y 2.29.

Applications > modulo sensor distancia > API keys

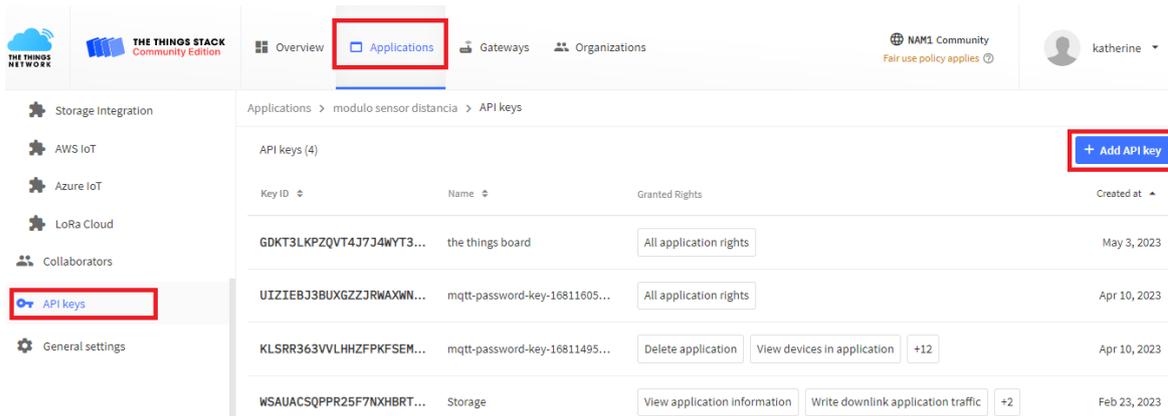


Figura 2.28. Creación API Key

Add API key

Name

Expiry date

Rights *

Grant all current and future rights

Grant individual rights

Create API key

Figura 2.29. Creación API Key

El Key ID es el parámetro *Password* que se necesitará en la integración, ver Figura 2.30.

Edit API key

Key ID *

Name

Expiry date

Rights *

Grant all current and future rights

Figura 2.30. API Key

Para crear la integración, seleccionar el *Uplink Data Converter*, *Downlink Data Converter* y el *API Key* generado en los pasos anteriores.

Los datos de *Region*, *Port* y *Username* se obtiene de TTN. Figura 2.31.

Applications > módulo sensor distancia > *Integrations* > MQTT

Connection information

MQTT server host

Public address

Public TLS address

Connection credentials

Username

Password [Go to API keys](#)

Figura 2.31. Datos: *Region*, *Port* y *Username* en TTN

En la Figura 2.32 presenta como se realiza la integración de Things Board con TTN.

TTN Integration
Integration details

Name* TTN Integration Integration type* The Things Stack Community

Enable integration
 Debug mode
 Allow create devices or assets

Uplink data converter* Uplink data converter Downlink data converter* Downlink data converter

Host type* Region Region* nam1 .cloud.thethings.network

Port* 8883

Credentials
Username* test-waste Password*

Enable SSL

Use API v3

Topic filters
Topic v3/+ /devices/+ /up QoS 0 - At most once

Execute remotely

Advanced settings

Figura 2.32. Integración de Things Board con TTN.

Finalmente, se revisa que se establezca la conexión en *Check Connection*, como se muestra en la Figura 2.33.

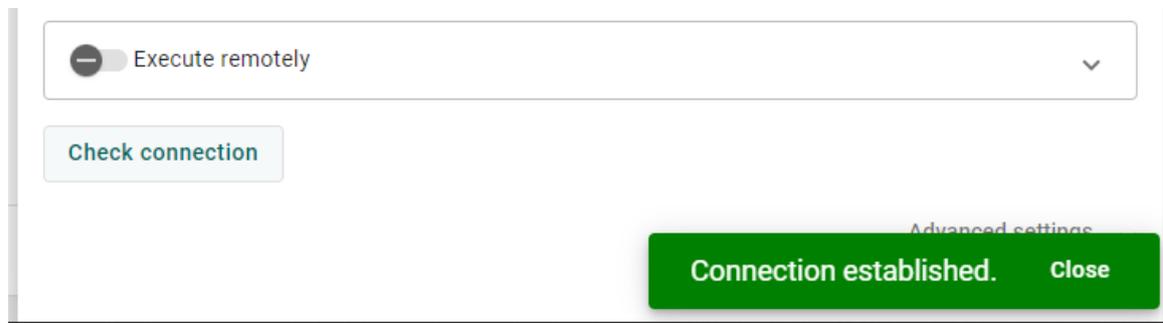


Figura 2.33. Conexión de *Things Board* con TTN establecida

Cuando la conexión es exitosa el nodo se registra automáticamente en la pestaña *Devices* de *Things Board*, Figura 2.34.

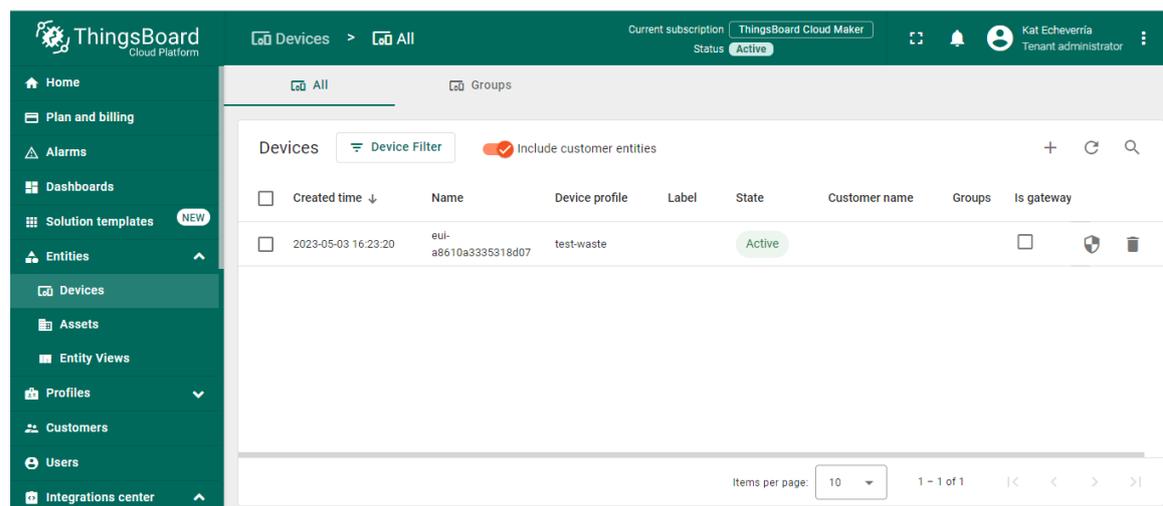


Figura 2.34. Registro del nodo en Things Board

Una vez que se observe que el nodo está registrado se puede crear el widget (Figura 2.35) para observar el porcentaje de llenado.

Devices > Latest telemetry > seleccionar la unidad que se va a observar > Click en Show on widget.

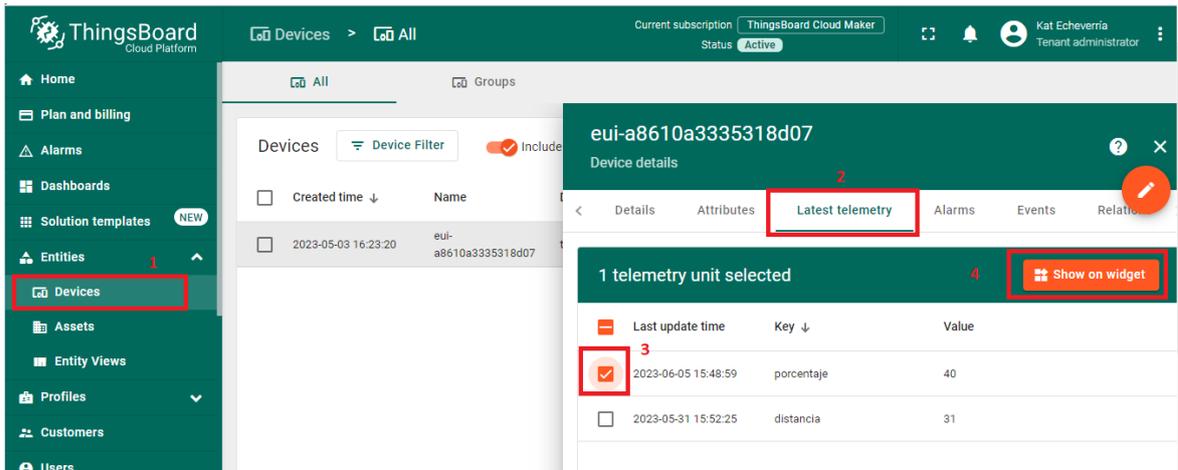


Figura 2.35. Generación de widget

Luego, se elige la forma en la que se requiere presentar los datos, y se agrega al *Dashboard* (Figura 2.36).

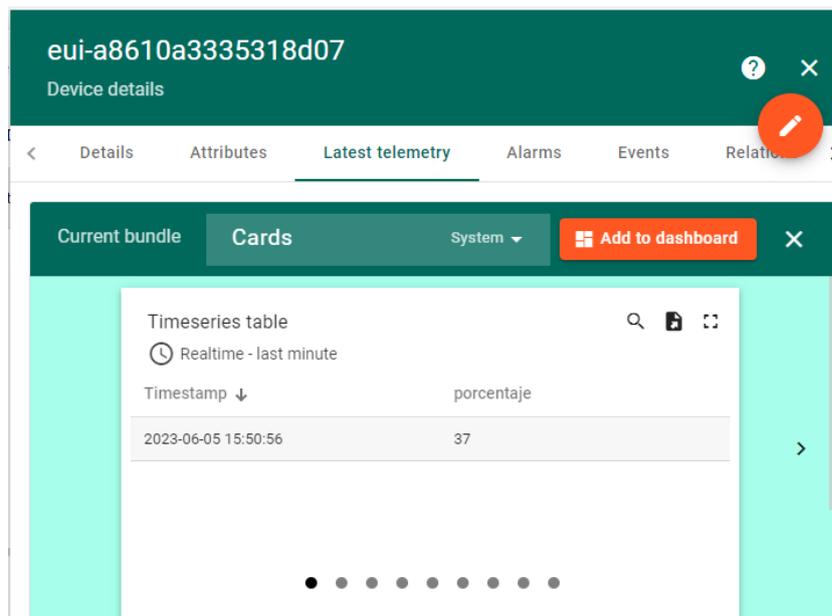


Figura 2.36. Selección de Widget

3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1. Resultados

En el caso del módulo de adquisición de datos, con el sensor de ultrasonido HC SR-04 se obtuvo, en primer lugar, las medidas de distancia entre el sensor y los residuos, estos valores son enviados al módulo MKRWAN 1310.

Mediante la Tabla 3.1 se pueden observar la variación de los datos capturados mediante el sensor.

Tabla 3.1. Resultado de distancia de vacío, variado la distancia

Distancia de prueba	Distancia de vacío medido con un metro (cm)	Distancia de vacío medida con el sensor
60cm	60cm	58 - 62 cm
50 cm	50 cm	48 - 52 cm
40 cm	40 cm	38 – 42 cm
30 cm	30 cm	28 – 32 cm
20 cm	20 cm	18 – 22 cm

Luego, el MKRWAN 1310, mediante el módulo de radio LoRa y la tecnología LoRaWAN reenvía los mensajes al Gateway LoRaWAN RAK7243 y los transmiten a la infraestructura de red.

La infraestructura de red TTN autentica el dispositivo, la gestión de las claves de cifrado y la gestión de la conectividad de los dispositivos en la red LoRaWAN.

Por último, se realiza la integración con Things Board para presentar los valores de distancia en el dashboard implementado en esta plataforma.

Una de las fallas recurrentes se produjo en la activación OTAA, como se muestra en la Figura 3.1. Si la activación no se realiza no se puede realizar la comunicación LoRaWAN, y los datos no serán transmitidos.

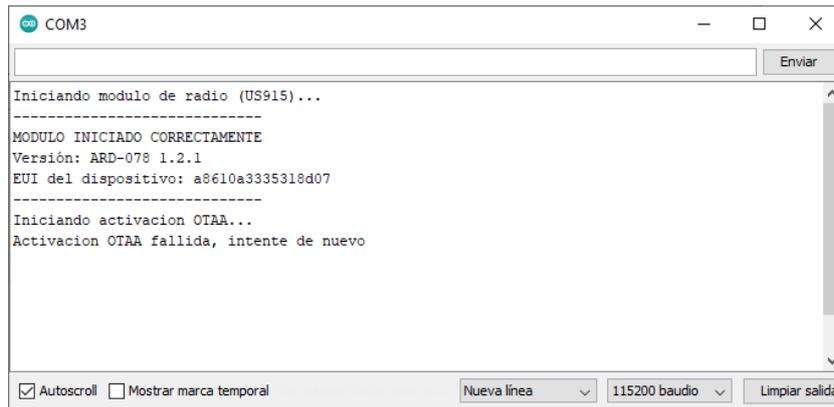


Figura 3.1. Activación OTAA fallida

Para resolver este problema, es importante verificar que la conexión a Internet sea estable ya que, al encender el módulo este requiere realizar la activación OTAA para que inicie la transmisión de los datos detectados. Una vez la activación es satisfactoria, cuando existe un error al enviar un valor de distancia, el módulo espera a recibir el siguiente valor sin tener que reiniciar el módulo.

Otro parámetro a considerar es el firmware del módulo, el firmware está diseñado para admitir la comunicación LoRaWAN y proporcionar las funcionalidades necesarias para la transmisión de datos a través de la red LoRaWAN. Si el firmware se encuentra desactualizado la activación OTAA puede presentar problemas. El firmware se encuentra integrado en la placa y no se puede actualizar directamente en el módulo. Sin embargo, al actualizar la librería MKRWAN en el IDE de Arduino (Figura 3.2), se tiene la última versión del firmware.

Herramientas > Administrar bibliotecas > Gestor de Librería > Colocar MKRWAN

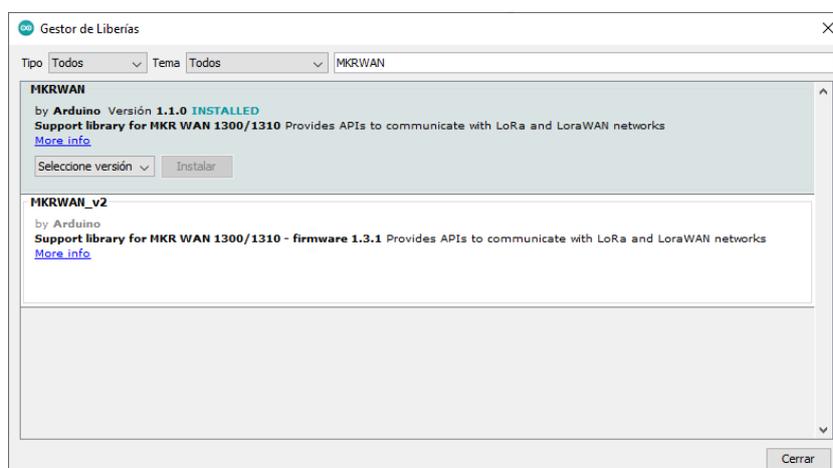


Figura 3.2. Librería MKRWAN en el IDE de Arduino

Antes de realizar la integración con Things Board, se comprueba que se esté recibiendo correctamente los valores de distancia detectados por el sensor. En wsl se escribe la siguiente línea de comando y se obtiene la información del módulo, como se muestra en la Figura 3.3:

```
mosquitto_sub -h "nam1.cloud.thethings.network" -p "1883" -u "test-waste@ttn" -P "NNSXS.UIZIEBJ3BUXGZZJRWAXWNQAGOTG7FVA7OZTAFPA.WYTHY4H3HQ3HS CXUUJPACZZUXMGE5VKOUC76KDXLLTPU37UHRBRA" -t "#"
```

```
kate@DESKTOP-2626QE6: ~
Selecting previously unselected package mosquitto-clients.
Preparing to unpack ./mosquitto-clients_2.0.15a-0mosquitto1-jammy1_amd64.deb ...
Unpacking mosquitto-clients (2.0.15a-0mosquitto1-jammy1) ...
Setting up libmosquitto1:amd64 (2.0.15a-0mosquitto1-jammy1) ...
Setting up libcbjson1:amd64 (1.7.15-1) ...
Setting up mosquitto-clients (2.0.15a-0mosquitto1-jammy1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
kate@DESKTOP-2626QE6: ~$ mosquitto_sub -h "nam1.cloud.thethings.network" -p "1883" -u "test-waste@ttn" -P "NNSXS.KLSRR363VVLHHZFPKFSQMQLLIBTMHBUZV3LESA.KHANWYFSDSLOSSVXNIB5ACEAASVTISNHUKC2ARVGKU73NDVW5PA" -t "#"
Connection error: Connection Refused: not authorised.
kate@DESKTOP-2626QE6: ~$ mosquitto_sub -h "nam1.cloud.thethings.network" -p "1883" -u "test-waste@ttn" -P "NNSXS.KLSRR363VVLHHZFPKFSQMQLLIBTMHBUZV3LESA.KHANWYFSDSLOSSVXNIB5ACEAASVTISNHUKC2ARVGKU73NDVW5PA" -t "#"
Connection error: Connection Refused: not authorised.
kate@DESKTOP-2626QE6: ~$ mosquitto_sub -h "nam1.cloud.thethings.network" -p "1883" -u "test-waste@ttn" -P "NNSXS.UIZIEBJ3BUXGZZJRWAXWNQAGOTG7FVA7OZTAFPA.WYTHY4H3HQ3HS CXUUJPACZZUXMGE5VKOUC76KDXLLTPU37UHRBRA" -t "#"
{"end_device_ids":{"device_id":"eui-a8610a3335318d07","application_ids":{"application_id":"test-waste"},"dev_eui":"A8610A3335318D07","join_eui":"0000000011111111","dev_addr":"260C9F08"},"correlation_ids":{"as:up:01GXPF59J0R4VCQNZE24FA7430","gs:conn:01GXPF71DDAMPT319BPFKMS8W2","gs:up:host:01GXPF71H7KDBYTB2HS6V1BA","gs:uplink:01GXPF59BHC3TV16Z79BNZMXTK","ns:uplink:01GXPF59B3HM64H2VGECH9A8RY"},"rpc:/ttn.lorawan.v3.GsNs/HandleUplink:01GXPF59B191HM3X01ZX2PDYM"},"rpc:/ttn.lorawan.v3.NsAs/HandleUplink:01GXPF59HYE4R0V8CC7REBRKTA"},"received_at":"2023-04-10T21:05:09.695125231Z","uplink_message":{"session_key_id":"AVds8-ourvRUBa9+ccz/aw==","f_port":2,"f_cnt":12,"frm_payload":"AAC4QQ==","decoded_payload":{"distancia":23},"rx_metadata":{"gateway_ids":{"gateway_id":"miot-pigw-01","eui":"B827EBFFEC667E"},"timestamp":1649397588,"rssi": -36,"channel_rssi": -36,"snr": 12,"location":{"latitude": -0.1648414058579469,"longitude": -78.47552885976047,"altitude": 2800,"source": "SOURCE_REGISTRY"},"uplink_token": "ChoKGAoMcMlvdClwaWd3LTaxEgi4+v//sSmFhDUnr+S8hoMCIX00aEGEPDeneKBIKDAh76AMA=="},"channel_index": 4,"received_at": "2023-04-10T21:05:09.398596373Z"},"settings":{"data_rate":{"lora":{"bandwidth": 125000,"spreading_factor": 8,"coding_rate": "4/5"},"frequency": "904700000"},"timestamp": 1649397588},"received_at": "2023-04-10T21:05:09.490313828Z"},"confirmed": true,"consumed_airtime": "0.102912s","network_ids":{"net_id": "000013","tenant_id": "ttn","cluster_id": "nam1","cluster_address": "nam1.cloud.thethings.network"}}
```

Figura 3.3. Información del módulo MKRWAN

Una vez confirmado que se realiza correctamente el envío de mensajes desde el módulo, se captura tres diferentes medidas de distancia para verificar el funcionamiento de la integración.

En la Figura 3.4 se presenta los correspondientes porcentajes de llenado, que detectó el sensor de ultrasonido dentro del contenedor de residuos sólidos, en The Things Stack.

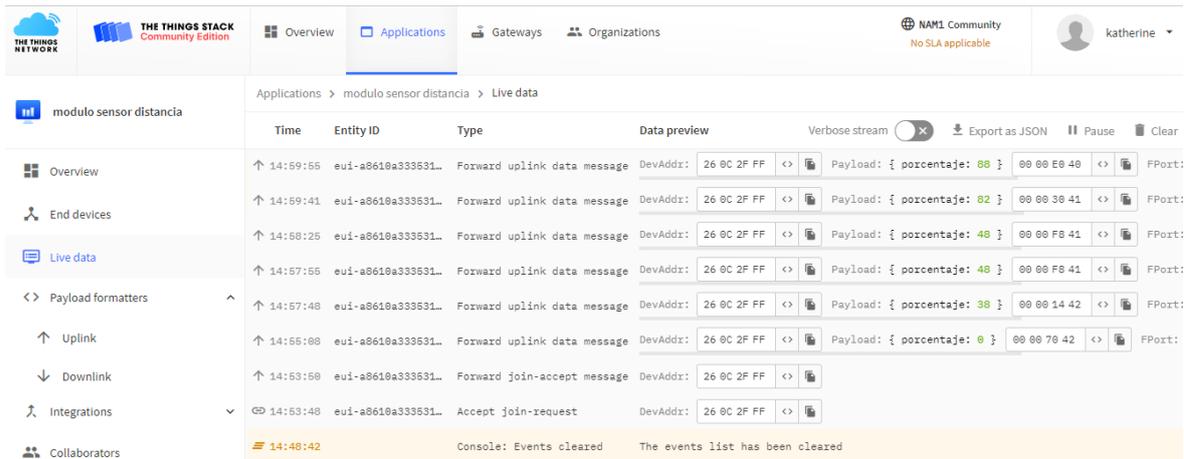


Figura 3.4. Porcentaje de llenado en The Things Stack

En la pestaña de Dashboard se puede agregar todos los estilos que se requiera la presentar los datos. En las Figuras 3.5, 3.6 y 3.7 se presentan el dashboard con los distintos valores de porcentaje.

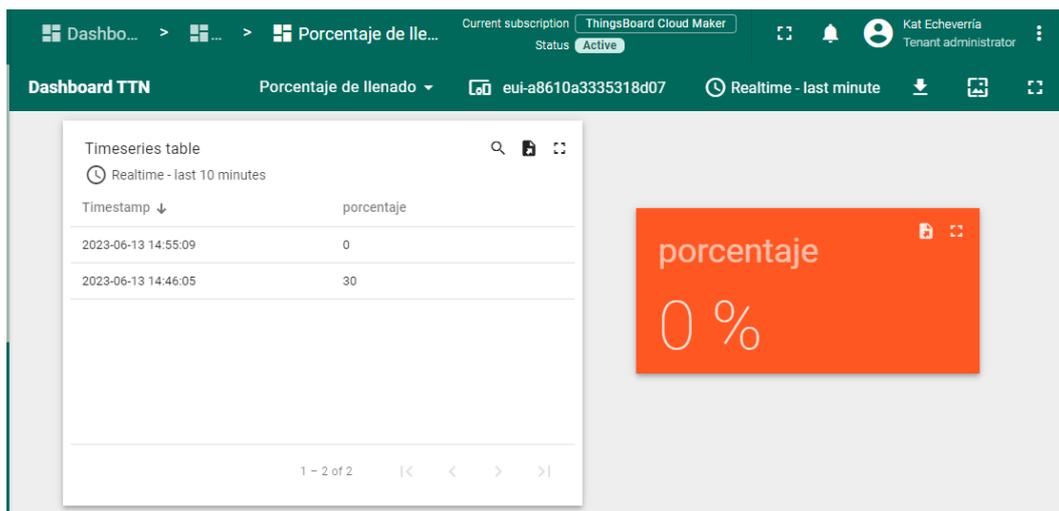


Figura 3.5. Porcentaje de llenado, contendor vacío

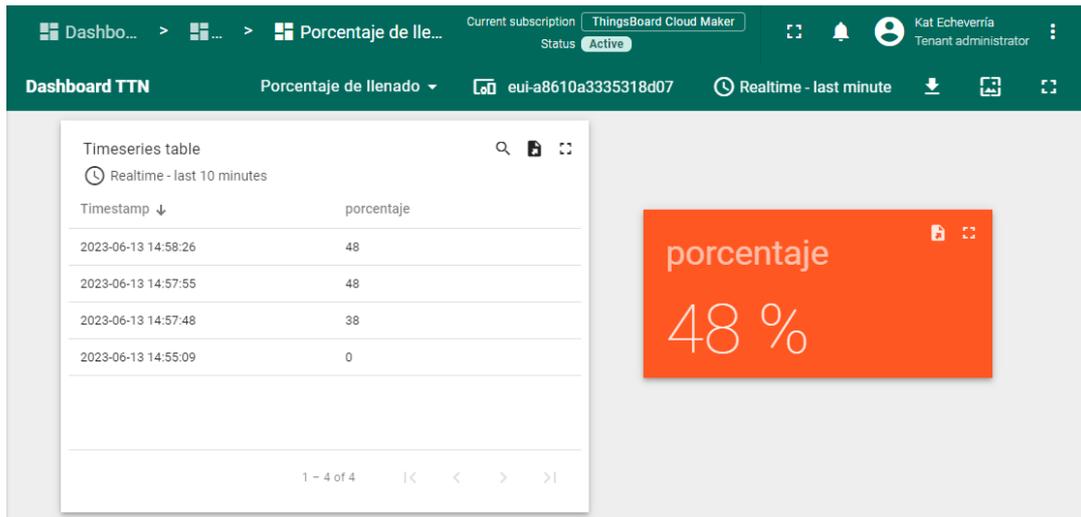


Figura 3.6. Porcentaje de llenado, contenedor al 48%

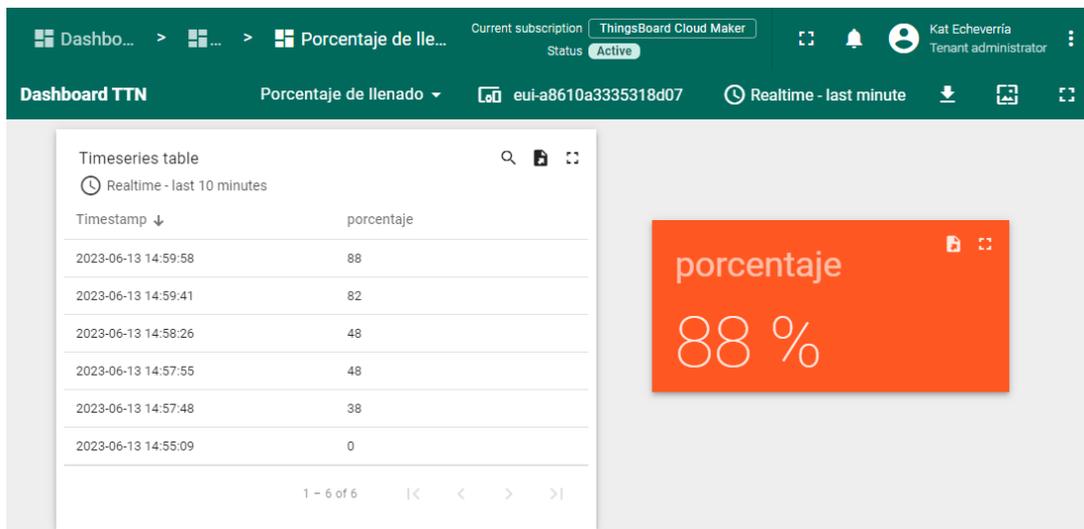


Figura 3.7. Porcentaje de llenado, contenedor al 88%

En base a la información obtenida de la distancia entre los residuos sólidos y la base del contenedor, se puede obtener el porcentaje de llenado.

El contenedor de basura tiene una altura máxima de 60 centímetros, la medida de distancia que se observa en la Figura 3.8 corresponde a la distancia entre el sensor hasta los residuos detectados.

Por ejemplo, con base en la Figura 3.5 para una distancia de 60 cm el porcentaje de llenado es 0%.

$$\text{porcentaje de llenado} = 100 \% - \left(\frac{60 \text{ cm} \times 100 \%}{60 \text{ cm}} \right) = 0\%$$

Para un porcentaje del 48%, Figura 3.6, se tiene una distancia de 29 cm de llenado, es decir entre el sensor y los residuos existe una distancia de 31 cm:

$$\text{porcentaje de llenado} = 100 \% - \left(\frac{31 \text{ cm} \times 100 \%}{60 \text{ cm}} \right) = 48 \%$$

Finalmente, para una distancia de 7cm significaría que la distancia de llenado es de 53 cm, lo que se traduce en un 88.33%, Figura 3.7

$$\text{porcentaje de llenado} = 100 \% - \left(\frac{7 \text{ cm} \times 100 \%}{60 \text{ cm}} \right) = 88 \%$$

Los datos recopilados por el sensor del contenedor inteligente pueden proporcionar información valiosa para la gestión de residuos a largo plazo. Los patrones de llenado y los flujos de residuos pueden ayudar a los municipios a tomar decisiones sobre la planificación urbana, la gestión de recursos y la optimización de la infraestructura de recolección de basura. Además, al reducir las recolecciones innecesarias y optimizar la gestión de residuos, los basureros inteligentes contribuyen a la sostenibilidad medioambiental, ya que se reduce el consumo de combustible de los vehículos de recolección y se disminuye la huella de carbono asociada con la administración de residuos.

Para llevar un control de estos datos, en el Monitor Serie del IDE de Arduino, Figura 3.8 se observa los valores de distancia y su correspondiente porcentaje de llenado.

```

COM3
Enviar
Iniciando modulo de radio (US915)...
-----
MODULO INICIADO CORRECTAMENTE
Versión: ARD-078 1.2.1
EUI del dispositivo: a8610a3335318d07
-----
Iniciando activacion OTAA...
Activacion OTAA satisfactoria
.....
Distancia: 60.00cm
Porcentaje de llenado: 0.00%
Payload Bytes: pB
Mensaje enviado correctamente!
Frame counter = 0
.....
Distancia: 60.00cm
Porcentaje de llenado: 0.00%
Payload Bytes: pB
Error al enviar el mensaje
.....
Distancia: 37.00cm
Porcentaje de llenado: 38.33%
Payload Bytes: CB
Mensaje enviado correctamente!
Frame counter = 1
.....
Distancia: 27.00cm

```

Autoscroll
 Mostrar marca temporal
Nueva línea ▾
115200 baudio ▾
Limpiar salida

Figura 3.8. Monitor Serie del IDE de Arduino

3.2. Conclusiones

En el contexto de las ciudades inteligentes, el Internet de las Cosas desempeña un papel importante al permitir la interconexión y la recopilación de datos de una amplia gama de dispositivos y sensores distribuidos por toda la ciudad, para mejorar la calidad de vida de los ciudadanos y optimizar la gestión urbana.

Existen varias formas de aportar al desarrollo de ciudades inteligentes, en cuestiones de movilidad, seguridad, eficiencia energética, agricultura, transporte y en la gestión de basura. Los contenedores de residuos sólidos equipados con sensores IoT pueden comunicar su nivel de llenado en tiempo real. Esto permitiría una recolección de basura más eficiente y evitaría el desbordamiento de los contenedores mejorando así la limpieza urbana.

LoRaWAN es tanto un protocolo como una tecnología. En cuanto al protocolo, LoRaWAN define la especificación de comunicación y las reglas que permiten que los dispositivos IoT se comuniquen entre sí y con la infraestructura de red. El protocolo establece los formatos de mensajes, los procedimientos de acceso al canal y la gestión de la red. Por otro lado, cuando se habla de tecnología, LoRaWAN se basa en la tecnología de modulación LoRa (Long Range). La tecnología LoRa permite una comunicación de largo alcance con bajo consumo de energía, lo que la hace ideal para aplicaciones de IoT. La modulación LoRa utiliza técnicas de espectro ensanchado y procesamiento de señales para lograr una mayor capacidad de penetración y alcance en comparación con otras tecnologías inalámbricas.

Para el despliegue de redes LoRaWAN y la interconectividad de dispositivos IoT, TTN presenta varias ventajas que facilita la implementación de aplicaciones. Por ejemplo, ofrece una cobertura de largo alcance que permite cubrir grandes áreas urbanas o rurales con un menor número de equipos. Además, gracias a su capacidad de escalabilidad, bajo costo y bajo consumo de energía se convierte en una opción viable para el desarrollo de investigaciones.

Para la implementación del contenedor de basura se consideraron varios requisitos que eran necesarios para su correcto funcionamiento, como la insensibilidad a la luz ambiental, un mayor rango de detección, la capacidad de detectar objetos transparentes, una mejor inmunidad al polvo y la suciedad, y la posibilidad de detectar múltiples objetos.

Para registrar y activar el módulo MKR WAN 1310 en la nube TTN se utiliza la activación OTAA debido a que ofrece una mayor seguridad y flexibilidad en comparación con el método de activación ABP, ya que las claves de sesión se generan dinámicamente durante

el proceso de unión y no se almacenan permanentemente en el dispositivo. Esto facilita la gestión y la rotación de claves de seguridad, lo que aumenta la protección contra posibles ataques.

ThingsBoard es una plataforma de IoT que permite la gestión, la recopilación y el análisis de datos en tiempo real de todos los dispositivos conectados, brindando a los usuarios la capacidad de crear aplicaciones personalizadas para sus necesidades específicas en el ámbito de la Internet de las cosas.

3.3. Recomendaciones

Para un ambiente de producción, se sugiere implementar el prototipo con una mayor cantidad de sensores, para que el área de detección sea más grande y cercana a valores reales.

Debido a que los gateways privados pueden desconectarse cuando el propietario lo requiera, se sugiere adquirir un Gateway LoRaWAN para realizar las pruebas sin correr el riesgo de desconexión.

Es importante generar la API KEY de acuerdo con el tipo de integración que se va a realizar, y se debe tomar en cuenta que esta no cambiará, aunque se cierre la plataforma por lo que no es necesario volver a generar una API KEY. Si existe una nueva, la integración ya establecida fallará.

Como trabajo futuro se puede integrar este prototipo a un sistema que permita generar alertas para que permitan indicar a los vehículos recolectores cuando se debe realizar el proceso de recolección de basura.

4. REFERENCIAS BIBLIOGRÁFICAS

- [1] Mora Bernal A., Ávila Larrea J., "Vista de La tasa de recolección de residuos sólidos: análisis desde la perspectiva tributaria ambiental | Iuris Dictio", *Revistas.usfq.edu.ec*, 2022. [Online]. Available: <https://revistas.usfq.edu.ec/index.php/iurisdictio/article/view/1813/2452>. [Accessed: 01- Aug- 2022].
- [2] Arduino, "Arduino MKR WAN 1300 (LoRa® connectivity)", Arduino Online Shop, 2022. [Online]. Available: <https://store-usa.arduino.cc/products/arduino-mkr-wan-1300-lora-connectivity?selectedStore=us>. [Accessed: 17- Jul- 2022].
- [3] Ayaskanta Mishra, Nisha Ghosh, Pujarini Jena, "Internet of Things based Waste Management System for Smart Cities: A real time route optimization for waste collection vehicles," *International Journal of Computer Sciences and Engineering*, Vol.7, Issue.4, pp.496-503, 2019.
- [4] Y. Chen, J. Nakazawa, T. Yonezawa, T. Kawasaki and H. Tokuda, "Cruisers: A Public Automotive Sensing Platform for Smart Cities," *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, Nara, Japan, 2016, pp. 767-768, doi: 10.1109/ICDCS.2016.53.
- [5] Dr.B.Paulchamy, E.Babu Thirumangai Alwar, K.Anbarasu, R.Hemalatha, R.Lavanya and K.M Manasa, "IOT Based Waste Management in Smart City" | *Ajast*, 2018, Volume 2, Issue 2, Pages 387-394, April-June 2018.
- [6] M. Tubaishat and S. Madria, "Sensor networks: an overview," in *IEEE Potentials*, vol. 22, no. 2, pp. 20-23, April-May 2003, doi: 10.1109/MP.2003.1197877.
- [7] Peris Martínez, J. (2020). Sistema de monitorización inalámbrica de temperatura mediante sensor de infrarrojos y microcontrolador ESP32. Universitat Politècnica de València. <http://hdl.handle.net/10251/153154>.
- [8] Proyectos con Arduino, "Sensores de distancia infrarrojos para Arduino", 2003. [Online]. Available: <https://proyectosconarduino.com/sensores/sensores-de-distancia-infrarrojos-para-arduino/>. [Accessed: 30- Jan- 2023].
- [9] Naylamp Mechatronics, "Sensor Ultrasonido HC-SR04,". [Online]. Available: <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>. [Accessed: 18-Jan-2023].

- [10] Tribalyte Technologies, "Sistema embebido: Conceptos fundamentales", [Online]. Available: <https://tech.tribalyte.eu/blog-sistema-embebido-caracteristicas/>. [Accessed: 05-Jan-2023].
- [11] Lora Alliance, "What is LoRaWAN® Specification", 2015. [Online]. Available: <https://lora-alliance.org/about-lorawan/>. [Accessed: 15-Jan-2023].
- [12] A. W. Azim, A. Bazzi, R. Shubair and M. Chafii, "Dual-Mode Chirp Spread Spectrum Modulation," in IEEE Wireless Communications Letters, vol. 11, no. 9, pp. 1995-1999, Sept. 2022, doi: 10.1109/LWC.2022.3190564.
- [13] A. Bassi, "Introducción a LoRaWAN", 2021. [Online]. Available: https://www.gotoiot.com/pages/articles/lorawan_intro/content.html/. [Accessed: 13-Jun-2023].
- [14] The Things Network, "LoRaWAN: Message Types", 2015. [Online]. Available: <https://www.thethingsnetwork.org/docs/lorawan/message-types/>. [Accessed: 16-Jan-2023].
- [15] The Things Network, "LoRaWAN: Message Types", 2015. [Online]. Available: <https://www.thethingsnetwork.org/docs/lorawan/message-types/>. [Accessed: 16-Jan-2023]
- [16] The Things Network, "LoRaWAN: Message Types", 2015. [Online]. Available: <https://www.thethingsnetwork.org/docs/lorawan/security/>. [Accessed: 16-Jan-2023].
- [17] L. Llamas, "Sensor de ultrasonido HC-SR04", 2015. [Online]. Available: <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>. [Accessed: 18-Jan-2023].
- [18] Arduino, "Related Boards". [Online]. Available: <https://store-usa.arduino.cc/products/arduino-mkr-wan-1310?selectedStore=us/>. [Accessed: 23-Jan-2023].
- [19] RAK wireless Official, "Gateway RAK7243C", [Online]. Available: <https://www.tel2u.com/gateway/rak-wireless-lora-wisgate-developer-pilot-gateway-pro-rak7243-2314419.html/>. [Accessed: 05-Jan-2023].
- [20] The Things Network, "ALL OVER THE WORLD" ,2015. [Online]. Available: <https://www.thethingsnetwork.org/community/>. [Accessed: 23-Jan-2023].

[21] The Things Stack, "Architecture". [Online]. Available: <https://www.thethingsindustries.com/docs/the-things-stack/concepts/architecture/>. [Accessed: 25-Jan-2023].