

Prototipo de Sistema de monitoreo basado en una red inalámbrica de sensores simulada, como apoyo a la planeación de rutas de recolección de basuras.

**Montañez Gómez, Miguel Ángel**

Universidad Nacional de Colombia

Facultad de Ingeniería

Bogotá, Colombia

2020

Prototipo de Sistema de monitoreo basado en una red inalámbrica de sensores simulada, como apoyo a la planeación de rutas de recolección de basuras.

**Miguel Ángel Montañez Gómez**

Trabajo final de Maestría para optar al título de:

**Magister en Ingeniería de Sistemas y Computación**

Director (a):

Ph.D. Luis Fernando Niño Vasquez

Línea de Investigación:

Nombrar la línea de investigación en la que se enmarca la tesis o trabajo de investigación

Grupo de Investigación:

Laboratorio de investigación en sistemas inteligentes (LISI)

Universidad Nacional de Colombia

Facultad Ingeniería de Sistemas e Industrial

Bogotá, Colombia

2020

*(Dedicatoria o lema)*

*A Dios por tantas bendiciones concedidas y la sabiduría para saber usarlas.*

*A mi esposa, por su apoyo incondicional y maravillosa forma de motivarme a ser mejor cada día.*

*A mis compañeros y profesores por compartir su invaluable conocimiento y ayuda*

*A mis padres por su convicción en mis capacidades y absoluta confianza*

**Declaración de obra original**

Yo declaro lo siguiente:

He leído el Acuerdo 035 de 2003 del Consejo Académico de la Universidad Nacional. «Reglamento sobre propiedad intelectual» y la Normatividad Nacional relacionada al respeto de los derechos de autor. Esta disertación representa mi trabajo original, excepto donde he reconocido las ideas, las palabras, o materiales de otros autores.

Cuando se han presentado ideas o palabras de otros autores en esta disertación, he realizado su respectivo reconocimiento aplicando correctamente los esquemas de citas y referencias bibliográficas en el estilo requerido.

He obtenido el permiso del autor o editor para incluir cualquier material con derechos de autor (por ejemplo, tablas, figuras, instrumentos de encuesta o grandes porciones de texto).

Por último, he sometido esta disertación a la herramienta de integridad académica, definida por la universidad.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

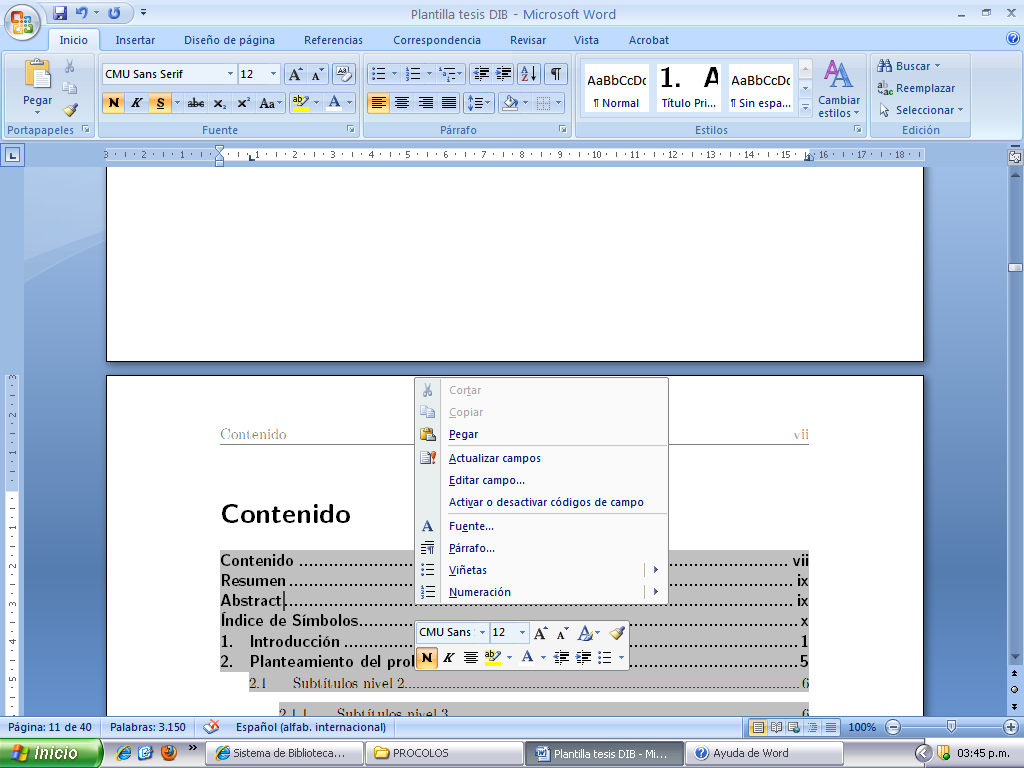
Miguel Ángel Montañez Gómez

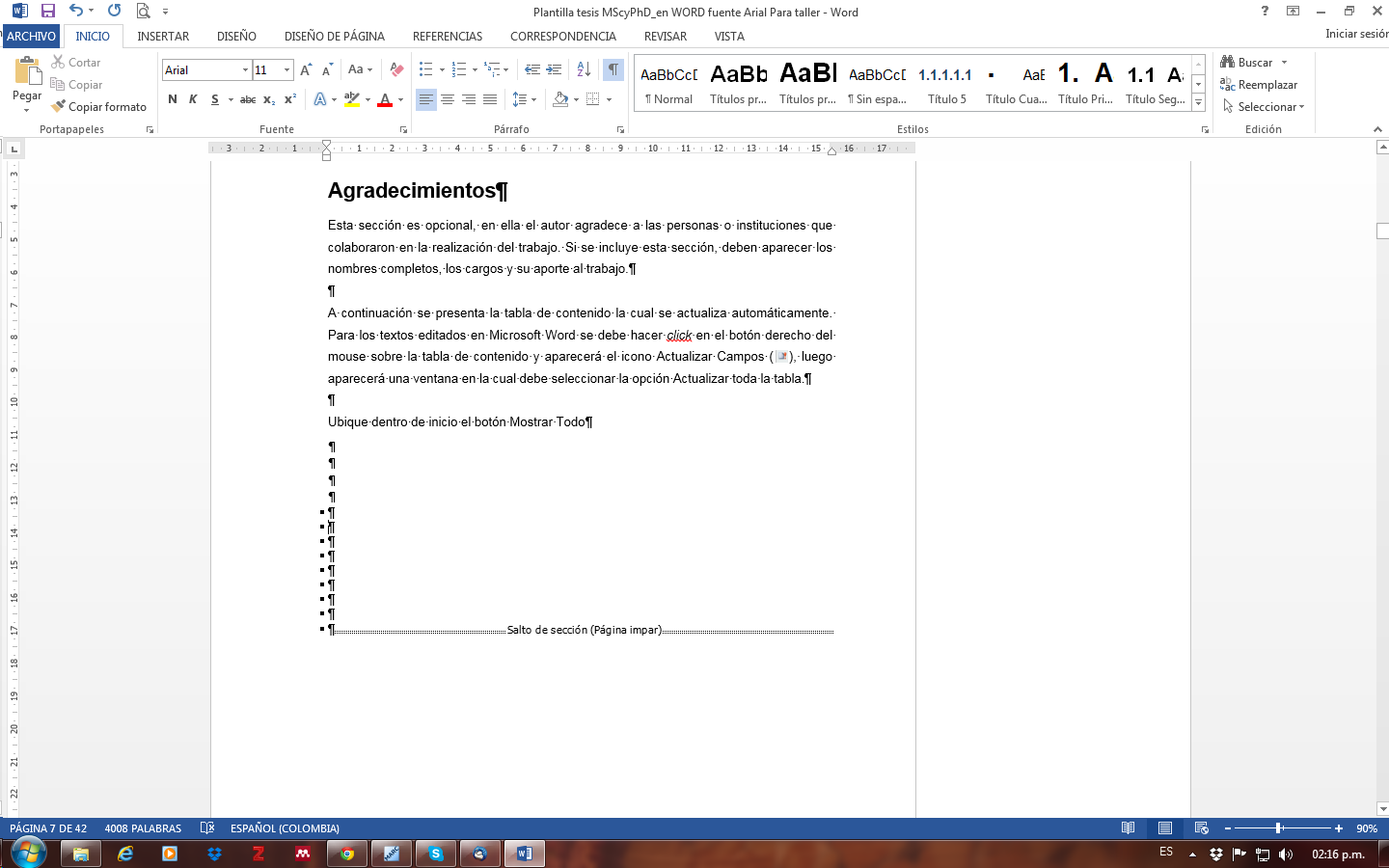
Fecha DD/MM/AAAA

Fecha

**Agradecimientos**

Esta sección es opcional, en ella el autor agradece a las personas o instituciones que colaboraron en la realización del trabajo. Si se incluye esta sección, deben aparecer los nombres completos, los cargos y su aporte al trabajo.

A continuación, se presenta la tabla de contenido la cual se actualiza automáticamente. Para los textos editados en Microsoft Word se debe hacer *click* en el botón derecho del mouse sobre la tabla de contenido y aparecerá el icono Actualizar Campos (), luego aparecerá una ventana en la cual debe seleccionar la opción Actualizar toda la tabla.

Tenga en cuenta al iniciar el diligenciamiento de la plantilla usar la opción del icono Mostrar Todo () Por medio de esta función podrá observar las secciones en que se encuentra construida esta plantilla, es muy importante conservar dichas secciones evitando borrar los saltos de sección. En caso de requerir más instrucciones sobre este tema por favor dirigirse al encargado de la Biblioteca Digital en cada una de las Sedes.

Resumen

**Prototipo de Sistema de monitoreo basado en una red inalámbrica de sensores simulada, como apoyo a la planeación de rutas de recolección de basuras:**

**(Resolución 023 de 2015. Artículo 02)\***

**SECCION PENDIENTE**

**\*Descripción:** debe incluir en sus preliminares, dentro del mismo pdf, tanto en inglés como en español, el título, el resumen y las palabras clave.

El resumen es una presentación abreviada. Se debe usar una extensión máxima de 250 palabras. Se recomienda que este resumen sea analítico, es decir, que sea completo, con información cuantitativa y cualitativa, generalmente incluyendo los siguientes aspectos: objetivos, diseño, lugar y circunstancias, pacientes (u objetivo del estudio), intervención, mediciones y principales resultados, y conclusiones. Al final del resumen se deben usar palabras claves tomadas del texto, las cuales permiten la recuperación de la información.

**Palabras clave: (Mínimo 3 y máximo 7 palabras, preferiblemente use lenguaje técnico-científico).**

**Abstract**

**Sensing system Prototype base on a Wireless Sensor Network as support for Solid Waste collection route planning: (Resolución 023 de 2015. Artículo 02)\***

**SECCION PENDIENTE**

**\*Descripción:** debe incluir en sus preliminares, dentro del mismo pdf, tanto en ingles como en español, el título, el resumen y las palabras clave.

**Keywords: (Mínimo 3 y máximo 7 palabras, preferiblemente use lenguaje técnico-científico).**

**Contenido**

Pág.

[Resumen IX](#_Toc42252778)

[Lista de figuras XII](#_Toc42252779)

[Lista de tablas XIII](#_Toc42252780)

[Lista de Símbolos y abreviaturas XIV](#_Toc42252781)

[Introducción 1](#_Toc42252782)

[1. Capítulo 1 3](#_Toc42252783)

[1.1 Subtítulos nivel 2 3](#_Toc42252784)

[1.1.1 Subtítulos nivel 3 3](#_Toc42252785)

[2. Capítulo 2 5](#_Toc42252786)

[2.1 Ejemplos de citaciones bibliográficas 5](#_Toc42252787)

[2.2 Ejemplos de presentación y citación de figuras 5](#_Toc42252788)

[2.3 Ejemplo de presentación y citación de tablas y cuadros 7](#_Toc42252789)

[2.3.1 Consideraciones adicionales para el manejo de figuras y tablas 8](#_Toc42252790)

[ Ejemplo de presentación y citación de ecuaciones 8](#_Toc42252791)

[3. Capítulo 3 9](#_Toc42252792)

[4. Capítulo (…) 11](#_Toc42252793)

[5. Conclusiones y recomendaciones 13](#_Toc42252794)

[5.1 Conclusiones 13](#_Toc42252795)

[5.2 Recomendaciones 13](#_Toc42252796)

[A. Anexo: Nombrar el anexo A de acuerdo con su contenido 15](#_Toc42252797)

[B. Anexo: Nombrar el anexo B de acuerdo con su contenido 17](#_Toc42252798)

[Bibliografía 19](#_Toc42252799)

Lista de figuras

Pág.

[**Figura 2‑1:** Tipos y parte del fruto de palma de aceite. 6](#_Toc42253129)

**Figura 3‑1:** Sensor de distancia laser Gyvl53l0xv2

Nota: Si es requerido, se pueden incluir lista de ilustraciones, graficas, diagramas, dibujos o fotografías. Tenga presente que estas lista deben ser generadas de forma automatizada utilizando las opciones que proporciona el software de procesamiento de texto.

Lista de tablas

Pág.

[**Tabla 2‑1**: Participación de las energías renovables primaria. 7](#_Toc42261026)

Nota: Si es requerido, se puede incluir la lista de cuadros, en caso que se utilicen en el desarrollo de la tesis o trabajo de investigación. Tenga presente que estas lista deben ser generadas de forma automatizada utilizando las opciones que proporciona el software de procesamiento de texto.

## Lista de Símbolos y abreviaturas

**Esta sección es opcional, dado que existen disciplinas que no manejan símbolos y/o abreviaturas. Se incluyen símbolos generales (con letras latinas y griegas), subíndices, superíndices y abreviaturas (incluir sólo las clases de símbolos que se utilicen). Cada una de estas listas debe estar ubicada en orden alfabético de acuerdo con la primera letra del símbolo (en esta plantilla, el título del tipo de símbolo está en letra Arial de 14 puntos y en negrilla). Para escribir la definición en las tablas, se puede usar la herramienta de referencia cruzada (para textos editados en Microsoft Word). A continuación, se presentan algunos ejemplos.**

**Símbolos con letras latinas**

| **Símbolo** | **Término** | **Unidad SI** | **Definición** |
| --- | --- | --- | --- |
| *V* | Diferencia de potencial a lo largo de un conductor | Voltio |  |
| *ABET* | Área interna del sólido |  | ver DIN ISO 9277 |
| *Ag* | Área transversal de la fase gaseosa | m2 | Ec. 3.2 |
| *As* | Área transversal de la carga a granel | m2 | Ec. 3.6 |
| *a* | Coeficiente | 1 | Tabla 3-1 |

**Símbolos con letras griegas**

| **Símbolo** | **Término** | **Unidad SI** | **Definición** |
| --- | --- | --- | --- |
| *α* | Factor de superficie |  | (*w*F,waf)(*A*BET) |
| *β* | Grado de formación del componente i | 1 |  |
| ** | Wandhafreibwinkel (Stahlblech) | 1 | Sección 3.2 |
|  | Porosidad de la partícula | 1 |  |
| *η* | mittlere Bettneigunswinkel (Stürzen) | 1 | Figura 3-1 |

**Subíndices**

| **Subíndice** | **Término** |
| --- | --- |
| bm | Materia orgánica |
| DR | Dubinin-Radushkevich |
| E | Experimental |

**Superíndices**

| **Superíndice** | **Término** |
| --- | --- |
| n | Exponente, potencia |

**Abreviaturas**

| **Abreviatura** | **Término** |
| --- | --- |
| *RSU* | Residuo Solido Urbano |
| *RIS* | Red Inalambrica de Sensores |
| *IoT* | Internet de las cosas |
| *COP* | Pesos Colombianos |
| *dB* | Decibels |
| *RFF* | Racimos de fruta fresca |

Introducción

**En la introducción, el autor presenta y señala la importancia, el origen (los antecedentes teóricos y prácticos), los objetivos, los alcances, las limitaciones, la metodología empleada, el significado que el estudio tiene en el avance del campo respectivo y su aplicación en el área investigada. No debe confundirse con el resumen y se recomienda que la introducción tenga una extensión de mínimo 2 páginas y máximo de 4 páginas.**

La presente plantilla maneja la fuente Arial para el texto de los párrafos y para los títulos y subtítulos. Sin embargo, es posible sugerir otras fuentes tales como Garomond, Calibri, Cambria o Times New Roman, que, por claridad y forma, son adecuadas para la edición de textos académicos.

Esta sección se **encabeza con la palabra introducción, escrita con minúscula (en la primera línea), con un espaciado anterior de 100 puntos y posterior de 24 puntos, interlineado sencillo y en letra negrilla de 20 puntos (en este caso se usa Arial).**

La presente plantilla tiene en cuenta aspectos importantes de la Norma Técnica Colombiana - NTC 1486 y el Manual de publicaciones de la APA, con el fin que sean usadas para la presentación final de las tesis de maestría, doctorado y especializaciones y especialidades en el área de la salud, desarrolladas en la Universidad Nacional de Colombia.

Las márgenes deben ser de 2,54 centímetros (1 pulgada) en la parte superior, inferior y exterior y de 3,6 centímetros en la margen interna (a partir de márgenes simétricos). La plantilla está diseñada para imprimir por lado y lado en hojas tamaño carta. Se sugiere que los encabezados cambien según la sección del documento (para lo cual esta plantilla está construida por secciones). El número de página se ubica en la parte superior derecha en las páginas impares y en la superior izquierda en las páginas pares (en letra Arial de 11 puntos, de acuerdo al formato presentado en esta plantilla). El título de cada capítulo debe estar numerado y comenzar en una hoja independiente (página impar) y con el mismo formato del título Introducción (**escrita con minúscula, en la primera línea, con un espaciado anterior de 100 puntos y posterior de 24 puntos e interlineado sencillo y en letra de 20 puntos y negrilla; en este caso se usa Arial).** El texto debe llegar hasta la margen inferior establecida. Se debe evitar títulos o subtítulos solos al final de la página o renglones sueltos.

Si se requiere ampliar la información sobre normas adicionales para la escritura se puede consultar la Norma Técnica Colombiana - NTC 1486 y el Manual de Publicaciones de la *American Psychological Association.*

La tesis o trabajo de investigación se debe escribir con interlineado a 1.5 líneas y después de punto aparte se dejan dos interlíneas (dos veces la tecla Enter). La redacción debe ser impersonal y genérica. La numeración de las hojas sugiere que las páginas preliminares se realicen en números romanos en mayúscula y las demás en números arábigos, en forma consecutiva a partir de la introducción que comenzará con el número 1. La cubierta y la portada no se numeran, pero si se cuentan como páginas.

A continuación se definen los objetivos que propusieron para el desarrollo de este trabajo:

OBJETIVO GENERAL

Desarrollar un prototipo de un sistema de monitoreo de contenedores de RSU, para planificar las rutas de recolección por demanda, usando como insumo información simulada.

OBJETIVOS ESPECIFICOS

* OE1: Desarrollar un dispositivo para medir de forma remota el nivel de llenado en un contenedor de Residuo Sólido Urbano.
* OE2: Implementar una red inalámbrica de sensores con tres dispositivos, usando una arquitectura de Internet de las Cosas (IoT).
* OE3: Simular la recolección de información sobre nivel de llenado de los  contenedores de basura ubicados en un área seleccionada de Bogotá (en la localidad de Engativá)
* OE4: Implementar una aplicación que permita generar y visualizar las rutas para la recolección y la localización de los contenedores que se deben atender.

# Marco Conceptual

## Sistemas de recolección de residuos

Estrategia de recolección de residuos en bogota.

## reciclaje

## Cuidades inteligentes

## Sistemas georeferenciados

## Sitema posicionamiento global GPS

## Internet de las cosas

## Software libre

## Hardware libre

## Sistemas embebidos

## Sensores

## comunicacion

## tecnologías inalámbricas RF

## transmisores y receptores (transceivers)

## LoRa

Cuando se diseñan aplicaciones con transmisión de datos de forma inalámbrica es importante reflexionar cual es la tecnología se amolda mejor a las necesidades de la aplicación que se desea implementar, por ejemplo el sistema de WiFi el muy conveniente para transmitir grandes volúmenes de datos a alta velocidad pero tiene un alcance limitado, otras tecnologías como las redes GSM superan la limitación de distancia pero a un costo energético elevado.

Algunas de aplicaciones como el monitoreo de variables ambientales en cultivos agrícolas, el monitoreo de la humedad en una región rural, el PH del agua en un afluente, en los que comúnmente los sitios que se deben monitorear se encuentran separados geográficamente y las variables que se desean medir se puede codificar en unos pocos bytes, LoRa se adaptan muy bien a estas Necesidades. Especialmente si los tiempos de medición de dichas variables no son muy frecuentes considerando que los datos no varían significativamente en periodos de tiempo prolongados, adicionalmente si debido a la ubicación remota de los sensores, resulta difícil realizar operaciones de mantenimiento como el reemplazo de baterías, de manera tal que la optimización de la energía se convierte en sea un aspecto crítico.



**Figura 2-1:** logotipo de productos LoRa original de semtech[[1]](#footnote-1)

Por sus siglas en ingles “Long Range Modulation” Es una tecnología para la trasmisión de información por vía inalámbrica que pertenece a la capa física y es comparable con WiFi o con Bluetooth. Es un mecanismo codificar información por medio de la modulación de onda diseñada con características de baja transmisión de datos usando bajo consumo energético y amplificando la cobertura a lo largo de redes que son redes conocidas como LPWAN (Low Power Wide Area Network)

LoRa es un tipo de modulación de espectro expandido que modula la información usando una señal que varía el continuamente su frecuencia FSCM (Frequency Shift Chirp Modulation) (Vangelista, 2017), para lo cual usa los parámetros frecuencia inicial (FqI) y final (FqF), su espectro comienza en el valor del parámetro FqI aumenta de manera lineal hasta alcanzar el parámetro FqF y repite el proceso de expandir la modulación en este rango de frecuencias, que están entre los 915 Mhz en America y Australia, en 868 Mhz en Europa y el 433 Mhz en algunos países de Asia como China y los cuales correspondes a la banda de frecuencias abierta para aplicaciones Industriales, Científicas y Medicas (ISM)

La cantidad de bits que se pueden codificar por símbolo depende de otro parámetro llamado factor de propagación SF que son configurables en 6 valores entre el 7 y el 12 y determinan la tasa de transmisión en conjunto con el ancho de banda (BW) que se elija entre los cuales están disponibles 125, 250 y 500 Mhz la relación de tasa de transmisión se puede expresar usando la siguiente ecuación:

Algunas de las propiedades más relevantes de esta modulación son: (solera, 2018)

Capacidad de la red: gracias al concepto de ortogonalidad lora puede codificar varias señales usando para ello diferentes factores de propagación sin afectar la decodificación de los mensajes

Localización: esta tecnología fue concebida inicialmente para la exploración con radares por lo que tiene un excelente desempeño en aplicaciones de localización y ubicación.

Envolvente constante: gracias a esta propiedad la demodulación de la señal es sencilla y puede reutilizar amplificadores para con ganacia programables.

Ancho de banda escalable: puede ajustar su ancho de banda y su frecuencia para operar en diferentes canales.

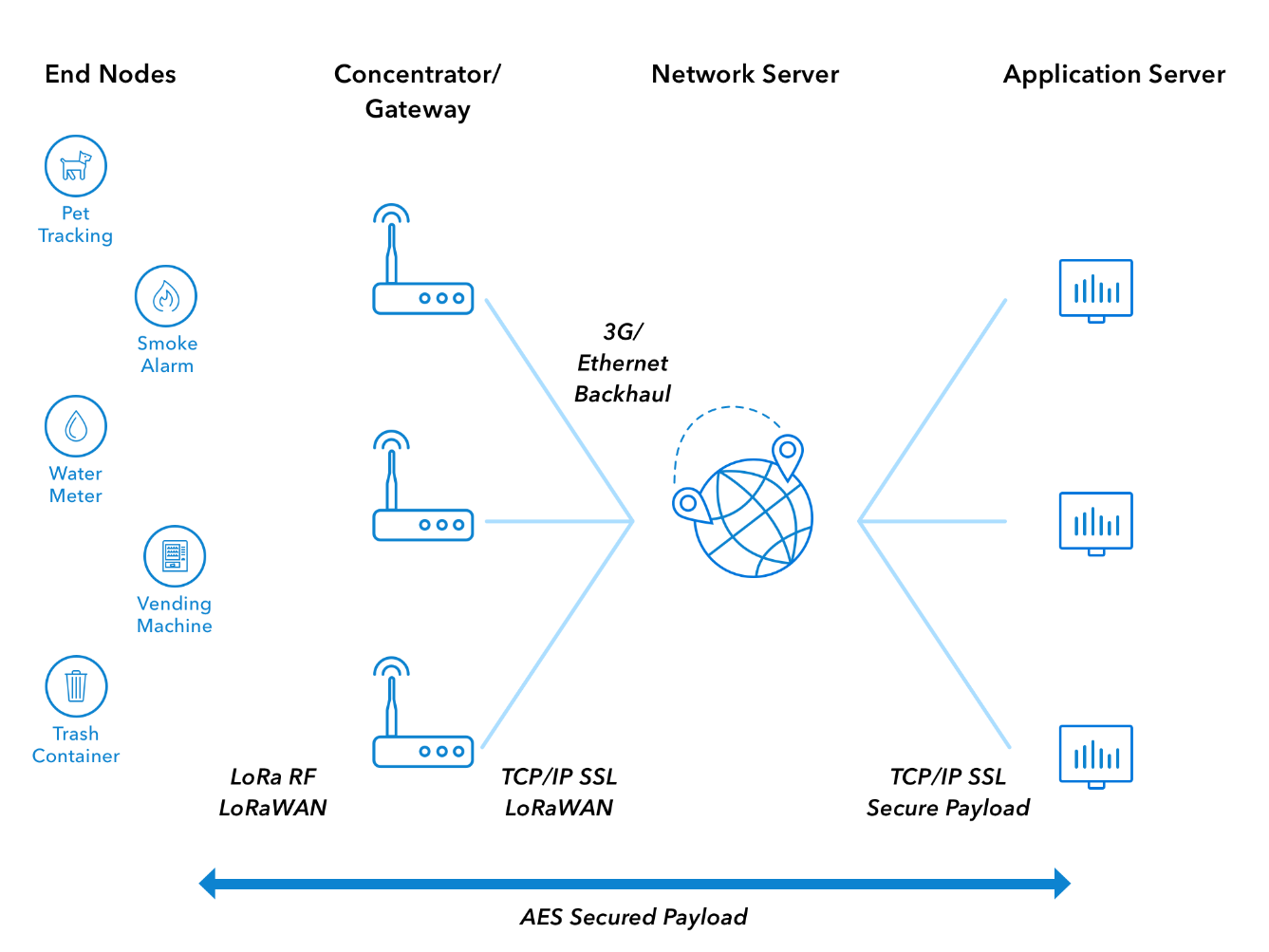
Inmunidad contra el efecto dopler: la modulación que usa LoRa tiene una baja sensibilidad gracias al desplazamiento de frecuencia asociado al efecto doppler, como el receptor está configurado para recibir la señal en un rango de frecuencia deja sin efecto la alteración de la frecuencia cuando los dispositivos de emisión se encuentran en movimiento.

Robustez: esta modulación es altamente resistente ante mecanismos de interferencia pues el receptor tiene selectividad alrededor de los 90 dB y una repulsión de co-canal de 20 dB.

## loraWAN

como ya hemos visto miestras que LoRa es una tecnología para la transición de datos que modula la frecuencia de la señal a nivel físico, LoRaWAN es una especificación para la implementación de LPWAN, que haciendo una clasificación según el estándar OSI corresponde al nivel 2 Control de Acceso Medio (MAC) en la que se definen las secuencias para el intercambio de los paquetes a nivel físico y ofrece servicios para administración del canal, el ancho de banda, la autenticación de dispositivos y el cifrado de datos mediante AES128, para la conformación de redes.

Las Redes LoRaWAN usan una topología de estrella en la que los nodos finales se comunican con un portal de acceso (gateway) que le da a la red acceso a internet y por medio del cual se enrrutan los paquetes enviados por los nodos hasta un servidor IoT, de manera que estos Gateway conforman una segunda estrella cuyo nucleo es el servidor de IoT.



**Figura 2-2.** Arquitectura típica red LoRaWAN, Extraida de The Things Network.

En este protocolo existen 2 componentes principales para conectar la red, la primera correspondiente a infraestructura, compuesta por los Gateway y las antenas que los nodos necesiten para comunicarse y la segunda una aplicación de administración, desde la cual se puedan registrar los nodos que hacen parte de esta aplicación, se pueda definir la estructura de la trama que remiten los nodos para poder decodificar la información y reescribirla en una codificación más simple de interpretar por un usuario final.

### Tipos de dispositivos

Cuando se genera un envio de paquetes en una red LoRaWAN si estos son emitidos desde los nodos hacia el Gateway se denominan mensajes ascendentes (Uplink) si los paquetes son enviados desde el Gateway hacia los nodos finales entonces los mensajes son descendentes (Downlink). Los nodos de la red pueden ser clasificados en 3 tipos según su configuración de comunicación.

**Clase A:**

Los dispositivos soportan comunicación bidireccional entre el dispositivo final y el Gateway y pueden enviar mensajes deliberadamente luego el dispositivo entra en estado de recepción o escucha durante 1 o 2 segundos en los que espera confirmación del servidor, si el servidor no responde en este tiempo, aun tendrá una opción de confirmar la recepción del mensaje cuando se ejecute la próxima comunicación ascendente. Con esta configuración los dispositivos obtienen un mejor rendimiento de batería.

**Clase B:**

Los dispositivos de establecen una ventanas de recepción con base a los tiempos que son establecidos por el Gateway, se debe tener en cuenta que si los periodos de escucha configurados son largos esto impactara directamente en rendimiento de la batería.

**Clase C:**

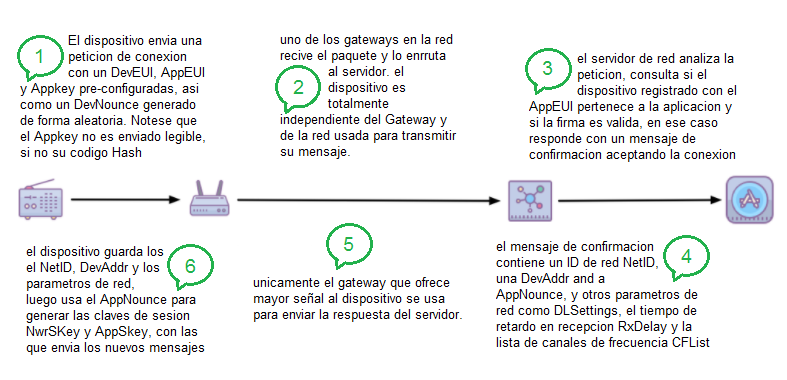
Los dispositivos se encuentra continuamente en modo de recepción y solamente cambian a modo de transmisión durante unas fracciones de segundo cuando deben emitir los datos, este tipo de dispositivos necesitan una fuente de alimentación externa.

### Autenticación

El protocolo LoRaWAN define 2 mecanismos de autenticación como herramienta para asegurar las aplicaciones, evitando que sean vulnerables ante ataques de terceros.

**Activación en el aire OTAA**

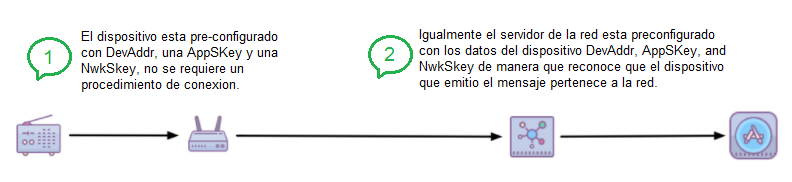
Este método de autenticación usa una clave y una firma, cada nodo dispone de una clave de aplicación única de 128-bits AppKey, esta clave se usa para generar una petición de conexión, esta petición no está encriptado sino firmada con la clave de aplicación, la petición también contiene valores únicos para el dispositivo DevEUI y para la aplicación AppEUI y un valor generado de forma aleatoria llamado DevNounce, el AppKey es usado para crear un código de integridad de mensaje (MIC), cuando el mensaje llega a su destino el servidor usa la clave para validar la integridad del mensaje, en ese caso el servidor genera 2 nuevas claves de 128-bits que serán usadas para enviar los demás mensajes. Por lo que se necesita un mensaje de confirmación para transmitir las 2 claves generadas. (Gresak & Voznak, 2020)



**Figura 2-3.** Activacion en el aire OTAA,   
tomada de (Raftery, 2018) adapatada por el autor

**Autenticación por personalización ABP**

En este método de autenticación las claves de sesión (NwkSKey y AppSkey) junto con la dirección del dispositivo (DevAddr) están definidas en el dispositivo, estas claves se codifican directamente en el mensaje que se trasmite y se enrrutan por medio del Gateway que reciba la señal al servidor de IoT, en donde se verifican si las claves y la dirección recibida corresponden a uno de los dispositivos que perteneces a la aplicación; en ese caso se procesan, adicionalmente se puede retornar un mensaje de confirmación, en caso contrario se omite el mensaje recibido. (Gresak & Voznak, 2020)



**Figura 2-3.** Autenticación por personalización ABP,   
tomada de (Raftery, 2018) adaptada por el autor.

## Bandas ISM

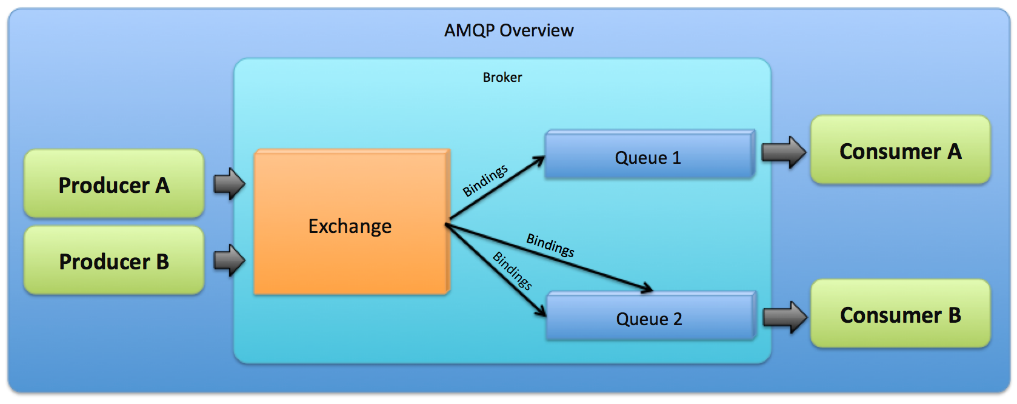
## Protocolos de comunicación nivel de aplicación.

Los protocolos de comunicación son estrategias que buscan que la comunicación entre las partes que componen un sistema sea eficiente, para ello la información debe ser clara, accesible, ordenada y que todas las partes del sistema puedan entenderla. De forma análoga en la comunicación humana cuando interactúan 2 individuos fácilmente puede establecer una secuencia de emisión y recepción alternando el tiempo de locución para establecer un dialogo, pero cuando existen múltiples individuos es necesario delegar la palabras a uno solo de ellos con el propósito de evitar el ruido y la confusión como en el caso de los debates.

En el caso de los sistemas el orden de la comunicación es fundamental para garantizar que las partes no solo reciben los mensajes que se les envían, si no además que los mensajes que reciben son solo los que les interesan y que tienen una secuencia lógica para evitar una mala interpretación de los mensajes y evitar el caos. A nivel de maquinas la importancia de los protocolos de comunicación es la interoperabilidad, que es la capacidad de comunicarse e intercambiar información para cooperar aunque estas no estén codificadas en el mismo lenguajes o estén programadas con la misma estructura. A continuación abordaremos 3 de los protocolos más populares en el ámbito de IOT.

### **Protocolo avanzado para encolamiento de mensajes (AMQP)**

Este protocolo centraliza su cadena en mensajes en un agente llamado Broker, que se encarga de realizar una retransmisión coherente de los datos, para ello crea una secuencia de mensajes permitiendo generar un operación asíncrona, esto quiere decir que el emisor y el receptor reciben los mensajes aunque estos no estén trabajando al mismo tiempo, de esta manera el destinatario pregunta por la información cuando tenga disponibilidad de hacerlo y el emisor puede continuar trabajando sin esperar que exista confirmación del receptor.



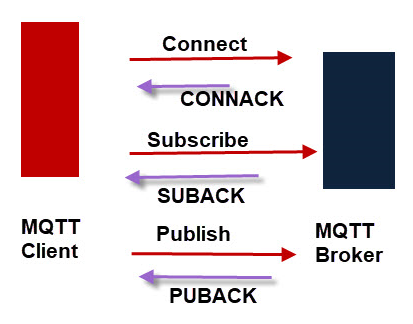
**Figura 2-3.** Secuencia de comunicación AMQP tomada de (VOLOV, 2016)

En AMPQ la emisión de un mensaje desencadena la secuencia de operaciones, luego de que el productor envía un mensaje, el bróker es el encargado de procesarlo, categorizarlo según la cola a la que pertenezca y distribuirlo atendiendo a las reglas que hayan sido definidas, el otro actor en esta comunicación es el consumidor quien se encarga de consultar si tiene mensajes pendientes por recibir, para ellos notifica al bróker cuales son las colas de su interés y el código de identificación del ultimo mensajes que recibió. A lo cual el bróker responde con una lista secuencial de mensajes que se han generado luego del último mensaje recibido por el consumidor para cada una de las colas de su interés. (1&1 IONOS, 2020)

### Transporte de telemetría por cola de mensajes (MQTT)

Este es un protocolo de mensajería estándar para IoT, diseñado como un sistema de publicación/ suscripción para el transporte de mensajes que resulta ideal para conectarse a dispositivos remotos, dejando una huella diminuta y aprovechando el ancho de banda. (MQTT,2020)

Bajo este esquema la información del sistema se centraliza en un servidor llamado Bróker, para hacer uso de los temas que son similares a las colas, los clientes envían un mensaje de conexión con la información necesaria para establecer la conexión: usuario, contraseña, identificación del cliente, a lo cual el bróker retornara un mensaje indicando si la conexión es exitosa o fallo. Una vez establecida la conexión, el cliente puede decidir si desea publicar información, para lo cual envía el contenido del mensaje y el tema al que corresponde o decide si se suscribe o revocar la suscripción a alguno de los temas que está siguiendo. Independientemente de la operación que elija el cliente siempre recibe confirmación por parte del bróker indicando e estado de la petición.



**Figura 1:** esquema de interacción cliente bróker en MQTT tomado de (Cope S, 2020)

El bróker MQTT usa un algoritmo llamado filtro con el cual selecciona los mensajes que son enviados a cada cliente basado en la organización jerárquica de los temas a los que los clientes se encuentran suscritos. el bróker mantiene los mensajes en la cola hasta que todos los clientes que se encuentran suscritos al tema confirmen que han recibido dicho mensaje.

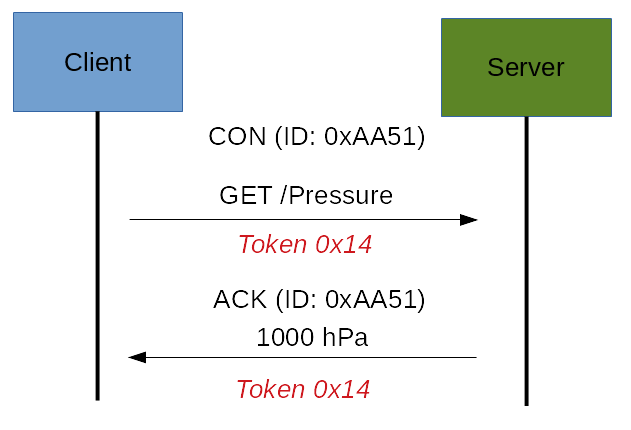
A diferencia de otros protocolos MQTT mantiene la conexión entre los clientes abierta y recicla dicha conexión para el envió de nuevos mensajes evitando latencias en la entrega de mensajes, para verificar dicha conexión cada cierto periodo de tiempo los clientes envían un paquete PINGRESP que es respondido con un paquete del mismo tipo por parte del servidor, esto permite que aunque la comunicación entre las partes sea asincrónica los mensajes están disponibles muy rápidamente lo cual lo hace bastante bueno para sistemas que requieren actualización constante de datos. (Hoplin, 2019)

### Protocolo para aplicaciones con restricciones (CoAP)

este protocolo fue diseñado para el intercambio de información en redes y dispositivos con restricciones de memoria, ancho de banda limitado baja potencia y baja disponibilidad, que se usa comúnmente en aplicaciones M2M, la analogía más inmediata para entender Coapa es pensar en una especie de HTTP para dispositivos IoT, es decir es una comunicación de tipo cliente servidor en el cual el cliente inicia una petición y espera una respuesta, sin embargo hay múltiples diferencias, la primera de ella es que este no usa TCP para el nivel de acceso a red, sino UDP que le permite el envío de mensajes asíncronos, los paquetes de CoAP son significativamente más pequeños que los de HTTP y están compuestos por un encabezado binario, una sección compacta de opciones y el contenido del mensaje.

**Modelo petición respuesta CoAP**

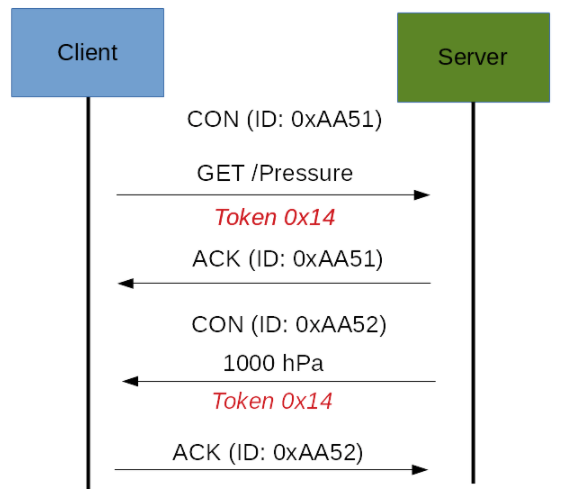
Este modelo es el segundo nivel de abstracción del protocolo, las peticiones se emiten con un número de identificación y una etiqueta de confirmación (CON) o no confirmación (NON), de manera que si el servidor está en la capacidad de responder inmediatamente una petición etiquetada con confirmación (CON) retornara un mensaje reconocimiento (ACK) con la respuesta o el código de error generado.



**Figura 1:** ejemplo respuesta inmediata req/resp CoAP tomado de (Azzola, F, 2019)

Como se observa en la imagen en CoAP un mensaje puede contener un token, que es una forma diferente de ID de mensaje que se usa para comparar la correpondencia entre la petición y la respuesta.

Si el servidor no puede dar respuesta inmediata a la petición envía un mensaje de reconocimiento (ACK) vacio, y apila la petición en una pila de pendientes, luego cuando puede responder la petición envía un nuevo mensaje de confirmación con la informacion requerida, en este caso es el cliente quien debe confirmar la recepción de la respuesta.



**Figura 1:** ejemplo respuesta posterior req/resp CoAP tomado de (Azzola, F, 2019)

## Seguridad en IoT (lineamientos OWASP)

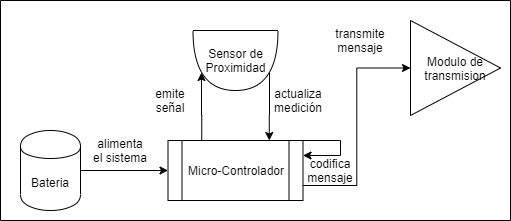
# Capítulo 1: Estado del Arte

Investigación y trabajos similares al que se está proponiendo

# Capítulo 3: Dispositivos de Medición

Uno de los elementos principales de un sistema de monitoreo es el nodo de medición, que es un dispositivo electrónico que se debe adaptar a los contenedores que están dispuestos en las calles para almacenar el residuo sólido y con el cual se logra estimar con un alto grado de precisión el nivel de llenado en cada contenedor. Con esta información se puede determinar si es necesario que dicho contenedor sea incluido en la planificación de las próximas rutas de recolección de la flota o su recolección puede ser postergada, con lo cual se puede hacer una administración de los recursos para maximizar su beneficio.

El dispositivo de medición que se compone de 4 elementos, un sensor de distancia que se encarga de medir el nivel de llenado, un micro controlador en el que almacenan las instrucciones y procedimientos que debe realizar el dispositivo, un módulo de trasmisión de radio encargado de modular y emitir la información de forma inalámbrica y una batería para alimentar el nodo. El siguiente diagrama de bloques muestra de manera abstracta la interacción de los componentes.



**Figura 3-1:** diagrama de bloques arquitectura del dispositivo de medición.

Opcionalmente el dispositivo puede disponer de un sensor GPS con el cual es posible determinar la ubicación del dispositivo sobre la superficie terrestre usando como referencia un sistema de coordenadas, aunque dado que los contenedores tienden a permanecer en el mismo lugar, no se requiere actualización constante de su ubicación, y gracias a que la base de datos que gestiona la información del sistema fue diseñada con un componente espacial, es suficiente registrar la ubicación geográfica del contenedor en el momento que se instala el dispositivo de medición para disponen de una georeferenciacion de los contenedores, lo cual disminuye aproximadamente en un 33% el costo del nodo.

A lo largo de este este capítulo se abordan algunas configuración de estos 4 componentes y soluciones que existen en el mercado con el propósito de definir cuál de ellas es la que mejor se adapta a las condiciones del sistema que se propone haciendo un balance entre costo, rendimiento y precisión.

**Sensor de distancia:**

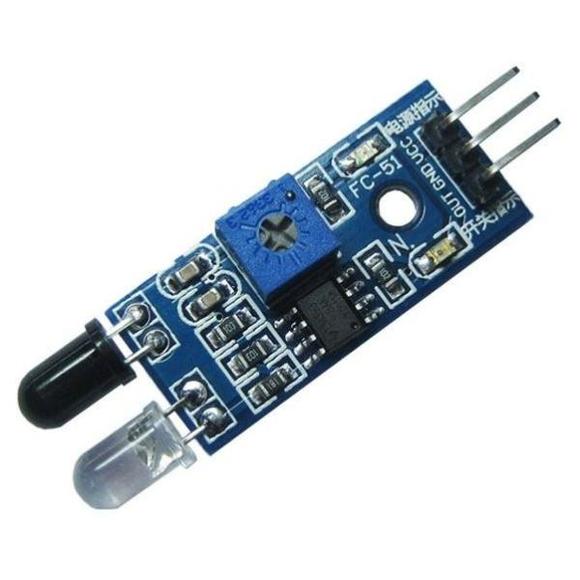
Es un dispositivo capaz de detectar la distancia hasta un obstáculo o cuerpo solido en un ángulo dirigido a continuación se abordan 3 de los sensores disponibles en el mercado.

**Sensor Laser Vl53l0x-Gyvl53l0xv2**

  
**Figura 3‑2:** sensor de proximidad laser Gyvl53l0xv2.

Este sensor emite una señal de láser[[2]](#footnote-2) y usa la reflexión para determinar la proximidad de los objetos, su alimentación es de 2,8 a 3,5 V y es un sensor de alta precisión que tienen una resolución de 1 milímetro y un rango entre 1 milímetro y 2 metros, pero su alcance efectivo depende de las condiciones ambientales como el ruido y la reflectancia del material del obstáculo, su costo aproximado es de 35.000 COP, es un circuito inter integrado I2C que se comunica por un bus de datos serial, se considera que su estructura sin cable y sus pequeñas dimensiones hacen que sea algo difícil de configurar, para el sistema que se propone.

**Sensor de luz infrarroja IR**



**Figura 3‑3:** sensor de proximidad infrarrojo FC-51.

Este sensor usa una señal de luz infrarrojo para reflejar el objeto, el sensor consta de un bombillo led que emite la señal y un detector de monitoreo de posición, que recibe el pulso de la luz reflejada y genera una señal análoga que depende de la ubicación del objeto reflectante. Su alimentación es de 3,3 V, existen varios dispositivos en el mercado y su rango está asociado a su valor comercial, el más económico cuesta 5.000 COP y su rango esta entre 2 y 30 cm, se pueden conseguir otros sensores cuyo costo de 104.000 COP y su rango esta ente 100cm y 550 cm. Estos rangos tampoco se ajustan muy bien a las dimensiones de los contenedores que se busca monitorear.

**Sensor de distancia por ultrasonido:**



**Figura 3‑4:** sensor de proximidadpor ultrasonido HC-SR04.

De manera análoga a los sensores basados en luz este tipo de sensor usa la reflexión de la señal para medir la distancia a un obstáculo, este sensor emite un sonido en una frecuencia imperceptible para el oído humano y mide el tiempo que tarda en regresar el sonido emitido convirtiendo el eco recibido en una señal eléctrica, para hallar la distancia se usa la constante de velocidad de propagación del sonido en el aire 334 m/s, empleando la siguiente formula

Este dispositivo el sensor más común de este tipo tiene la referencia HC-SR04, tiene una resolución de 1 cm y un rango entre 5 cm y 500 cm, su valor en el mercado es de 6.000 COP y dispone de 4 pines estándar que facilitan su conexión con placas de desarrollo como arduino o protoboard genéricas y tiene existen varias librerías de código abierto que permiten que su configuración sea fácil.

Para este trabajo se consideraron 3 configuraciones del nodo de medición en los cuales se usaron diferentes micro controladores y módulos de transmisión de radio frecuencia con el ánimo de encontrar de forma experimental cuál de estas configuraciones se adapta mejor al sistema de monitoreo, teniendo en cuenta factores como la distancia de comunicación, la atenuación de la señal y el costo.

A manera de resumen de extrajo la información en la siguiente tabla para hacer un análisis más rápido de los sensores revisados.

| **Característica** | **Laser** | **Infrarrojo** | **Ultrasonido** |
| --- | --- | --- | --- |
| *Referencia* | Vl53l0x-Gyvl53l0xv2 | DS - IR | HC-SR04 |
| *Voltaje de entrada (V)* | 2,8 a 3,5 | 3,3 | 5 |
| *Resolución* | 1 milímetro | Depende del costo | 1 cm |
| *Precisión* | Alta | NA | media |
| *Rango (alcance)* | 1 milímetro hasta 2 metros | Rango está asociado a su valor comercial el mas económico 2 hasta 30 cm, sin embargo otros ofrecen rangos de 1 hasta 5,5 metros. | entre 5 cm y 5 metros |
| *Precio (COP)* | $ 35.000 | el más económico cuesta $5.000 otros de $104.000 | $ 6.000 |

**Tabla 3‑1:** comparación sensores de proximidad.

Librerías usadas para la comunicación de los dispositivos con el servidor IOT

## **Arduino**



**Figura 3‑5:** Logo Arduino.

la definición oficial en la página Web de Arduino lo define como: una plataforma electrónica de código abierto, que tiene la capacidad de interpretar estímulos de entrada como la intensidad de luz por medio de un sensor, o la pulsación de un botón y a partir de esos estímulos desencadenar acciones por ejemplo encender un motor o enviar un mensaje. (Arduino, 2018) al componente hardware de arduino lo llamamos tarjeta, a la tarjeta se le pueden indicar como operar ante determinado estímulo de entrada por medio de un conjunto de sentencias que se tramiten al micro controlador que viene integrado en la tarjeta, por medio de una interfaz de bus serial que se conecta fácilmente al computador a través del puerto USB.

Arduino no solo está compuesto por el hardware dispone además de un lenguajes de programación de alto nivel y de un entorno de desarrollo llamado arduino studio basado en Processing[[3]](#footnote-3), una de las características más importantes de la plataforma es su comunidad, pues al tratarse de código abierto usuarios experimentados dan soporte a la plataforma y contribuyen desarrollado y publicado diferentes tipos soluciones, para el uso y beneficio de la comunidad, que pueden ser instalados como librerías desde el gestor de paquetes del entorno de desarrollo. Otras de las ventajas al usar arduino son:

**Económicas**: las tarjetas de arduino están construidas con componentes de buena calidad y bajo costo lo que hace que su costo en el mercado esté por debajo de los 50 dólares, lo cual es económico si se compara con otras tarjetas de desarrollo disponibles en el mercado. (Msv, J. 2019)

**Multiplataforma:** el hardware de arduino está diseñado para integrarse de forma sencilla con sistemas unix, Macintosh OSX y Windows.

**Entorno de desarrollo:** tiene un entorno de desarrollo muy sencillo, compuesto por una interfaz grafica en donde se puede escribir el código y controles para configurar la tarjeta sobre la que se esta desarrollando, como acciones para compilar el código, cargarlo al micro controlador y un monitor para hacer seguimiento a las operaciones que se ejecutan en la tarjeta.

**Software libre y extensible:** las librerías y ejemplos de arduino son publicadas como código abierto de manera que todos los desarrolladores pueden abrir el código que este en un lenguaje legible y modificarlo o ajustarlo de acuerdo a sus necesidades y el código estándar C puede ser incluido directamente en el programa de arduino.

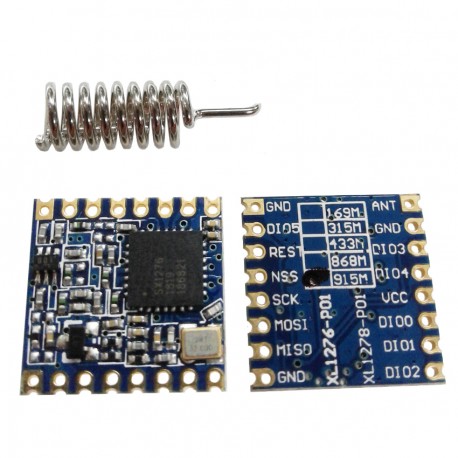
**Hardware libre y extensible:** todos los planos de las tarjetas arduino están publicados bajo la licencia de creative commons, para que los diseñadores de circuitos experimentados puedan hacer sus propias versiones del módulo extendiéndolas y mejorándolas, incluso usuarios sin muchas experiencia pueden revisar los planos de los circuitos y asi entender mejor cómo funcionan. (Arduino, 2018)

Una de las configuraciones de dispositivo de medición se ha elegido el **arduino UNO**: es una tarjeta compuesta con un micro controlador ATMega 328P tiene un conjutno de 14 pines de entrada o salida digital, 6 pines de entrada análoga y un resonador cerámico de 16 Mhz, un puerto de conexión USB, un puerto Jack para suministro de corriente, un puerto ICSP[[4]](#footnote-4), la tarjeta se puede conectar al computador por medio de un cable USB para empezar iniciar un prototipo de forma rápida. (Arduino Uno Rev3, 2017)

**Módulos de trasmisión:**

La compañía Semtech quien patento la tecnología de modulación LoRa, desarrollo un módulo electrónico que recibe la señal en bits y la modula en una de las frecuencias de onda de la banda ISM predefinidas por LoRa, se trata de la familia SX-12xx (SX1276, SX1277, SX1278, SX1279) los cuales incorporan el modem de espectro ensanchado, como se había visto previamente, logrando distancias mayores a las que se consiguen con técnicas de modulación como FSK o OOK, su sensibilidad esta 8 dB por encima de FSK pero usando un amplificador de bajo costo se puede incrementar la sensibilidad del receptor en más de 20 dB, este dispositivo modula la frecuencia constantemente y con esto consigue tener un selectividad de la señal y un bloqueo de interferencia que mejor la confiabilidad de la comunicación.

Estos módulos ofrecen alta flexibilidad en la configuración permitiéndole decidir al usuario parámetros del espectro como el ancho de banda (BW), factor de propagación (SF), tasa de corrección de error (CR) (semtech, 2015)



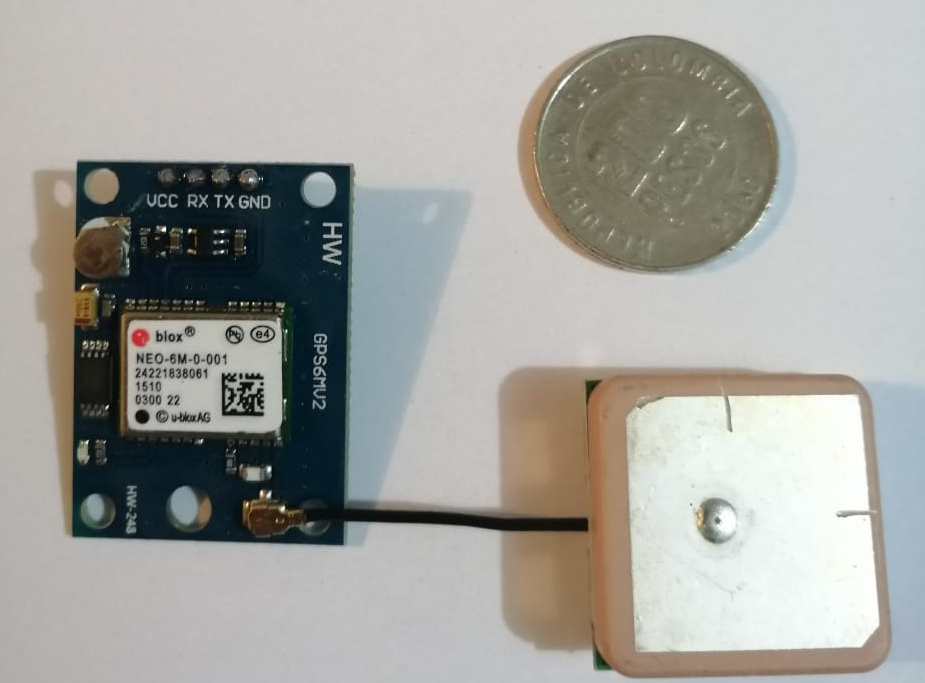
**Figura 3-6:**  módulo de transmisión SX1276

El modulo que se eligió para la primera configuración de dispositivos de medición es el SX1276 este modulo dispone de 14 pines dentro de los cuales se destacan los pines GND y 3,3V que son los que suministran energía al módulo, los pines de la interfaz periférica serial SPI[[5]](#footnote-5): SCK reloj serial, MOSI salida del mestro entrada del esclavo, MISO entrada del maestro salida del esclavo, NSS selección del esclavo, el pin de reinicio Reset y los pines de entrada digital DI0, DI1, DI2, demás dispone de un espacio para la conexión de una antena. Lo particular en este modulo es que las resitencias, los diodos y los condensadores que la componen están calibrados para maximizar la señal en 915 Mhz.

El modulo es muy pequeño mide 2cm X 2,7 cm, la distancia entre sus pines en muy poca y no se ajusta a una regleta de pines estándar lo que no solo dificulta se soldadura, sino también su montaje en instrumentos genéricos como la protoboard W-102 de manera que para poder ensamblarla se requiere de adaptarla a una baquela universal.

Para la ensamblar el primer circuito del dispositivo de transmisión se usaron los 3 componentes mencionados en este capítulo: arduino uno, módulo SX1276 y sensor HC-SR04

**Módulos de Geo localización:**

****

**Figura 3-8:** Modulo de localización Neo 6M Ublox

Este módulo utiliza tecnología Ublox de triangulación de señal satelital para detectar la ubicación espacial con una alto grado de precisión, dispone de una antena cerámica de 2,5 x 2,5 cm que genera alta ganancia, dicha antena se conecta al módulo por medio de un puerto UFL el dispone de 4 pines TX, RX, VCC, GND, y tiene un voltaje de operación de 3,3 V se usa con mayor frecuencia el módulo TX puesto que para conocer el posicionamiento solo se requiere la lectura de las sentencias NMEA[[6]](#footnote-6) que genera el Chip, las cuales tienen la siguiente estructura.

**$GPRMC,044235.000,A,4322.0289,N,00824.5210,W,0.39,65.46,020615,,,A\*44**

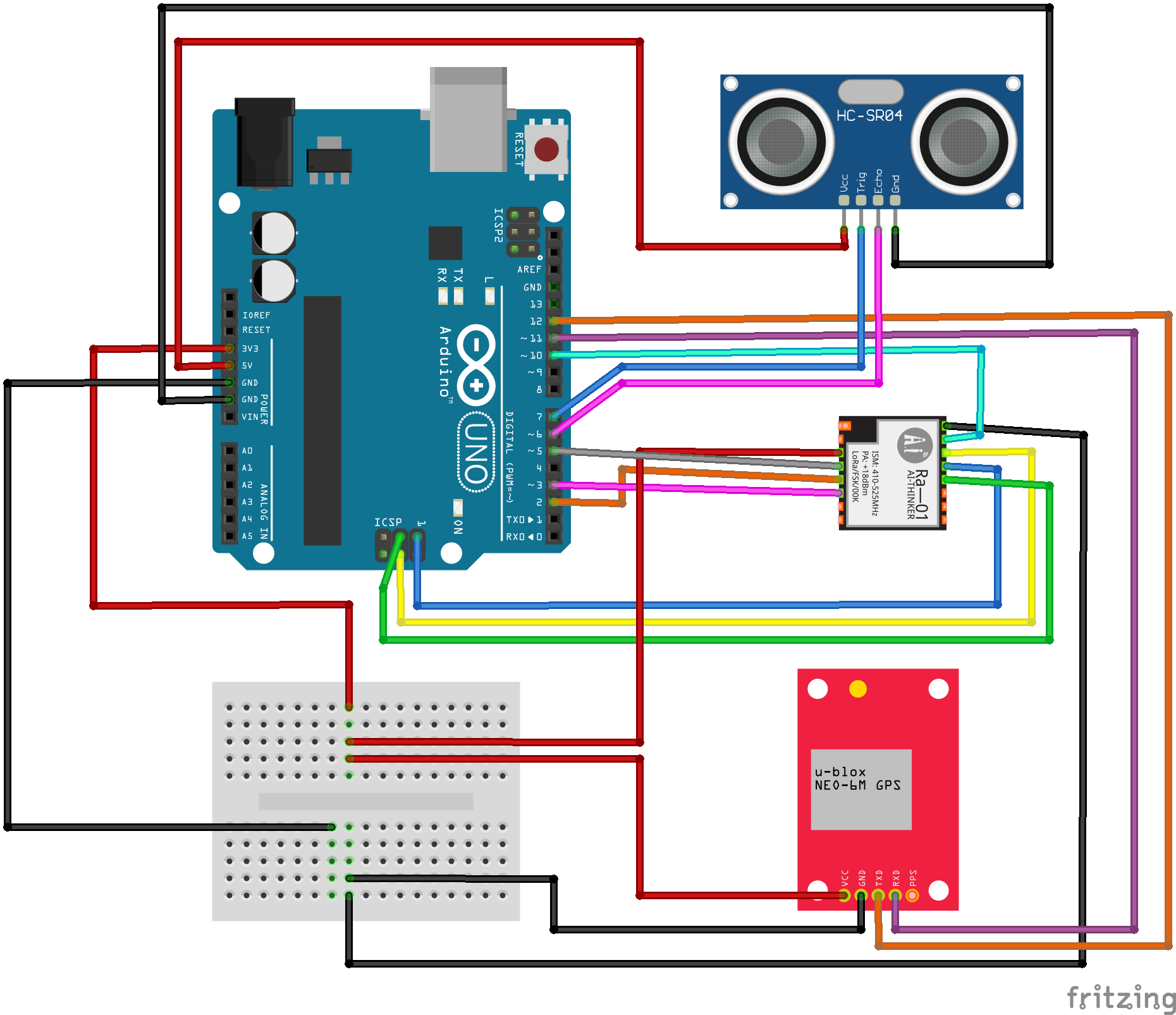
Que según el protocolo MNEA ya mencionado se pueden interpretar asi:

* **044235.000**representa la hora GMT (04:42:35)
* **“A”**es la indicación de que el dato de posición está fijado y es correcto. “V” sería no válido
* **4322.0289**representa la longitud (43º 22.0289´)
* **N**representa el Norte
* **00824.5210**representa la latitud (8º 24.5210´)
* **W** representa el Oeste
* **0.39** representa la velocidad en nudos
* **65.46** representa la orientación en grados
* **020615** representa la fecha (2 de Junio del 2015)

Información tomada de (Naylampmechatronics, 2016)

El modulo dispone de un bombillo led, el cual se mantiene apagado hasta que el modulo reconozca la localización, fecha y la hora, una vez encuentre esta información comenzara a prender y apagar de forma intermitente. Su velocidad de puerto serial es configurable desde los 4800 hasta los 115200 baudios pero se recomienda operarlo en 9600 baudios.

### Conexión de los componentes físicos



**Figura 3-7:** esquema de conexión Arduino, SX1276 neo 6M y HC-SR04.

Para la conexión de los componentes del primer módulo de medición se uso la configuración que se muestra en la siguiente tabla

| **Arduino** | **SX1276** |
| --- | --- |
| *3,3 V* | 3,3 V |
| *GND* | GND |
| *DI0* | D2 |
| *DI1* | D3 |
| *MISO* | MISO |
| *MOSI* | MOSI |
| *SCK* | SCK |
| *D10* | NSS |
| *D5* | Reset |

**Tabla 2-1:** conexiones de arduino uno a SX1276

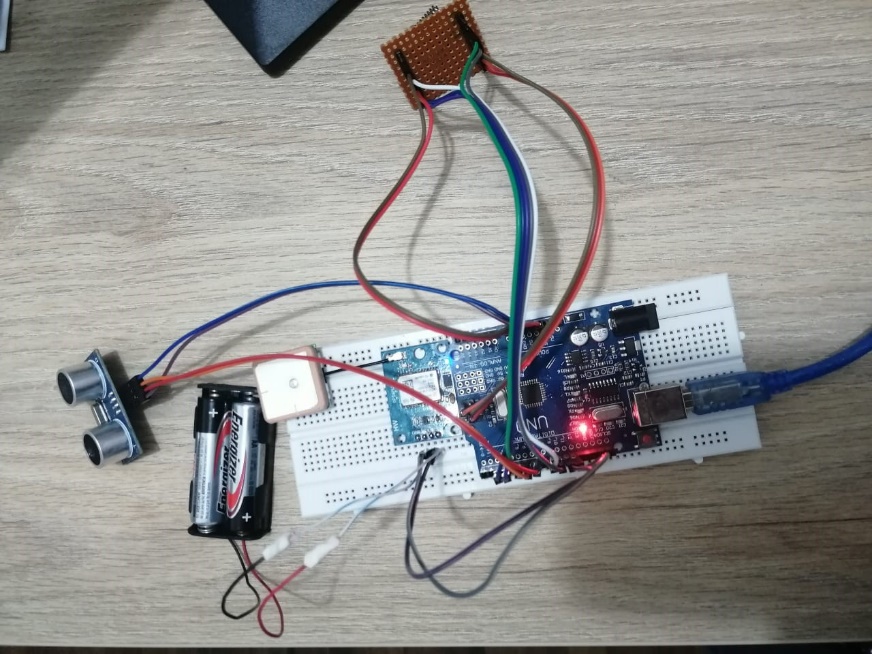
| **Arduino** | **HC-SR04** |
| --- | --- |
| *5 V* | 5 V |
| *GND* | GND |
| *D6* | Echo |
| *D7* | Trigger |

**Tabla 2-2:** conexiones de arduino uno a HC-SR04

| **Arduino** | **Neo-6M** |
| --- | --- |
| *3,3 V* | 3,3 V |
| *GND* | GND |
| *D11* | RX |
| *D12* | TX |

**Tabla 2-3:** conexiones de arduino uno a HC-SR04

Luego de haber realizado las conexiones con la configuración descrita anteriormente se obtuvo un circuito ensamblado como se puede ver a continuación

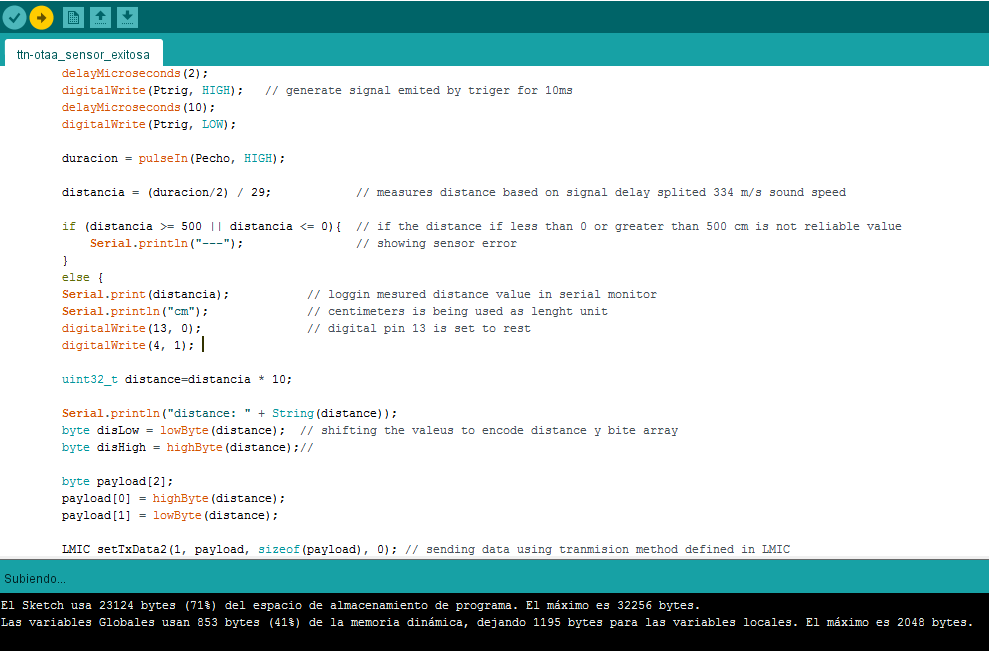


**Figura 3-8:** ensamblado de los componentes físicos configuración 1

### Programación lógica del micro-controlador

Los programas en arduino se componen de 2 secciones principales una llamada setup donde se configuran las variables, librería y objetos que se usaran en el programa, la otra llamada loop donde se incluyen instrucciones que se ejecutan repetidamente mientras la placa esté conectada a una fuente de energía.

Para empezar se importan las librerías que se instalaron LMIC y SPI, en la sección de setup se deben definir cuáles serán los pines de la tarjeta que usaremos para enviar o recibir señales, para esta configuración se han definido los pines 6 como entrada y 7 como salida que son los que están conectados al sensor de proximidad. La sección de loop ha sido reemplazada por una función llamada do\_send, que será la encargada de enviar el mensaje. Esta sección contiene el código que se ve a continuación.



**Figura 3-9:** segmento del programa en el entorno de arduino estudio.

El programa emite una señal durante 10 microsegundos y cuenta el tiempo que tarda esa señal en retornar al pin, la distancia se calcula dividiendo la duración en 2 pues esta tarda el doble dado que debe ir hasta encontrar el obstáculo y regresar, luego este tiempo se multiplica por la velocidad del sonido. Por otra parte se uso la librería TinyGPS para administrar los datos del sensor de ubicación y la librería SoftwareSerial de arduino para adaptar 2 pines como buses seriales para recibir y transmitir datos los cuales fueron el pin 11 para recepción (RX) y 12 para transmisión (TX).

La distancia obtenida es codificada en un arreglo que contiene el bit mayor y el bit menor, y las coordenadas fueron multiplicadas por 10 mil para quitar los puntas decimales y codificados con la misma estrategia en las posiciones 0 a la 5 del mismo arreglo, finalmente este arreglo es el que se pasa como carga útil (payload) del paquete que será enviado usando la librería por medio de la función para asignar los datos de transición (LMIC\_setTxData2).

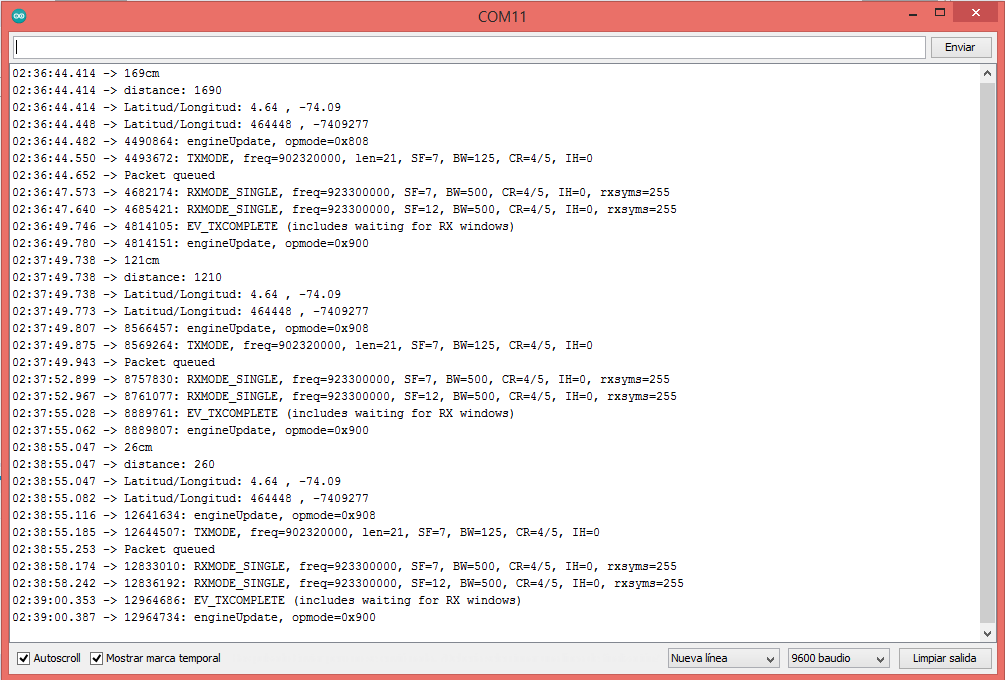
Para poder almacenar el programa en el micro controlador se conectó la tarjeta Arduino uno al puerto USB de un computador, se inició el entorno de desarrollo y se configuro la tarjeta que se conectó, en este caso Arduino Uno, y se seleccionó el puerto por el cual estaba conectado la tarjeta



**Figura 3-10:** configuración de la tarjeta en el entorno de desarrollo.

### Pruebas de emisión de los mensajes

Asistidos por el monitor serial que provee el entorno de desarrollo como herramienta de visualización vamos a revisar lo que ocurre dentro del micro controlador. En la siguiente imagen se pueden observar la distancia y las coordenadas geográficas obtenidas y su valor a codificar, luego se activa el módulo de radio en modo de transición y se emiten 2 pulsos en la frecuencia de los 902 MHz uno con el factor de propagación (SF) en 7 y otro con este factor en 12, los paquetes quedan en cola, dado que aún no se ha activado ninguna puerta de enlace que comunique el nodo con internet, sin embargo podemos constatar que el nodo está intentando emitir los mensajes por la banda de frecuencias seleccionada lo cual es el resultado esperado con esta prueba.



**Figura 3-11:** verificación de las operaciones en el micro controlador desde el monitor serial en Arduino.

## Dragino



**Figura 3-11:** Logo Dragino

Dragino es uno de los fabricantes de hardware mejor posicionado en transceptores con modulación lora, tiene una amplia gama de dispositivos para varias necesidades en su página oficial se encontró la siguiente descripción de la compañía:

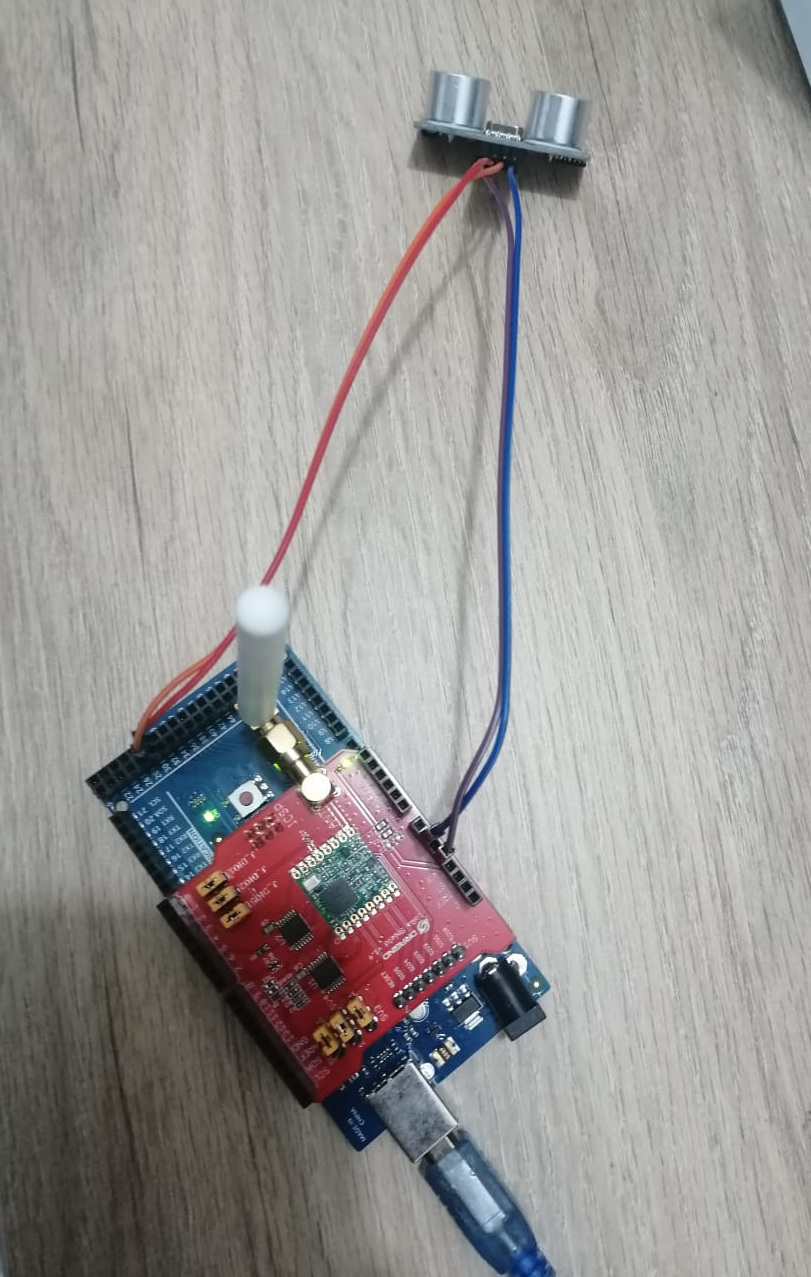
“*Dragino se ubica en Shenzhen, China, que es uno de los centros de diseño, producción, manufactura y abastecimiento de componentes electrónicos del mundo. Dragino se especializa en la Industria IoT y continuamente evalúa nueva tecnología para mejorar la experiencia del usuario en soluciones IoT, ofrecen un conjunto de productos para diseñar, probar y producir productos innovadores con IoT, disminuyendo el tiempo, reduciendo los costos y mitigando los riesgos.*” (Dragino, 2015)

Dragino ofrece productos ensamblados que eventualmente podrían ser adaptados para la aplicación que se está proponiendo, por ejemplo: el sensor de distancia LDDS75 LoRaWan, que integra un sensor de ultrasonido y un módulo LoRa preconfigurados, además dispone de una batería de 4000 mA en litio pensada para durar hasta 10 años (*Dragino Distance*, 2020) o el dispositivo de seguimiento LGT-92 que integra un micro controlador de bajo consumo, un módulo de posicionamiento L70, un acelerómetro de 9 ejes para identificar los tipos de movimientos que realizan los objetos todo esto alimentado por una batería de recargable 1000 mA diseñado para la implementación aplicaciones de cadena de suministro o control de tráfico en ciudades inteligentes. (*Dragino GPS, 2019*)

Lamentablemente dragino no ofrece un equipo que integre todos los sensores que se propusieron en la arquitectura del nodo de mediciones para este sistema, de manera es posible simplemente comprar una solución ensamblada (on the shell), sin embargo Dragino ofrece equipos que se pueden acoplar fácilmente con placas de desarrollo estándar y facilitan la construcción de estas soluciones.

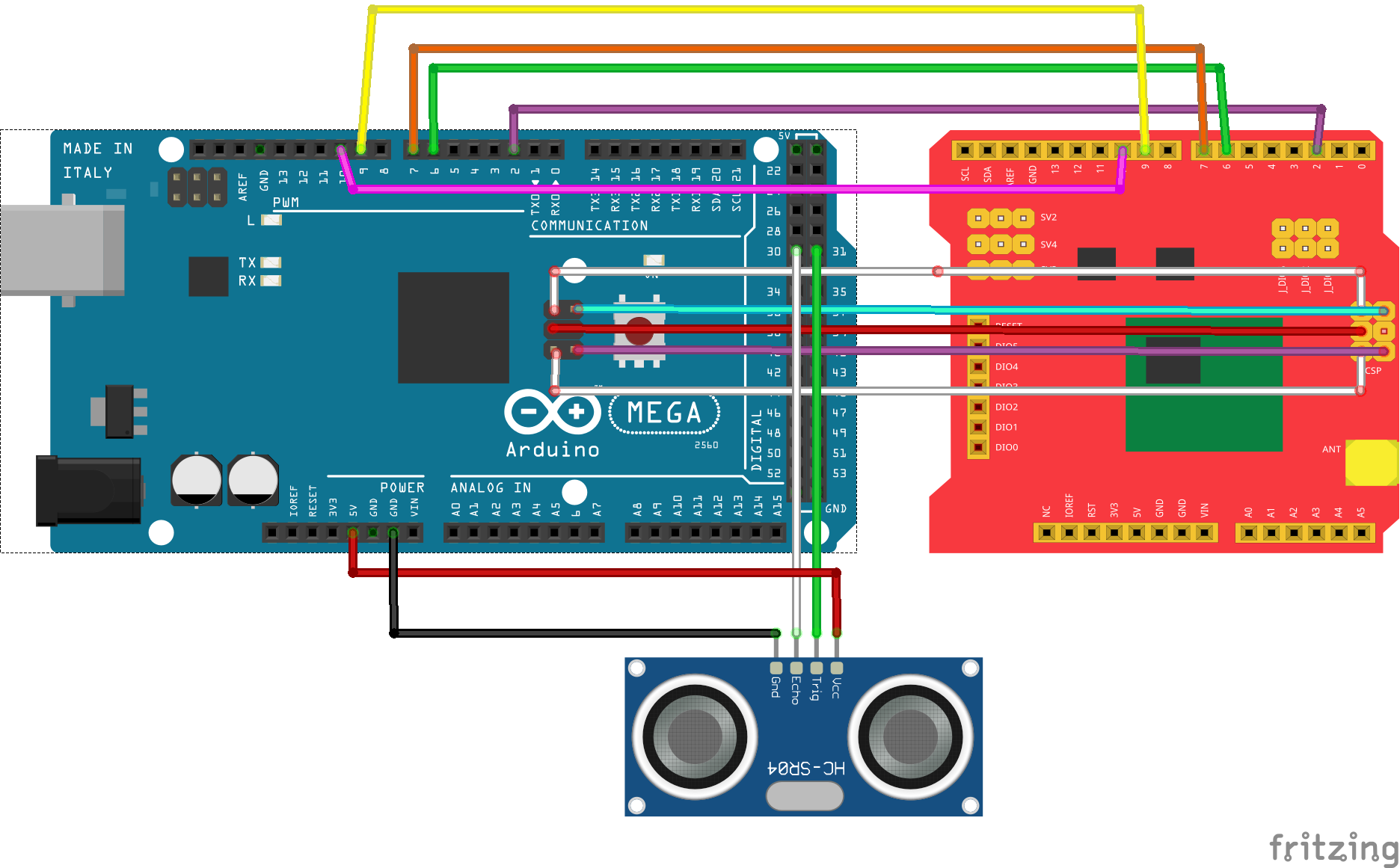
**Dragino Shield:** es una placa en la que viene integrada con un módulo SX12XX, un amplificador de 20 dB y una antena la función de la placa es la de hacer un puente entre los pines del módulo y los pines de arduino, la placa es compatible con las versiones: UNO, Mega, Leonardo y DUE, sobre esta placa será posible integrar los componentes definidos en la arquitectura para nuestro dispositivo de medición.

En este experimento se decidió usar un arduino Mega que dispone de un microprocesador más potente que el arduino uno y con un poco más de memoria la cual será necesaria para grabar las instrucciones que usaremos



**Figura 3-11:** integración de Dragino Shield y Arduino Mega 2560

### Conexión de los componentes físicos



**Figura 3-12:** esquema de conexión Arduino Mega, Dragino Shield y HC-SR04

En este caso la conexión de los componentes resulta más simple dado que se requirió únicamente la superposición de las placas arduino y dragino, sus pines se encuentran perfectamente acoplados, luego usando conectores Jumper Dupont de macho a hembra se conectó el sensor de Proximidad usando la siguiente configuración de pines

| **Arduino** | **HC-SR04** |
| --- | --- |
| *5 V* | 5 V |
| *GND* | GND |
| *D30* | Echo |
| *D31* | Trigger |

**Tabla 2-X:** conexiones de Dragino a HC-SR04

Para simplificar esta configuración se prescindió de la conexión del módulo de ubicación GPS por razones que serán expuestas en la sección 4 de este capítulo. Sin embargo se obtuvieron las coordenadas aproximadas del sensor usando el servicio de mapas de Google y se codifico dicho dato en la trama que se va a enviar para poder reutilizar las funciones de codificación y decodificación que se implementaron en la primera prueba.

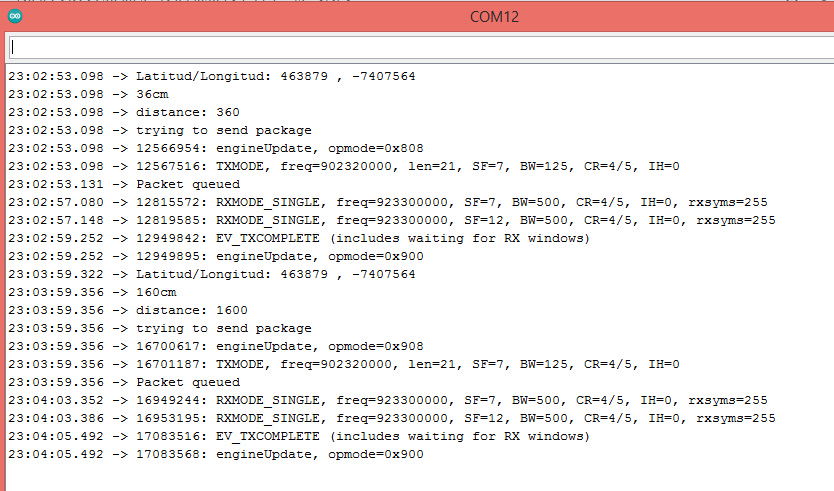
**Programación lógica del micro-controlador:** para programar el micro controlador se usó la estructura que se utilizó en el primer nodo de medición con la diferencia que en este; se sustitutillo la sección de la lectura de los datos de módulo de GPS por dos valores constantes de latitud y longitud, la definición de los pines del sensor de proximidad fueron ajustados por los numero 30 para echo y 31 para trigger y el mapeo de los pines entre la placa dragino y arduino quedo de la siguiente manera.

| **Arduino** | **Dragino** |
| --- | --- |
| *D2* | DIO 0 |
| *D6* | DIO 1 |
| *D7* | DIO 2 |
| *MISO* | MISO |
| *MOSI* | MOSI |
| *SCK* | SCK |
| *D10* | NSS |
| *D9* | Reset |

**Tabla 2-X:** entre arduino Mega 2560 y Dragino Shield

### Pruebas de emisión de los mensajes

Nuevamente asistidos por el monitor de puerto serial identificamos las operaciones que ocurren dentro del micro controlar, en la parte superior se puede visualizar la coordenada geográfica que fue simulada, esta vez no fue necesario multiplicarla por 10 mil dado que simplemente se incluyó sin cifras decimales, a continuación podemos observar el dato recuperado por el sensor de proximidad el cual multiplicamos por 10 con el propósito de eliminar las comas decimales y hacer más fácil la codificación del dato, luego el módulo de radio entra en modo de trasmisión que emite en 902 MHz y usa un factor de propagación de 7, el paquete queda en estado encolado y el modulo entra en modo de recepción en la frecuencia de 923 MHz con factores de propagación en los 7 y los 12, genera una ventana de tiempo en la cual espera confirmación del servidor IoT el cual aún no ha sido configurado, y se repite la secuencia. Lo cual es el resultado esperado en esta prueba.



**Figura 3-11:** verificación de las operaciones en el micro controlador desde el monitor serial en Dragino.

## TTGO

Lilygo es otra compañía productora de dispositivos electrónicos enfocada en soluciones con internet de las cosas, tiene su centro de operaciones en la provincia de Shenzhen en Hong Kong en su página oficial encontramos una descripción de la compañía:

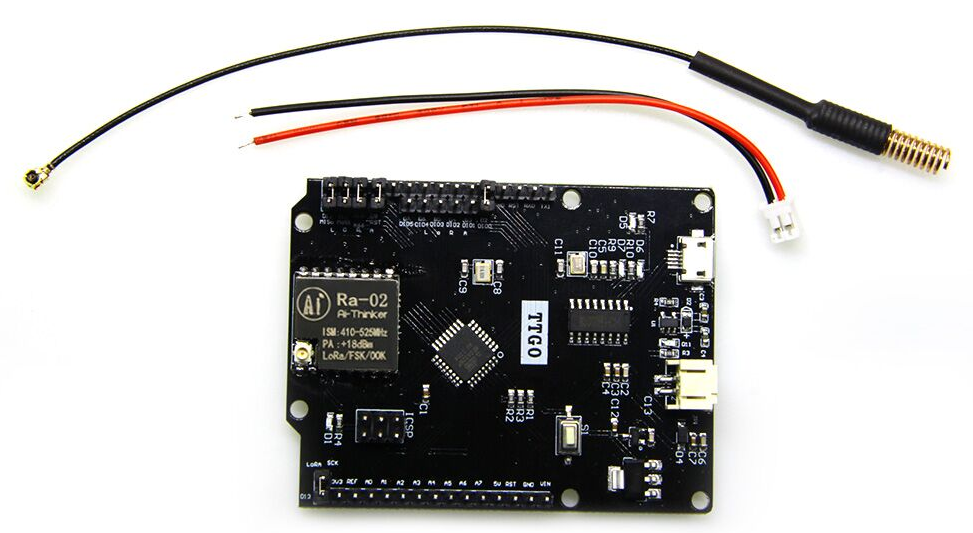
“*dedicados al desarrollo de IoT, hacer que los desarrollos sean más fácil es el concepto de producto de Xinyuan, lanzamos series de productos de código abierto con diversos tipos de micro controladores y módulos IoT, hasta kits educativos para hacer posibles los desarrollo y convertir ideas creativas en realidades y se puedan compartir, hacer que todos podamos explorar el mundo y conectar los objetos es la visión de Xinyuan*.” (Lilygo-Xinyuan,2017)

Algunos de los productos destacados de Lilygo incluyen carros de jugete controlados remotamente por relojes inteligentes, kits para el monitoreo de la frecuencia cardiaca y kits de sensores que se ensamblan a una pantalla led (reloj) como fichas de lego, haciéndolo fáciles de conectar y muy didácticos para la pedagogía electrónica en jóvenes.

**Figura 3-11:** Kit de sensores de múltiples funciones y reloj inteligente de Lilygo.   
(Tomada página oficial Lilygo)

**TTGO Para Arduino UNO LoRa MEGA328 433MHZ SX1278**

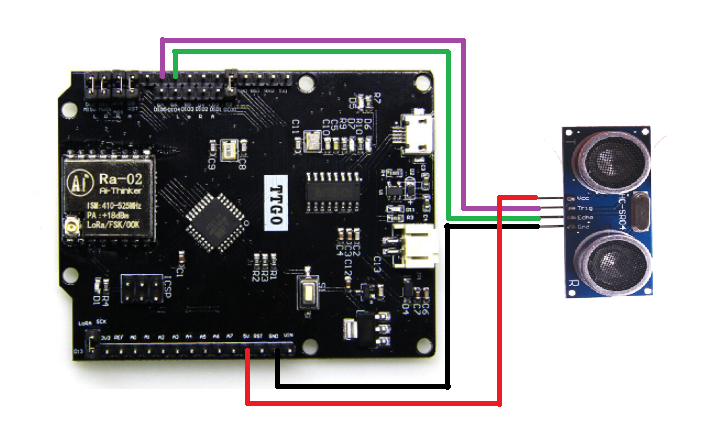
****

**Figura 3-11:** tarjeta TTGO compatible con arduino SX1278.  
(Tomada página oficial Lilygo)

Es una tarjeta que es compatible con la configuración de pines de arduino uno y se puede programar desde el entorno de desarrollo de arduino uno, es interesante porque la misma placa integra en micro controlador y el módulo de transmisión, lo cual facilita mucho el trabajo, dado que viene ensamblado de la misma forma que se definió en la arquitectura del nodo de medición y los pines sobrantes pueden ser usados para conectar sensores como el de proximidad o el de GPS

una de las particularidades fue que en el momento en el que se ordenó este artefacto aún no se tenía clara la frecuencia de las bandas autorizadas en Colombia y por un error se pidió la placa que integra el módulo SX1278 el cual opera en una frecuencia en la banda de 433 MHz, aun así se decidió usar este para hacer la prueba de concepto siendo conscientes que obtendríamos buenos resultados únicamente cuando el nodo estuviese próximo al Gateway, pero que este mismo procedimiento puede ser replicado usando a una de las placas con módulos que operan a 915 MHz. Como el modulo “TTGO SX1276 LoRa ESP32 433/470MHz 868 / 915MHz Bluetooth WI-FI Internet Antenna Development Board”

### Conexión de los componentes fisicos



**Figura 3-12:** esquema de conexión TTGO y HC-SR04

En esta configuración del Nodo de medición no hacen falta el ensamble de piezas adicionales gracias a que el módulo de trasmisión y el micro controlador están conectados directamente en el circuito de la tarjeta de desarrollo lo único que hace falta para satisfacer la arquitectura propuesta es conectar el sensor de proximidad lo cual podemos hacer fácilmente con la siguiente configuración de pines

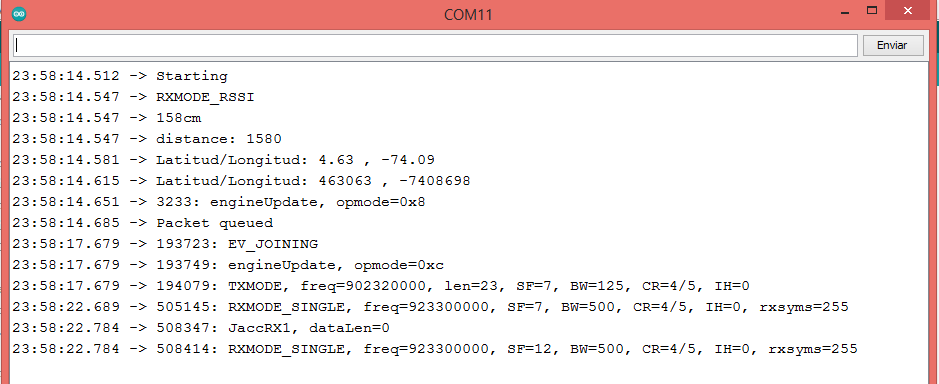
| **TTGO** | **HC-SR04** |
| --- | --- |
| *5 V* | 5 V |
| *GND* | GND |
| *D6* | Echo |
| *D7* | Trigger |

**Tabla 2-X:** conexiones pines de TTGO y HC-SR04

**Programación lógica del micro-controlador:** en esta ocasión se usó el programa que se había escrito para el nodo uno, fue necesario hacer algunas adaptaciones los pines del sensor de proximidad fueron definidos en 7 para echo y 6 para triger, se reemplazó el código del sensor con el que se obtenían las coordenadas geográficas por 2 valores constantes para latitud y longitud, y el mapeo de pines entre el micro controlador y el módulo de trasmisión quedo de la misma manera que en el nodo 2 con la única diferencia que se deshabilito el DIO 2.

### Pruebas de emisión de datos

En este caso podemos observar una operación muy similar a la de los nodos anteriores dado que el programa que se grabó en el micro procesador usa la misma secuencia para gestionar los recursos. En este caso ocurre una pequeña variación que es muy importante en las próximas etapas del proceso de comunicación. Tras haber obtenido los datos del sensor de proximidad y simular las coordenadas geograficas, el módulo de radio encola el mensaje, se hace énfasis que es este caso se puede leer el mensaje **EV\_JOINING**, esto indica que el mensaje ha sido recibido por un Gateway y que se inició un protocolo de autenticación con el servidor de IoT para iniciar una comunicación, lo cual se abordara en profundidad en el capítulo 5.



## Comparación de las opciones:

| **Criterio** | **Arduino** | **Dragino** | **TTGO** |
| --- | --- | --- | --- |
| *Costo (COP)* | $97.000 | $170.000 | $80.000 |
| *disponibilidad* | Disponible | Importación de USA | Importación de China |
| *Ensamble* | Complejo | Medio | Sencilla |
| *Funcionalidad* | Bueno | Bueno | Bueno |

**Tabla 2-X:** tabla comparativa de los nodos de medición.

Haciendo un balance entre las opciones que se validaron consideramos que resulta más conveniente importar placas de la marca Lilygo desde china no solo porque su costo es inferior a las otras 2 opciones, sino además porque facilita considerablemente el ensamble de los componentes gracias a que el micro controlador y el módulo de transición vienen conectados de fábrica, el único aspecto que está en su contra es la disponibilidad lo cual no se considera, un problema pues si bien la tarjeta puede tardar un poco en llegar al ser importada de china, se ha constatado constatado que el proveedor es confiable y está en la capacidad de hacer entregas en Colombia. una evaluación de los dispositivos se hace en el capítulo 5 en el cual se incluyen dentro de la decisión el criterio distancia de cobertura que es uno de los criterios más relevantes para el estudio.

## Librería Arduino-LMIC de Matthijs Kooijman

La librería LMIC fue creada para manejar los parámetros de la comunicación entre los dispositivos de medición y el servidor de IoT del cual hablaremos en el capítulo 5, esta librería es de código libre bajo una licencia Eclipse Public License 1.0. La librería se puede instalar fácilmente desde el gestor de librerías en arduino estudio. Pero también esta publicada en la cuenta de github del autor (Matthijs kooijman, 2015)



**Figura 2-x:** instalación de la Librería desde el gestor de librerías de arduino studio.

Esta librería ofrece una implementación completa para dispositivos LoRaWAN de clase A y clase B, soportando las bandas de 868 y 915 dentro de las características que ya han sido validadas están:

* El envió de paquetes ascendentes, considerando los ciclos de trabajo.
* La encripcion de mensajes y verificación de integridad
* La recepción de mensajes de confirmación en la primera ventana de tiempo RX2
* Personalización de las frecuencias y parámetros de tasa de datos
* La conexión por metodo OTAA

Hay algunas características que aún están pendientes por verificacion, entre ellas: la recepción de paquetes descendentes en la primera ventana de tiempo RX1, la recepción y procesamiento de comandos MAC, y la operación de dispositivos de clase B.

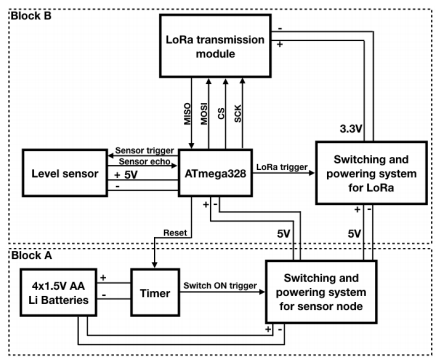
La comunidad generada en torno a la librería se ha encargado de mejorar continuamente no solo reportando los errores que obtienen con los diferentes tipos de hardware, si no registrando y documentando las soluciones que han encontrado para resolver los inconvenientes lo cual ha permitido que la librería madure muy rápidamente, para ofrecer una operación robusta pero además para ganar versatilidad con otras tarjetas menos populares que arduino.

## Consideraciones adicionales de nodo de medición

Las configuraciones y pruebas que se realizaron con los nodos de medición previamente expuestos son apenas una prueba de concepto de para validar la funcionalidad y operatividad del dispositivo, sin embargo para adaptar esos nodos en un entorno productivo real, aún se requieren resolver los aspectos que se mencionan a continuación.

**Necesidad de recubrimiento externo:** dado que los dispositivos estarán expuestos a condiciones climáticas normales como el sol y la lluvia, que posiblemente causen daños en los circuitos. Se requiere diseñar un recubrimiento externo o empaque hermético para evitar filtraciones de agua, se sugiere que sea en una pasta dura para proteger el dispositivo de golpes y que tenga un compartimiento independiente para poner las baterías y reemplazarlas fácilmente sin tener que tocar los demás componentes del circuito.

**Optimización del uso de la batería:** dado que la batería es uno de los componentes que debe ser mejor administrado para reducir el tiempo de mantenimiento de los nodos, de manera que es necesario analizar la forma de disminuir el consumo de la tarjeta, bien sea reemplazándola por una de menor consumo, poniéndola en modo de hibernación (sleep) durante los periodos que no se está realizando mediciones o sustituyendo el regulador de voltaje por un módulo de potencia de ultra bajo consumo encargado de activar la tarjeta únicamente para tomar y transmitir las mediciones y luego apagar la placa completamente (Cerchecci et al., 2018)o el diseño de un circuito que transforme la energía solar en corriente eléctrica por medio de una celda fotovoltaica como proponen (Aslam et al., 2020)



**Figura 2-x:** diagrama de bloques para nodo de bajo consumo tomado de (Cerchecci et al., 2018).

**Prescindir del módulo de localización:** teniendo en cuenta que los contenedores de basura raras veces cambian de ubicación, en lugar de ello tienden a mantenerse constantemente en el mismo sitio y que esquema de datos del sistema está dotado de un componente espacial, es posible obtener emparejar la ubicación del nodo y la del contenedor en el que se instale y persistir esta información directamente en la base de datos para tener un sistema de información geo referenciado, lo cual no solamente disminuye el costo de producción del nodo final aproximadamente en un 33%, sino que permite ser más eficiente en el momento de tomar las mediciones, dado que el módulo de GPS puede tardar de 1 a 3 minutos triangulando los datos de los satélites cuando lleva algún tiempo inactivo, y esto afectara directamente la eficiencia de la batería.

# Capítulo 3: RIS y Arquitectura IoT

Indiscutiblemente el nodo de medición es un elemento importante en una red inalámbrica de sensores RIS, pero es solo uno de los eslabones de la cadena, además de los nodos existen otros 3 componentes imprescindibles en una LPWAN: Los Gateway que se encargan de mantenerse escuchando los mensajes de radio emitidas por los nodos, se encargan de desmodular dichas señales transformarlas en pulsos digitales, posteriormente la versión digital del mensaje es en-rutada a través de internet hacia el siguiente componente. El servidor que es el encargado brindar servicios para gestionar la información recibida, validando su integridad, pertinencia y publicando esta información, en el tercer componente, un intermediario que llamaremos bróker y cuya responsabilidad principal es la de notificar dicha información a las aplicaciones. En este capítulo se abordaran de manera detallada la configuración y puesta en marcha de cada uno de estos componentes como parte de la implementación del sistema de monitoreo para la planificación de rutas de recolección de RSU.

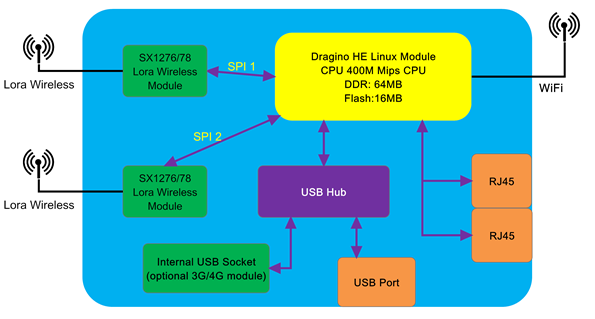
## **Gateway**

También conocidos como puertas de enlace son dispositivos electrónicos que actúan como puentes que unen 2 protocolos de comunicación por medio de la decodificación de señal. En el caso concreto de LoRaWAN los Gateway reciben mensajes emitidos por los nodo finales son modulados usando LoRa y que son recibidos usando componente físico llamado concentrador, cada uno de los mensajes el decodificado en forma de bits y se repite hacia el servidor de red usando un esquema IP.(Santos Filho et al., 2020)

**Existen varios tutoriales que indican detalladamente el procedimiento para construir Gateway artesanales en los que con una tarjeta de desarrollo multipropósito como una Raspberry pi 3 y un concentrador de señal se puede ensamblar el dispositivo, él además se han creado software para permitir configurar las frecuencias de trasmisión y la ruta servidor en el que se procesaran los mensajes. Con este tipo de soluciones también se hace necesario construir una antena con ciertas especificaciones técnicas para amplificar la ganancia de la señal en el Gateway, no obstante encontramos una solución lista que ya viene ensamblada y dispone de una interfaz de configuración por un precio cómodo por lo cual decidimos usar este componente dentro de nuestra red en lugar de gastar tiempo y dinero experimentando con la fabricación de un Gateway artesanal.**

### **LG02 de Dragino.**

**El dispositivo elegido para usar en nuestra red fue el Gateway de 2 canales de LG02 de Dragino, el cual es capaz de enlazar una señal LoRa conectándose a internet a través de WiFi, Ethernet, o de manera celular en la red 3G/4G usando un módulo LTE, esta cantidad de interfaces proveen buena flexibilidad para que los usuarios conecten sus nodos a internet, el dispositivo usa software Linux para controlar los 2 módulos 1276 este le permite operar en modo full dúplex**[[7]](#footnote-7) **incrementando la eficiencia de la comunicación.**

****

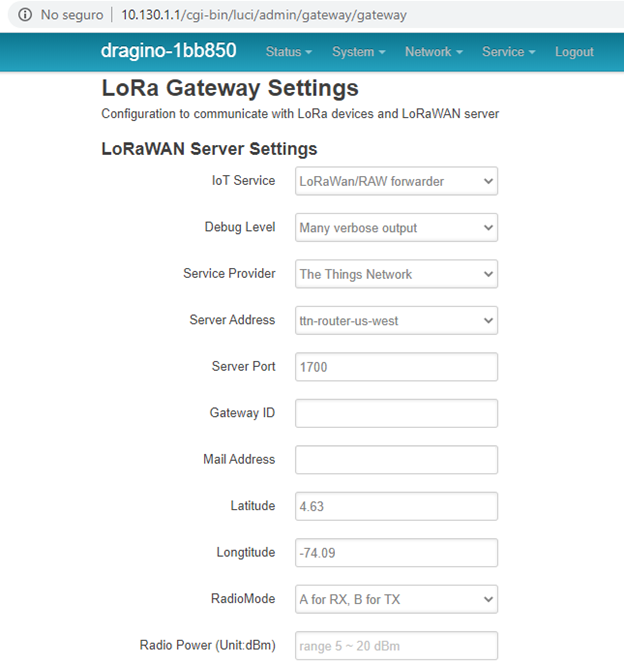
**Figura 5-1:** diagrama general componentes Gateway LG02 tomada de (*Dragino LG02*)

**Además de operar como Gateway dispositivo es configurable en diferentes modos de operación para ajustarse a las diferentes necesidades de conexión IoT funcionando como: Repetidor LoRa, modo de receptor MQTT, cliente TCP/IP o servidor TCP/IP, esta es una solución de bajo costo y por esto su capacidad de servicio está limitada a la conexión de hasta 300 nodos finales sin embargo su costo es aproximadamente la cuarta parte de lo que cuesta un Gateway con un módulo concentrador potente como el SX1301.**

### **Configuración del Gateway**

Para la configuración del dispositivo se conectó vía Ethernet y se siguió el manual del fabricante, una vez el dispositivo estuvo conectado a la red fue posible acceder a su interfaz de configuración ingresando por medio de un navegador web a la siguiente ruta: http://10.130.1.1/cgi-bin/luci/admin.

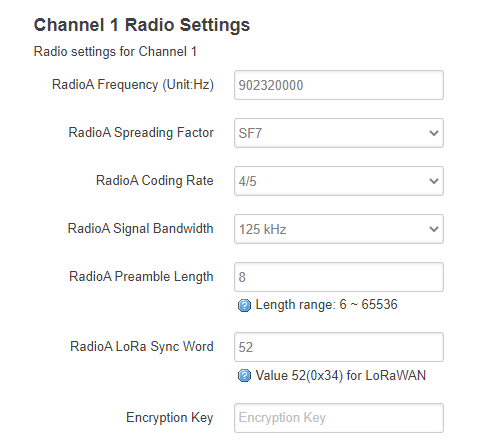
Esta ruta muestra una interfaz de autenticación a la que solo se puede acceder usando usuario y contraseña suministrados por el fabricante los cuales están en la etiqueta del producto. Una vez allí se selecciona el modo de operación, en este caso se usara como Gateway LoRa.



**Figura 5-1:** Interfaz de configuración del LG02.

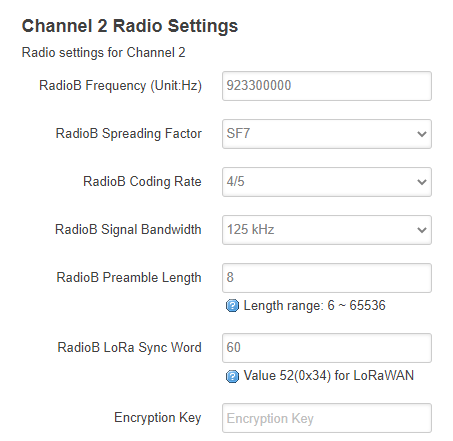
**Los parámetros que se observan en la imagen anterior incluyen el modo en el que se usara el Gateway, en nuestro caso una repetidor de datos crudos, es decir sin procesar, el nivel de información que queda en el registro del Gateway, el servidor IoT que recibirá los mensajes, la ubicación física del servidor, el puerto habilitado para la recepción de comunicaciones, las coordenadas geográficas, los canales de radio, se usara el canal A para recibir y el canal B para transmitir, la potencia de la señal de radio en Dbs y el correo electrónico de notificación**

**En la siguiente sección se deben configurar los parámetros con los que se están modulando las señales desde los nodos, es importante tener en cuenta que los parámetros con los que emiten los nodos deben coincidir exactamente con los que recibe el Gateway en nuestro caso los parámetros de recepción quedaran de la siguiente manera.**



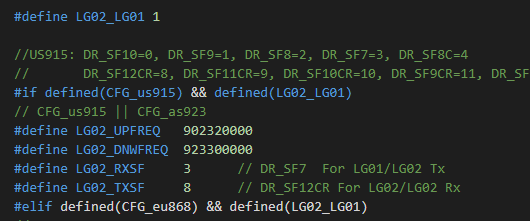
**Figura 5-1:** Configuración del canal de recepción LG02.

**Aquí se pueden distinguir: la frecuencia de modulación en Hercios, el factor de propagación que varía entre 7 y 12, el radio de codificación que será de 4/5 y el ancho de banda que puede ser 125 kHz, 250 kHz o 500 kHz. De manera similar se deben configurar los datos de transmisión, considerando nuevamente que estos parámetros serán los que se establezcan en los nodos como datos de recepción para que efectivamente reciban las señales descendentes.**



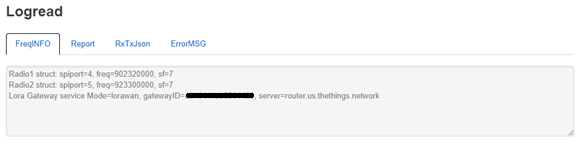
**Figura 5-1:** Configuración del canal de transmision LG02.

Para que los cambios tengan efecto se debe reiniciar el Gateway. En los nodos esta información se puede definir en el archivo de configuración de radio que se encuentra en la siguiente ruta src\lmic\config.h, como se puede ver las frecuencias y SF son inversas a las que se definieron en el Gateway.



**Figura 5-1:** Configuración de los parámetros de transmisión y recepción en la librería.

**Para validar los parámetros con los que opera el Gateway podemos simplemente revisar el registro que se en la pestaña de información de frecuencias, podemos observar cada uno de los canales, el modo de operación, el identificador del dispositivo, el cual dejara una marca en los paquetes que transiten a través de el y el servidor al cual se enviaran los mensajes.**

****

**Figura 5-1:** Registro de parametros de operación del LG02.

**Algo que nos interesa revisar son los mensajes que recibe el Gateway, para esta prueba demás de tener el Gateway configurado y conectado a la red, activamos uno de los nodos que se configuraron en el capítulo 2 suministrándole corriente, luego de un par de minutos en la pestaña de trasmisión se empezaron al obtener los mensajes que percibe el Gateway.**



**Figura 5-1:** Registro de mensajes retransmitidos por LG02.

Si estos mensajes corresponden con los cuales emitió el nodo que activamos podríamos afirmar que la prueba es exitosa, sin embargo dado que para efectos de seguridad la información del mensaje se encuentra codificada es complicado identificar si estos mensajes que se muestran en el registro pertenecen al nodo que fue activado o pertenecen a un nodo desconocido en la misma zona de cobertura, lo cual es poco probable pues para haber sido recibido, debe estar configurado con los mismos parámetros que nosotros usamos en el Gateway, de cualquier manera esto podemos validarlo una vez que decodifiquemos la información del mensaje una vez llegue al servidor.

## **Servidor IoT**

Uno de los pilares de IoT es la capacidad de identificar objetos del mundo real y dándoles una identidad en internet a través de la virtualización para poder monitorearlos, un servidor IoT es el lugar donde se programan las reglas para que los dispositivos puedan interactuar o mejor cooperar para cumplir un objetivo común, sin necesidad de la intervención humana. Por otra parte es aquí donde se agrupa la información de los dispositivos físicos y se analiza en conjunto y encontrar patrones o tendencias que permitan modelar e implementar soluciones más prácticas y eficientes.

Para entender de forma concreta que es un servidor IoT, se puede pensar un servicio publicado en internet que ofrece funcionalidades a los desarrolladores de sistemas de IoT para comunicar los dispositivos físicos y las aplicaciones; algunos de estos servicios incluyen el registro de dispositivos sean nodos finales o Gateways, la creación de aplicaciones desde las cuales se puede hacer seguimiento a los mensajes enviados por los dispositivos. Uno de los aspectos importantes en una aplicación IoT es la seguridad lo cual se consigue por medio de la autenticación de dispositivos y la verificación de integridad de los mensajes, existen servidores que permiten la decodificación de mensajes y algunos servicios para el post procesamiento de datos, como esquemas de visualización de información, integración con servidores de mapas o servicio de bases de datos en la nube.

Algunos de los servidores IoT más populares en la nube son Ubidots, ThingWorx, WSO2 IoT, ThingsBoard, AWS IoT Analitics y ThingSpeak, sin embargo dependiendo de la tecnología de trasmisión que se haya elegido en la capa física hay algunos que resultan integrarse mejor a nuestras aplicaciones en el caso de redes que usen el protocolo LoRaWAN unos de los servidores que mejor implementan los componentes de este protocolo es The Things Network TTN y es por esta razón que se ha elegido para el sistema.

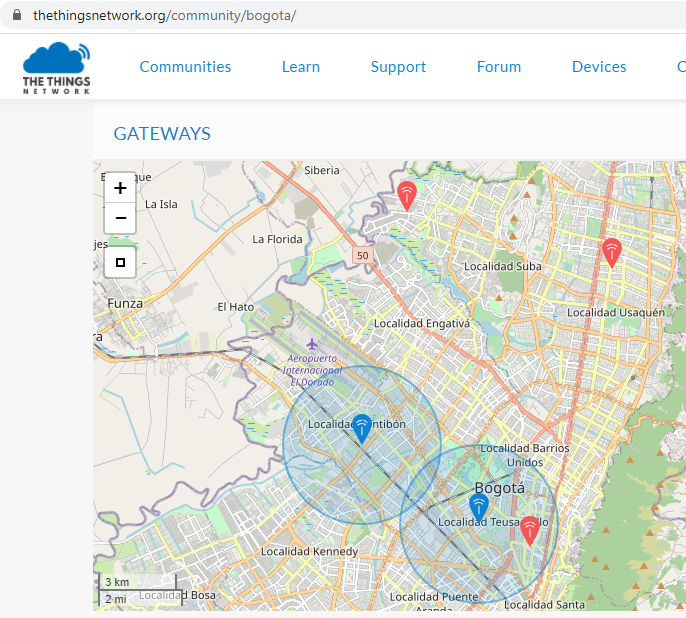
### The Things Network TTN



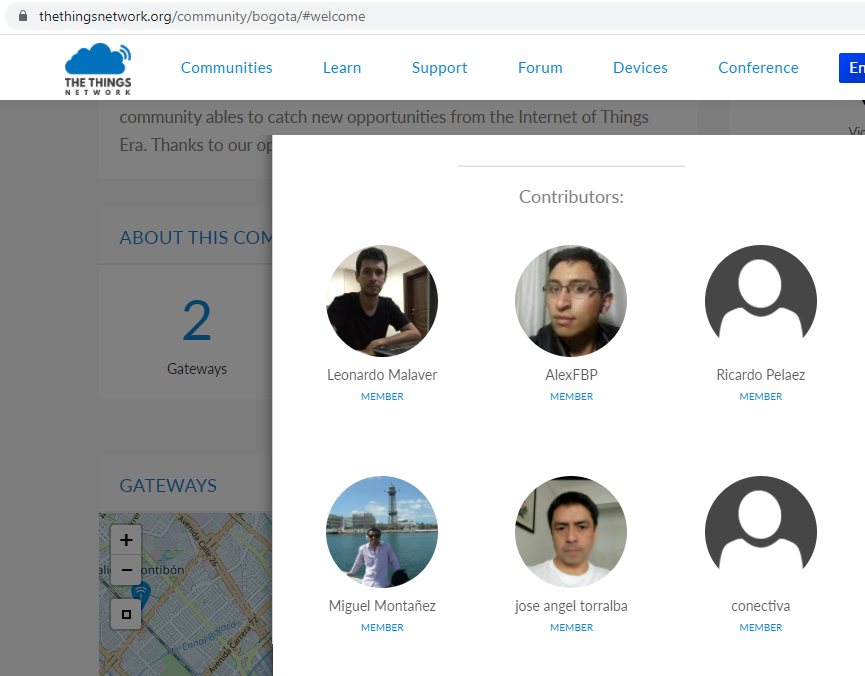
**Figura 5-1:** Logo comunidad TTN en Bogotá.

Suministra un servidor de red LoRaWAN que usando componentes minimalistas enfocados en los dispositivos, el servidor se encarga de la lógica de conexión lo cual es conocido como sistema de subyacente o backend, algunas de las funciones que proporciona TTN están relacionadas a los Gateways son el agenda miento, gestión de los dispositivos obedeciendo a los ciclos de labor y en enrutamiento de mensajes descendentes a través de los Gateway que ofrezcan mejor señal al dispositivo final. Hay funciones relacionadas con los dispositivos como la administración del estado de los dispositivos en la red, ofrece funcionalidades relacionadas con las aplicaciones como los son los Brokers y los manejadores que se usan para procesar el tráfico en la red indicando que información especifica requiere ser en rutada a cada aplicación haciendo uso de protocolos como AMQP o MQTT.

TTN es además una comunidad en la que los usuarios pueden cooperar para crear aplicaciones, compartir dispositivos y encontrar la distribución de Gateways que permita maximizar la cobertura en la cuidad, dándole mayor soporte a las aplicaciones, esto requiere ofrecer servicios de interoperabilidad en los que las aplicaciones se descentralicen, un servicio de descubrimiento para anexar a la red los nuevos nodos que se activen en el área de cobertura, ofrecer servicios que faciliten el enrutamiento de los mensajes para garantizar que son recibidos en el sitio que se requieren.



**Figura 5-1:** localización de Gateways registrados en Bogotá.



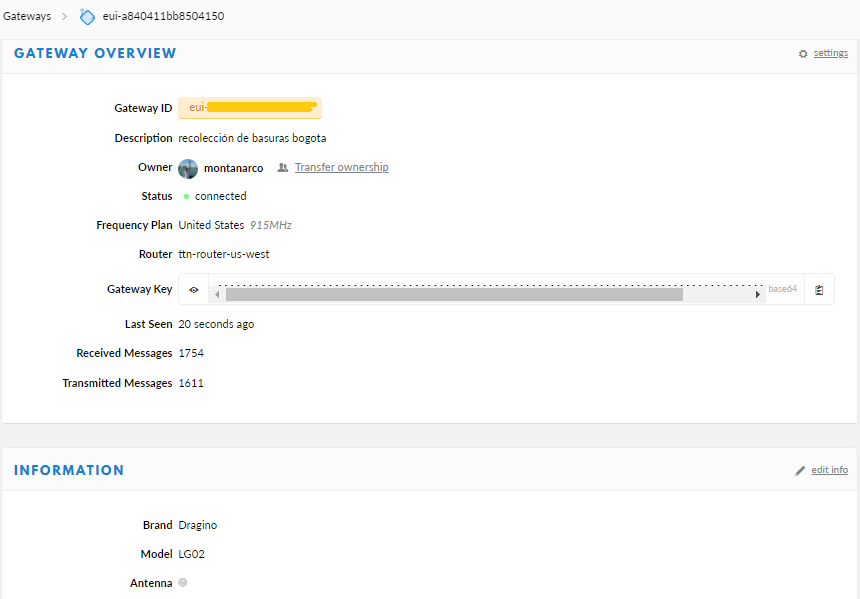
Para poder integrar los dispositivos en TTN se requieren algunas configuraciones preliminares.

### **Registro de dispositivos**

Antes de usar the things network se debe crear una cuenta de usuario esto le permite tener control sobre y administrar los dispositivos que se creen desde su cuenta de usuario y le garantiza que solo la persona con los datos de acceso a la cuenta podrá cambiar la información y la configuración de dichos dispositivos.

**Registro del Gateway:**

TTN dispone de existe una opción para registrar el Gateway en su consola de administración, para esta operacion se requieren algunos datos como el código de identificación, el plan de frecuencias en el que opera, la ubicación, el tipo de antena que usa y una descripción, luego de registrado se puede observar desde la consola de la siguiente manera.

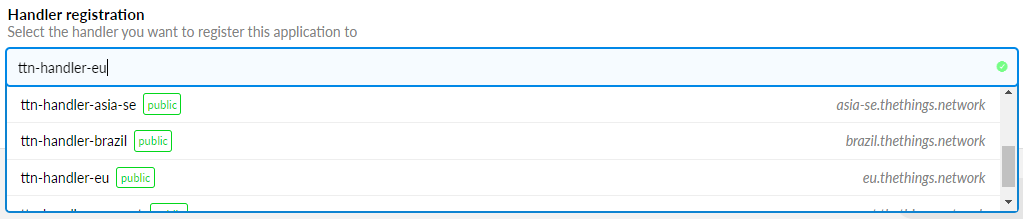


**Figura 5-1:** registro exitoso del Gateway en la consola de TTN.

Una vez el servidor recibe una petición con el código de identificación con el que el Gateway ha sido registrado, lo identifica y asocia y actualiza su estado, esto se puede observar en la interfaz en la figura XXXXX se puede observar que el Gateway esta conectado y la última vez que ha el servidor a recibió señal del Gateway.

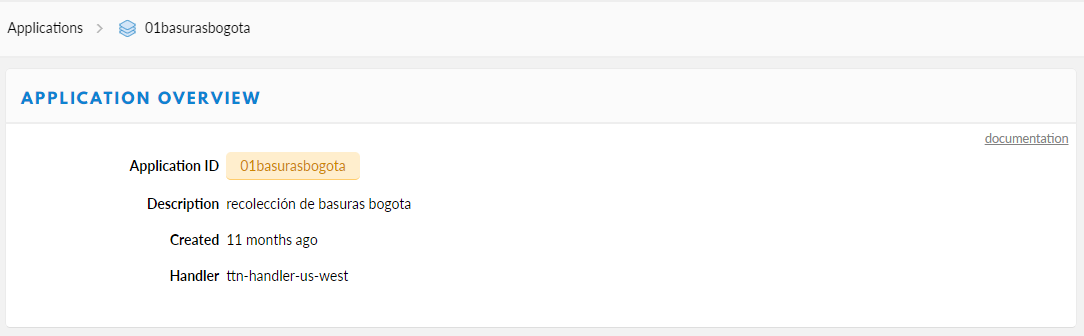
**Creación de la aplicación:**

Desde la consola de administración en la sección de aplicaciones también se creó una aplicación en la cual se recogerán los datos de los sensores que se reciban, para ello se requiere asignar un nombre a la aplicación, el servidor genera un código de aplicación, una descripción y la selección de un manejador de tráfico, del listado que se encuentra disponible, se recomienda que el manejador sea el mismo que el servidor asigno durante el registros del Gateway el cual está muy relacionado con el plan de frecuencias seleccionado.



**Figura 5-1:** selección de un manejador de tráfico para la aplicación.

Cuando la aplicación queda registrada se podrá ver su información desde la consola

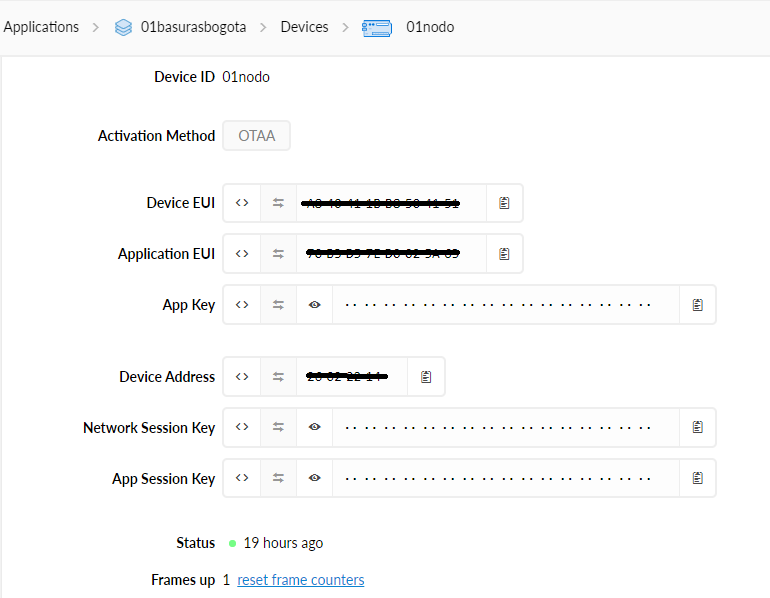


**Figura 5-1:** información de la aplicación que se creó.

**Registro de los nodos:**

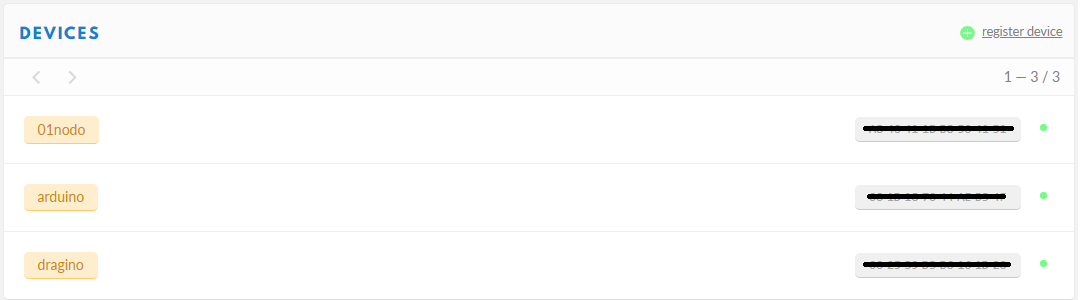
El registro de nodos se hace en la sección dispositivos de la aplicación que se creó anteriormente, con esto se consigue que los dispositivos queden asociados a dicha aplicación. Dicho de otra manera que los mensajes que se reciben desde los dispositivos que están asociados a la aplicación sean procesados con las mismas reglas que se programan a nivel de aplicación, para registrar un dispositivo es suficiente con darle un código de identificación, se sugiere que este código tenga un significado relevante que nos dé cierto grado de información acerca de que función cumple en la red, dado que aunque esta información también puede ser enviada en la trama del mensaje aumentara el tamaño del paquete que es algo que se desea evitar considerando las limitaciones de ancho de banda.

Una vez el dispositivo ha sido creado se requiere configurar un mecanismo de activación que como vimos en los conceptos generales podía ser OTAA o ABP, para este caso se eligió OTAA por lo TTN genera las claves necesarias para poder procesar esta autenticación.



**Figura 5-1:** Registro del 01nodo en la aplicación TTN.

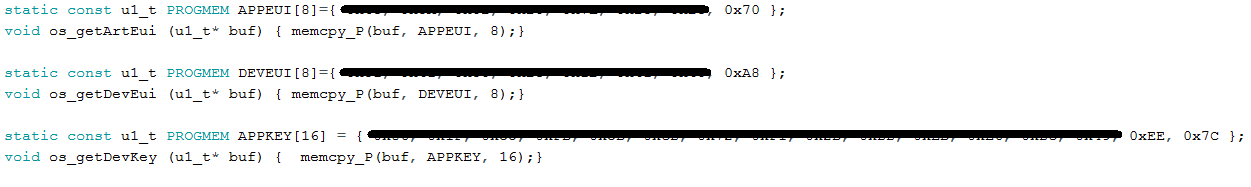
De forma análoga se procedió con el registro de los otros 2 dispositivos que se ensamblaron en el capítulo 2 con lo cual quedaron registrados los 3 dispositivos, el punto verde al final de cada línea indica que ya se ha establecido una conexión entre el servidor y el nodo.



**Figura 5-1:** lista de dispositivos asociados a la aplicación.

**Autenticación OTAA:**

Dado que ya disponemos de los códigos requeridos para gestionar la activación; Device EUI, App EUI y App Key podemos hacer uso de dichos códigos para configurar el nodo y así poder establecer un canal de comunicación seguro entre el nodo y el servidor.

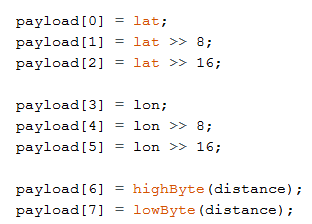


**Figura 5-1:** configuración de los códigos de autenticación OTAA en los nodos.

Estos códigos serán parte de la trama cuando se realice la petición de conexión, el servidor validara la correspondencia de estos datos y en caso de haber coincidencia retornara por medio de un mensaje descendente la dirección del dispositivo, que será usara por el Gateway para enrutar el mensaje de retorno, además de un ID de red y un AppNounce, que el nodo usara para generar el AppSKey y el NetSKey con los que empezara a enviar los mensajes al servidor. La demostración de la comunicación se hará mas adelante en este mismo capitulo

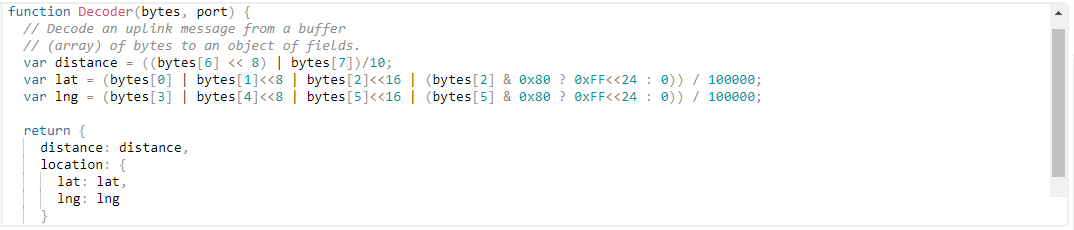
****Función de decodificación:****

**Los datos emitidos por el nodo final se organizan en un arreglo de 8 bytes de los cuales se usan 3 para la coordenada de latitud, 3 para la coordenada de longitud y 2 para la distancia que obtiene el sensor de proximidad, es necesario hacer un corrimiento de bits para que los datos muy grandes se ajusten al tamaño del arreglo esta codificación se realiza a nivel de nodo de la siguiente manera:**



**Figura 5-1:** organización del arreglo con carga útil en los Nodos finales.

Cuando esta información llega al servidor los bytes están codificado en Hexadecimal y no son legibles de manera que para que sean útiles deben ser decodificadas, esta es una de las características que nos suministrar TTN, nos permite escribir funciones JavaScript que sean comunes para decodificar los mensajes cuyo origen sean los dispositivos que están asociados a la aplicación esta funciones se llaman payload formats. Para poder decodificar el arreglo de bytes que estamos construyendo en los nodos la función de decodificación queda de la siguiente manera.



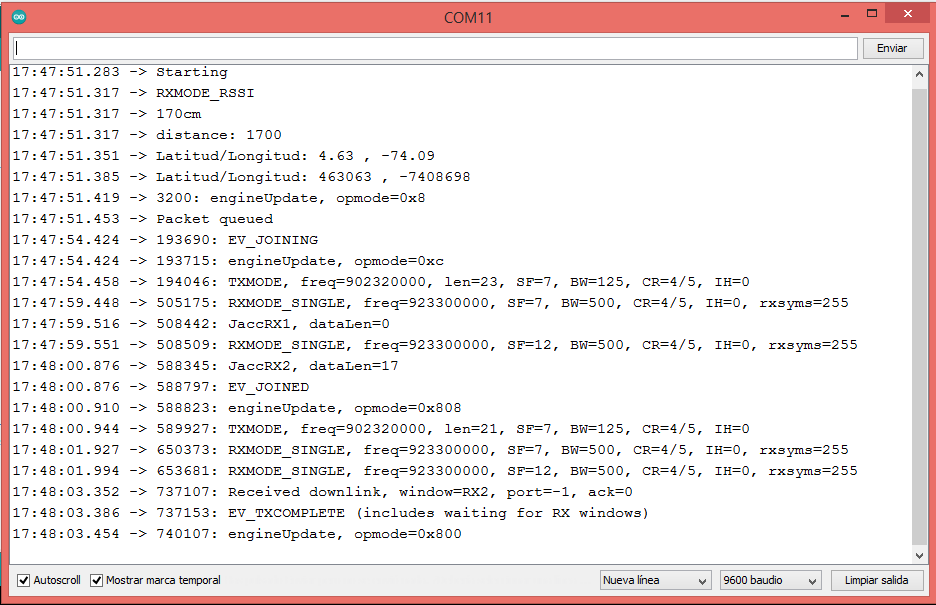
**Figura 5-1:** función para decodificación de la carga útil recibida en el servidor.

De forma similar TTN ofrece continuar procesando los datos luego de que estos son decodificados se tiene un conversor con la cual podemos hacer transformaciones de los datos, un validador para determinar si los datos recibidos tienen un nivel de integridad, cumple con los formatos y están dentro de los valores o rangos permitidos y finalmente dispone de una sección de codificación para los datos que se retornaran a los nodos.

## Prueba de recepción de datos:

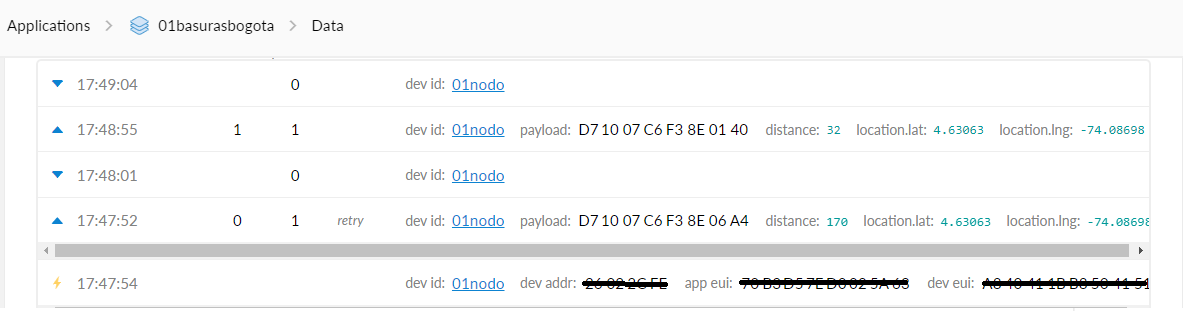
Luego de haber configurado los dispositivos en el servidor, tanto el Gateway como los nodos finales, luego de configurar los datos de activación por mecanismo OTAA y de haber configurado las funciones de decodificación, veremos el flujo de datos desde los 3 nodos hasta el servidor.

### Pruebas con TTGO (01nodo):



**Figura 5-1:** inspección del registro del 01nodo usando el monitor serial.

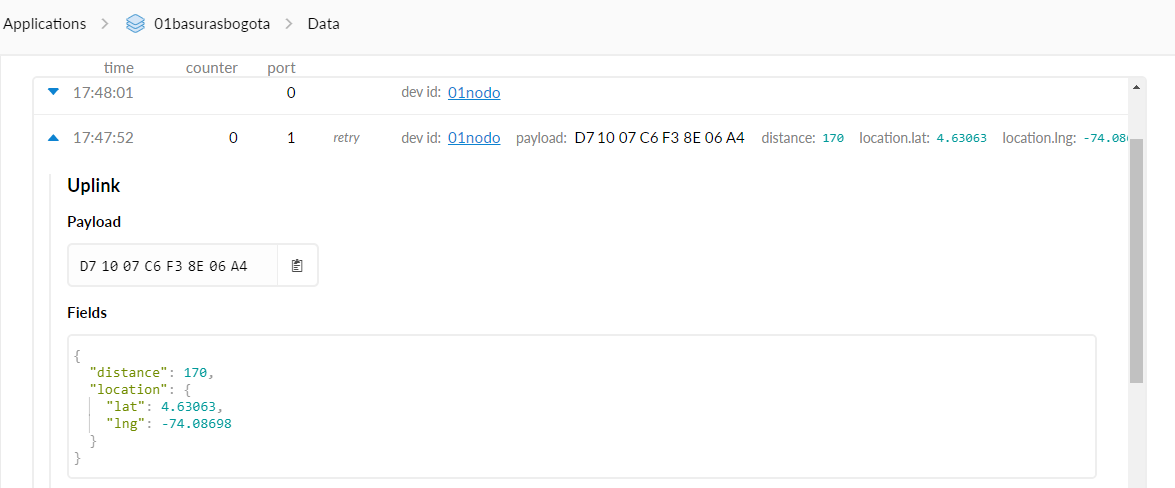
Aquí podemos observar que luego de empaquetar el los datos obtenidos por los sensores de ubicación y proximidad en paquete queda en espera para ser enviado, se puede ver el mensaje EV\_JOINING que indica que el nodo inicio una negociación con el servidor para establecer conexión, luego 7 líneas más abajo se ve el mensaje EV\_JOINED, lo cual indica que ya se estableció conexión con el servidor y 6 líneas más abajo se puede ver el mensaje EV\_TXCOMPLETE, lo cual indica que el mensaje ha sido entregado y confirmado por el servidor. Si revisamos el flujo de datos de la aplicación en la interfaz del servidor podemos observar lo siguiente.



**Figura 5-1:** datos emitidos por el 01nodo en la aplicación TTN.

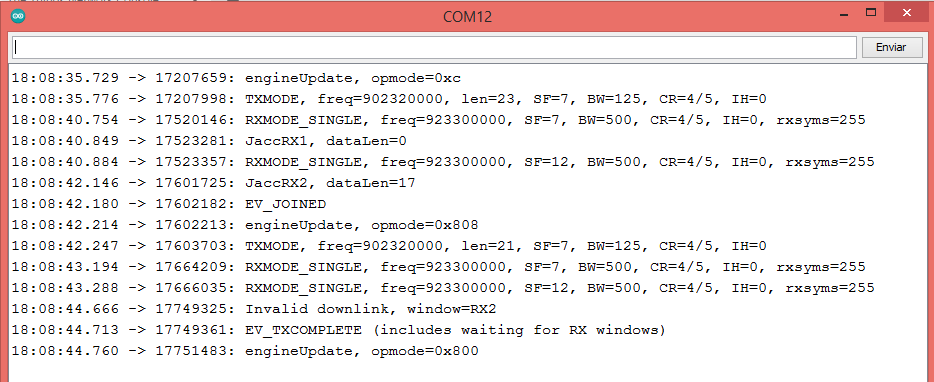
Leyendo la secuencia de abajo hacia arriba podemos observar que el servidor detecto que el dispositivo estaba encendido y realizo una solicitud de conexión enviando **dev addr**, **app eui** y **dev eui**, luego de establecer conexión se recibe un mensaje con carga útil codificada como D71007C6F38E06AE, y un mensaje de respuesta en el que no se está enviando ninguna información descendente, el mensaje se repite a las 17:21:01.

Inspeccionando el mensaje podemos observar que la función de decodificación está operando correctamente pues transforma los valores recibidos en hexadecimal y retorna los valores de distancia, latitud y longitud.



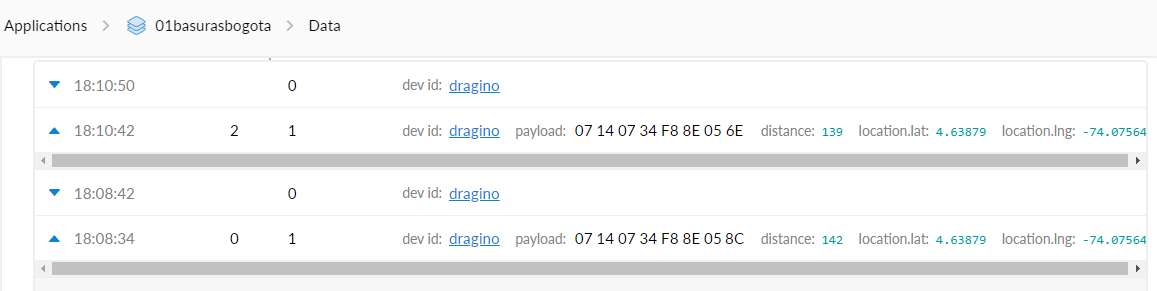
**Figura 5-1:** salida de la función de decodificación en tiempo de ejecución.

### Pruebas con Shield (Dragino):



**Figura 5-1:** inspección del registro del Dragino usando el monitor serial.

En la imagen se puede observar que luego de la conexión con el servidor la trasmisión ha sido completada por el mensaje EV\_TXCOMPLETE.

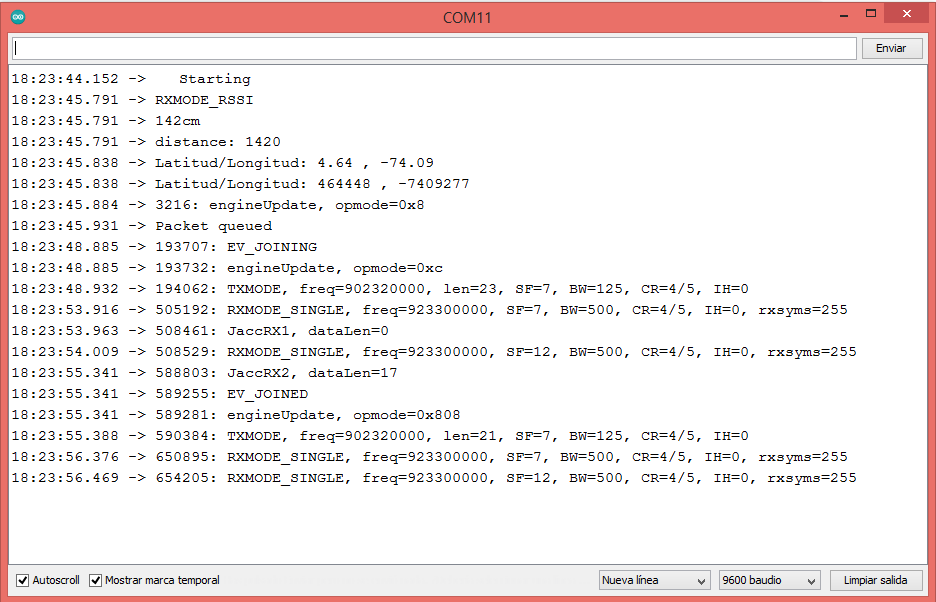


**Figura 5-1:** datos emitidos por el Dragino en la aplicación TTN.

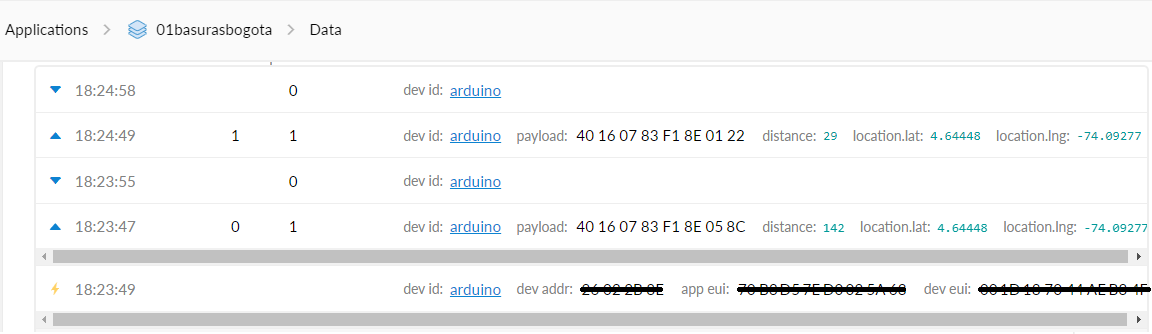
Y se puede ver como los datos son recibidos y decodificados en el servidor.

### Pruebas con Arduino y SX1276

La prueba se repitió con el tercer dispositivo y se obtuvieron los resultados esperados.



**Figura 5-1:** inspección del registro del Arduino usando el monitor serial.



**Figura 5-1:** datos emitidos por el Arduino en la aplicación TTN.

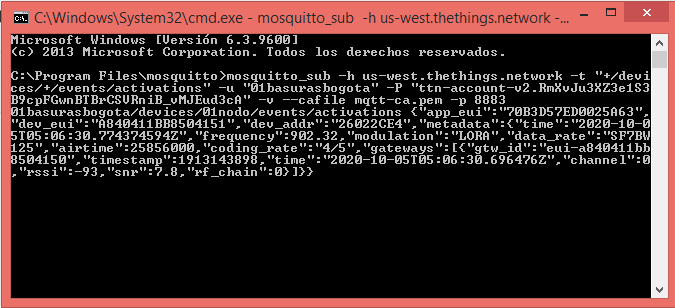
## **Bróker MQTT**

Tras recibir los datos al servidor IoT, se requiere de un mecanismo con el cual se comunique a las aplicaciones que hacen usan de los datos, TTN soporta 2 de los protocolos de la capa de comunicación MQTT y AMQP, sin embargo para todas la aplicaciones TTN cofigura por defecto un Broker MQTT usando una implementación de Eclipse Mosquito, teniendo en cuenta que esto simplifica mucho el proceso de extracion de datos pues ya se cuenta con un bróker, hace falta solamente la implementación de un cliente para poder extraer los datos que se reciben en la aplicación. Para configurar cliente bróker se realizaron los siguientes pasos:

1 se descargó e instalo mosquito de la página <https://mosquitto.org/download/>

2. se descargó el certificado de para añadir seguridad en la capa de transporte TLS <https://console.thethingsnetwork.org/mqtt-ca.pem> que ofrece TTN

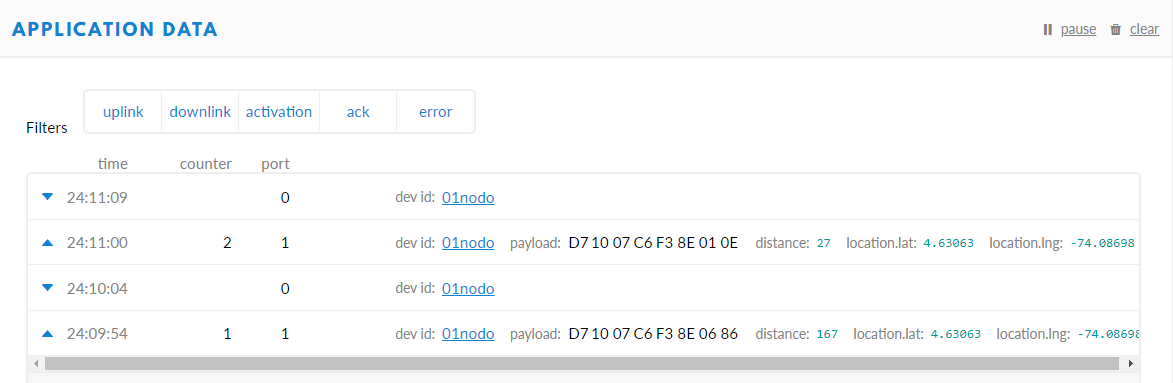
3. se hizo una prueba para verificar que dispositivos estaban siendo activados en la aplicación para lo cual se usó la siguiente sentencia



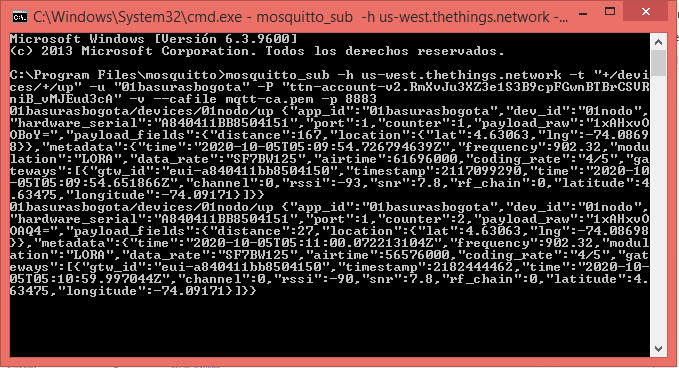
**Figura 5-1:** datos recibidos tema de dispositivos activados en la app 01basurasbogota.

4. se hizo la suscripción al canal de los mensajes que ingresan a la aplicación en modo ascendente

5 se compara la información de los mensajes que llegaron al servidor IoT y los que se encuentran en el Broker y se valida la correspondencia



**Figura 5-1:** mensajes recibidos en el servidor para la app 01basurasbogota.



**Figura 5-1:** datos recibidos tema de mensajes ascendentes en la app 01basurasbogota.

## Prueba Cliente MQTT

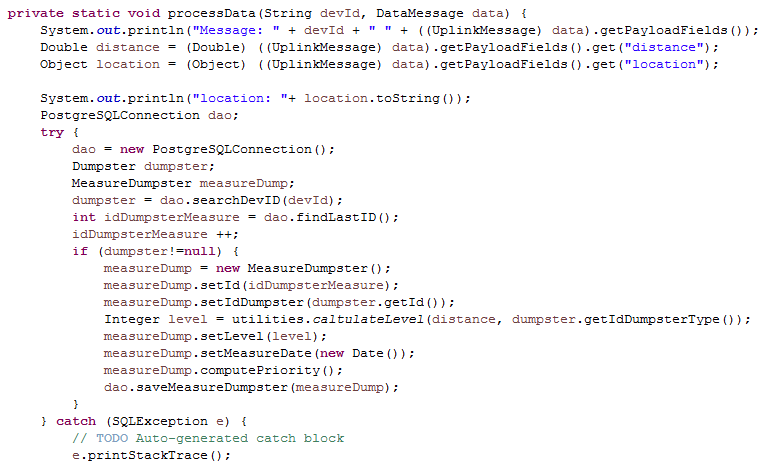
Este ejercicio fue suficiente para determinar que la informacion se podía extraer del servidor por medio de un cliente MQTT, sin embargo se requiere una implementación más flexible que nos permita hacer un pos procesamiento de los datos.

La comunidad de TTN también creo un paquete de desarrollo para embeber el cliente en una aplicación Java el código fuente es de libre acceso y se encuentra publicado en el siguiente repositorio <https://github.com/TheThingsNetwork/java-app-sdk> este repositorio contiene una implementación del protocolo MQTT y una para el protocolo AMQP.

Lo que se hizo es que de una copia del kit de desarrollo ofrecido por TTN se generó un cliente que se suscribe al canal de mensajes ascendentes, que contienen los datos de nuestra aplicación, en la función de recepción de mensajes se invoca una función pos procesamiento que se encarga de extraer la hora de la medición el dispositivo, y los valores que fueron retornados por la función de decodificación, usa estos valores y el tipo de contenedor al que esta enlazado el sensor para estimar el nivel de llenado y persiste los datos en la base de datos de nuestra aplicación.

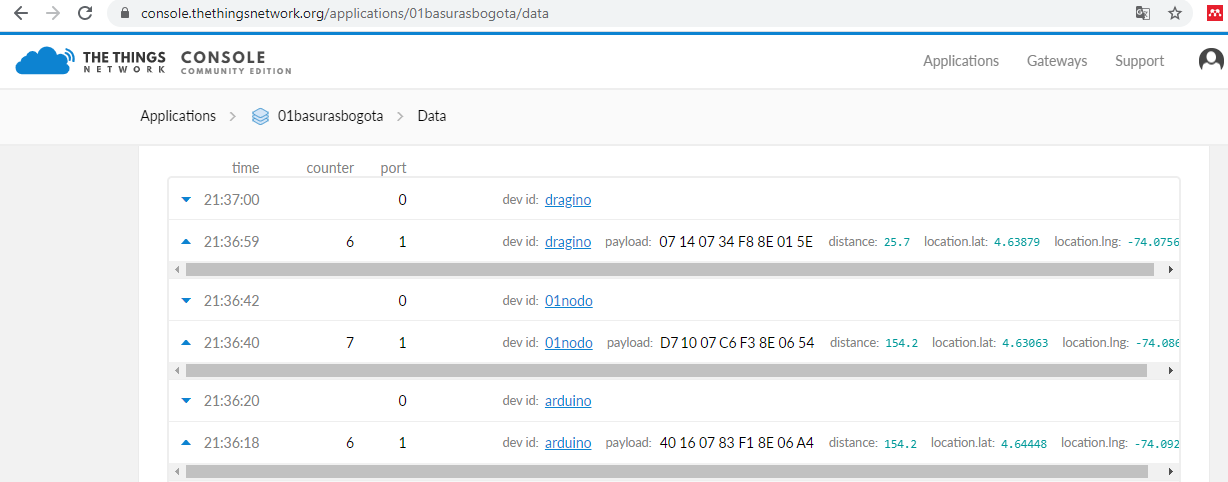


**Figura 5-1:** función SDK TTN que se suscribe a los mensajes ascendentes de la aplicación mediante mosquitto.



**Figura 5-1:** función pos procesamiento de datos.

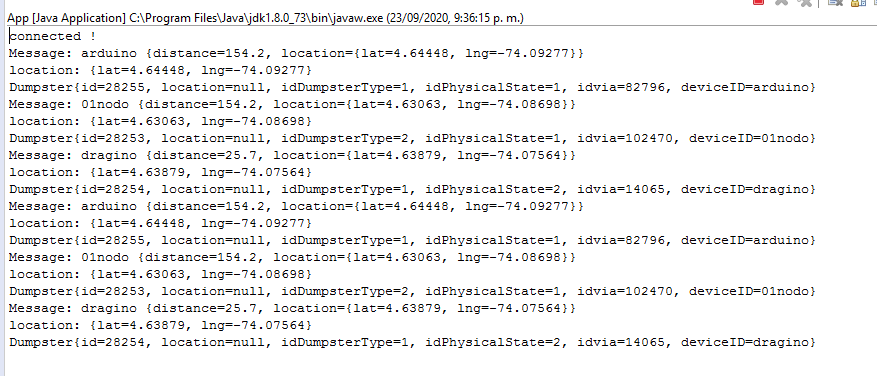
Esta aplicación se compilo y se compilo como una aplicación java estática. Luego de conectar los nodos finales y tener el Gateway escuchando se obtuvieron los siguientes resultados



**Figura 5-1:** registro de datos de la aplicación TTN.

Los datos que se estaban obteniendo en el servidor IoT estaban siendo recuperados con apenas un par de segundos de retraso por la implementación del cliente MQTT que se creó en java, los mensajes ascendentes que transmite el bróker son muy complejos, pero para simplificar la lectura fueron reducidos a un mensaje que contiene la siguiente estructura:

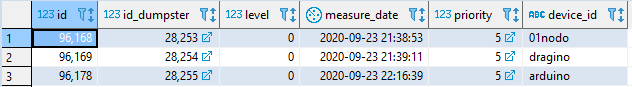
Identificador del dispositivo { distance = medición Distancia, location = {lat= coordenada Latitud, lon= coordenada Longitud}}.



**Figura 5-1:** registro de datos del cliente MQTT en java.

Estos datos serán un recurso valioso que se usara en las aplicaciones que se implementaran en los próximos capítulos, por esta razón fue necesarios almacenar estos datos en una entorno desde el cual fuera fácil de acceder por nuestras aplicaciones y decidimos que fuese una base de datos por la seguridad que ofrece y la capacidad que tiene para integrarse con otros sistemas.

Los registros quedaron almacenados en la base de datos con la siguiente estructura



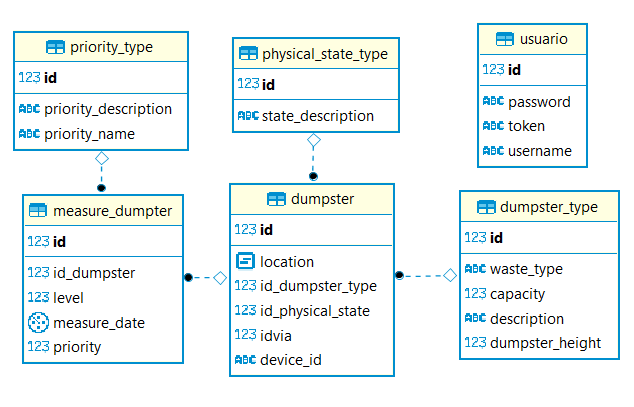
**Figura 5-1:** consulta de los últimos registros recibidos en la tabla de mediciones.

## **Base de Datos**

La el motor de base de datos que se eligió es postres SQL, que es una base de datos relacional con un lenguaje de consulta estructurado, pero que opera con una licencia de código abierto su licencia establece lo siguiente:

“permiso para usar, copiar, modificar y distribuir este software y su documentación para cualquier propósito, sin pago de cuotas” (postgreSQL Licence)

Pero además de la licencia, una de las características que nos interesan de postgres es el complemento para manejo de datos espaciales postGIS, dado que en nuestro sistema queremos poder referencias ubicaciones de las calles por las que los camiones recolectores deben hacer sus recorridos, dependiendo de la ubicación de los contenedores, lo cual será tema del capítulo 5. Por lo pronto se desea mostrar la estructura de datos que se está usando para almacenar los datos de los sensores.



**Figura 5-1:** modelo relacional del segmento de BD relacionado con contenedores y mediciones, generado usando DBeaver.

La imagen se muestra 6 tablas de las cuales se hace una breve descripción:

**Tipo de contenedor:** (dumspert\_type) en la cual se caracterizan los tipos de contenedores que dispone la ciudad y que varian según su capacidad y el tipo de residuo al que están destinados.

**Estado físico:** (physical state) que caracteriza cual es la condición del contenedor, indicando si se encuentra en buen estado, si requiere mantenimiento o esta dañado y debe ser reemplazado.

**Contenedor:** (dumpster) la cual es el registro correspondiente a la abstracción un contenedor real, en el cual se registra su tipo, su estado físico, su ubicación, la via sobre la cual está ubicado y el identificador del nodo de medición que tiene asociado y que se registró en el servidor IoT.

**Tipo de prioridad:** (priority\_type) en esta tabla se caracterizan las urgencias de atender un contenedor con base a su nivel de llenado.

**Medición de contenedor:** (measure dumpster) en esta tabla se persisten las mediciones registradas para cada contenedor, asociando el contenedor al que le pertenece la medición, el instante en el que se reportó dicha medición, el nivel de llenado reportado y la prioridad asociada a ese nivel.

# Capítulo 4: Simulación de Datos

Uno de los problemas que se enfrentaron durante la formulación de este sistema, fue que no se dispone de un registro de la producción de basuras en Bogotá o si existen no son de uso público y son mantenidos en reserva por las entidades que administran el sistema de recolección actual de basuras UAESP[[8]](#footnote-8), pero adicionalmente a que para calcular una ruta se requiere un número significativo de contenedores y sus respectivas mediciones sin embargo como se vio en el capítulo 2 actualmente solo se dispone de 3 nodos físicos para lo cual no es suficiente para evaluar como operaria el resto del sistema si ya se dispusiese de una cobertura de nodos en un área amplia.

La simulación es un instrumento que permite recrear de forma aproximada en comportamiento de un fenómeno o proceso del mundo real y crear un modelo para conducir experimentos comúnmente numéricos que nos permitan entender este comportamiento, dada una serie de condiciones y generar estrategias para obtener mejores resultados del procedimiento o fenómeno que se estudia. (Cabrera, 2010)

La simulación que se propone consisten en usar las ubicaciones de los contenedores reales y asumir que cada uno de ellos tiene instalado un nodo de medición y que se encuentran en el rango de cobertura de un Gateway, conformando así una RIS que reporta una vez al día el nivel de llenado de dichos contenedores, con el fin de usar esta información poder planificar la operación de recolección. La región de estudio ha sido limitada a una sola de las localidades de Bogotá, porque la manipulación de información geográfica es costosa en términos de procesamiento, sin embargo esta prueba de concepto en una zona limitada nos permitió definir la validez del sistema en una región más amplio dado que la estrategia es replicable de forma sistemática a las demás localidades de la cuidad, siempre y cuando disponga de la información de ubicación de los contenedores.

Se eligió la Localidad de Engativá pues junto con la localidad de barrios unidos acumulan el 28,68% de todos los contenedores de acuerdo con información del Bogotá Limpia.

“*Se tienen prevista la ubicación de 2.868 contenedores en dos cuadrantes. El primero irá desde la avenida Boyacá hasta la carrera 96, entre la calle 80 y el humedal Juan Amarillo. El segundo será desde la avenida Boyacá hasta el río Bogotá, entre la calle 80 y la calle 72*. *El operador Área Limpia anunció que instalará 2.726 en Suba*.”

(El Espectador)

## **Conjunto de datos**

Debido a que no fue posible encontrar información referentes a la recolección de residuo sólido urbano RSU en Bogotá dada la carencia de registros de acceso público se optó por analizar un conjunto de datos de una ciudad que tuviese algunas características similares a las de Bogotá como: su densidad poblacional, la distribución de calles y esquemas de recolección de basuras, lo cual resulto ser complejo dado que cada cuidad es única y que exista una coincidencia para los aspectos mencionados entre 2 ciudades es improbable.

Sin embargo en Kagle que es uno de los repositorios de datos más grandes que existe en internet, se encontró un conjunto de datos generado por la gobernación de Austin Texas, USA que fue publicado para uso general, que contiene información de los lugares de descarga de RSU, con identificaciones, tipos y pesos de las cargas, las fechas y descripción de las rutas en las que se recolectaron estos residuos entre 2008 y 2016. La información del conjunto de datos indica donde se generan el RSU, quien se hace cargo de él y a donde va.

De acuerdo con el repositorio de datos esta información fue recolectada con el ánimo de responder a las siguientes preguntas:

* ¿Que tanto RSU está generando la cuidad de Austin Texas?
* ¿Cuáles son las rutas que mayor cantidad de RSU producen? y ¿quiénes hacen un mejor proceso de reciclaje?
* ¿existen cambios o comportamientos periódicos?
* Intentar predecir el las rutas de recolección a partir de los datos históricos.

Estas preguntas atrajeron bastante nuestra atención a que giran en torno a aspectos muy relacionados a los que se pretenden modelar en la simulación y a que tienen una motivación en común; tratar de predecir las rutas de recolección, por esta razón se eligió este conjunto de datos para ser analizado.

## **Limpieza de datos**

## **Análisis exploratorio**

## **Representación geográfica de las rutas**

## **Modelo de persistencia datos**

## **Generación de datos**

El análisis llevado a cabo computado en conjunto con la densidad de población de los barrios y la distribución de los contenedores nos permitió predecir el comportamiento de la generación de residuos

## **Representación geográfica de los datos generados**

## **Técnicas de agrupamiento**

# Capítulo 5: Generación de Rutas de Recolección

La clasificación del tipo de residuos sean estos desecho general o material reciclable, permite estimar el volumen del material que se encuentra en el contenedor, ya que comúnmente el desecho pesa más que el material reciclable, así con un cálculo aproximado del volumen que se agrupa en todos los contenedores se puede determinar cuál es la cantidad de camiones que se requieren para realizar una operación de recolección.

## **Modelo matematico**

# Capítulo 6: Interfaz visualización de Rutas

Se deben incluir tantos capítulos como se requieran; sin embargo, **se recomienda que la tesis o trabajo de investigación tenga un mínimo 3 capítulos y máximo de 6 capítulos (incluyendo las conclusiones).**

# Análisis y Resultados

Se deben incluir tantos capítulos como se requieran; sin embargo, **se recomienda que la tesis o trabajo de investigación tenga un mínimo 3 capítulos y máximo de 6 capítulos (incluyendo las conclusiones).**

# Conclusiones y recomendaciones

## Conclusiones

**Las conclusiones constituyen un capítulo independiente y presentan, en forma lógica, los resultados del trabajo. Las conclusiones deben ser la respuesta a los objetivos o propósitos planteados. Se deben titular con la palabra conclusiones en el mismo formato de los títulos de los capítulos anteriores (Títulos primer nivel), precedida por el numeral correspondiente (según la presente plantilla).**

**Las conclusiones deben contemplar las perspectivas de la investigación, las cuales son sugerencias, proyecciones o alternativas que se presentan para modificar, cambiar o incidir sobre una situación específica o una problemática encontrada. Pueden presentarse como un texto con características argumentativas, resultado de una reflexión acerca del trabajo de investigación.**

## Recomendaciones

Se presentan como una serie de aspectos que se podrían realizar en un futuro para emprender investigaciones similares o fortalecer la investigación realizada.

1. Anexo: Nombrar el anexo A de acuerdo con su contenido

**Los Anexos son documentos o elementos que complementan el cuerpo del trabajo y que se relacionan, directa o indirectamente, con la investigación, tales como acetatos, cd, normas, etc. Los anexos deben ir numerados con letras y usando el estilo “Título anexos”.**

1. Anexo: Nombrar el anexo B de acuerdo con su contenido

**A final del documento es opcional incluir índices o glosarios. Éstos son listas detalladas y especializadas de los términos, nombres, autores, temas, etc., que aparecen en el trabajo. Sirven para facilitar su localización en el texto. Los índices pueden ser alfabéticos, cronológicos, numéricos, analíticos, entre otros. Luego de cada palabra, término, etc., se pone coma y el número de la página donde aparece esta información.**

Bibliografía

**La bibliografía es la relación de las fuentes documentales consultadas por el investigador para sustentar sus trabajos. Su inclusión es obligatoria en todo trabajo de investigación. Cada referencia bibliográfica se inicia contra el margen izquierdo.**

**Bajo la Resolución 023 de 2015. Artículo 2. Parágrafo 1.**

**La plantilla no especifica la norma bibliográfica que se debe utilizar. Se brindará la libertad para aplicar la norma para el manejo de las referencias bibliográficas, de acuerdo con el estándar de cada área del conocimiento, siempre y cuando ésta se aplique con rigurosidad.**

**Se recomienda el uso de gestores bibliográficos como Mendeley, Zotero, etc. A continuación, se lista algunas instituciones que brindan parámetros para el manejo de las referencias bibliográficas:**

Ejemplo Referencias bibliográficas

|  |  |  |
| --- | --- | --- |
| ****Institución**** | ****Disciplina de aplicación**** | ****Vínculos y ejemplos**** |
| **AMA (Asociación Médica de los Estados Unidos)** | **Ambito de la salud (psicología, medicina)** | [HealthLinks.Washington.edu/hsl/StyleGuides/AMA.htm](http://healthlinks.washington.edu/hsl/styleguides/ama.htm) (manual de estilo de la AMA, que sirve de estándar para las disciplinas que se ocupan de medicina, salud y ciencias biológicas).  [Liunet.edu/Cwis/Cwp/Library/Workshop/CitAMA.htm](http://www.liunet.edu/cwis/cwp/library/workshop/citama.htm) (ejemplos). |
| **American Psychological Association (APA)** | **Ambito de la salud (psicología, medicina) y en general en todas las ciencias sociales.** | [APAStyle.org](http://www.apastyle.org/).  [Biblioteca.udg.es/Info\_General/Guies/Cites/Citar\_Llibres.asp](http://biblioteca.udg.es/info_general/Guies/Cites/citar_llibres.asp) (reglamento).  [Liunet.edu/Cwis/Cwp/Library/Workshop/Citapa.htm](http://www.liunet.edu/cwis/cwp/library/workshop/citapa.htm) (ejemplos). |
| Harvard System of Referencing Guide | **Todas las disciplinas** | Disponibles en: <http://libweb.anglia.ac.uk/referencing/harvard.htm> |
| JabRef y KBibTeX | **Todas las disciplinas** | Herramientas de LateX para la gestión de referencias bibliográficas. |

**(continúa)**

**Ejemplo Referencias bibliográficas (continuación)**

|  |  |  |
| --- | --- | --- |
| ****Institución**** | ****Disciplina de aplicación**** | ****Vínculos y ejemplos**** |
| **Manual** | **Todas las disciplinas** | Patrias, K.: National Library of Medicine recommended formats for bibliographic citation. Bethesda (Maryland, EE. UU.): National Library of Medicine (Reference Section), 1991. |
| **Modern Language Association (MLA)** | **Literatura, artes y humanidades.** | [MLA.org](http://www.mla.org/)  [Biblioteca.udg.es/Info\_General/Guies/Cites/MLA.asp](http://biblioteca.udg.es/info_general/Guies/Cites/MLA.asp) (reglamento).  [Liunet.edu/Cwis/Cwp/Library/Workshop/CitMLA.htm Ejemplos](http://www.liunet.edu/cwis/cwp/library/workshop/citmla.htm) |
| **National Library of Medicine (NLM) (Biblioteca Nacional de Medicina)** | **En el ámbito médico y, por extensión, en ciencias.** | [NLM.NIH.gov](http://www.nlm.nih.gov)  [NLM.NIH.gov/Pubs/Formats/RecommendedFormats.html](http://www.nlm.nih.gov/pubs/formats/recommendedformats.html) (formatos recomendados) |
| **Universidad de Chicago/Turabian** | **Periodismo, historia y humanidades.** | [ChicagoManualOfStyle.org](http://www.chicagomanualofstyle.org/)  [BedfordStMartins.com/Hacker/Resdoc/History/Footnotes.htm](http://www.bedfordstmartins.com/hacker/resdoc/history/footnotes.htm) (Reglamento I) o  [BedfordStMartins.com/Online/Cite7.html](http://www.bedfordstmartins.com/online/cite7.html) (Reglamento II).  [liunet.edu/cwis/cwp/library/workshop/citchi.htm](http://www.liunet.edu/cwis/cwp/library/workshop/citchi.htm) (ejemplos de la universidad de Chicago)  [liunet.edu/cwis/cwp/library/workshop/citchi.htm Liunet.edu/Cwis/Cwp/Library/Workshop/Citchi.htm](http://www.) (ejemplos de las reglas de Turabian) |
| **Vancouver** | **Todas las disciplinas** | [Fisterra.com/Recursos\_Web/Mbe/Vancouver.asp](http://www.fisterra.com/recursos_web/mbe/vancouver.asp) (estilo de Vancouver 2000). |

# **Bibliografía**

*Arduino Uno Rev3 | Arduino Official Store*. (2017). Arduino UNO 2017. <https://store.arduino.cc/arduino-uno-rev3>

Azzola, F. (2019, May 28). *CoAP Protocol: Step-by-Step Guide*. Dzone-Coap. <https://dzone.com/articles/coap-protocol-step-by-step-guide>

ksestupinan. (2018, October 28). *Bogotá está instalando 10.000 contenedores*. Bogota-Asi-Vamos. https://bogota.gov.co/asi-vamos/contenedores-para-residuos-en-bogota

Cope. S. (2020, 10 september). How MQTT Works -Beginners Guide, steves internet guide. <http://www.steves-internet-guide.com/mqtt-works/>

*Dragino: About Us*. (2015). <https://www.dragino.com/about/about.html>

*Dragino GPS.* *LoRaWAN GPS Tracker with 9-axis accelerometer-LGT92*. (2019, August 11). Dragino GPS. https://www.dragino.com/products/lora-lorawan-end-node/item/142-lgt-92.html

*Dragino Distance*. *LDDS75 LoRaWAN Distance Detection Sensor*. (2020, June 10). <https://www.dragino.com/products/lora-lorawan-end-node/item/161-ldds75.html>

*Dragino LG02*. *LG02 Dual Channels LoRa IoT Gateway*. (2019, October 31). <https://www.dragino.com/products/lora-lorawan-gateway/item/135-lg02.html>

Hoplin, I. (2019, August 6). *MQTT: un protocolo abierto de red y su importancia en el IoT*. Open-MQTT. https://www.hwlibre.com/mqtt/

1&1 IONOS España S.L.U. (2020, September 30). *Advanced Message Queuing Protocol (AMQP)*. IONOS Digital Guide. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/advanced-message-queuing-protocol-amqp/

Lilygo-Xinyuan. *Company Profile - Shenzhen Xin Yuan Electronic Technology Co., Ltd*. (2017, May 12).. <http://www.lilygo.cn/about.aspx?TypeId=1&FId=t1:1:1>

*MQTT - The Standard for IoT Messaging*. (2020, May 3). MQTT. https://mqtt.org/

Msv, J. (2019, 10 mayo). *10 DIY Development Boards for IoT Prototyping*. The New Stack. <https://thenewstack.io/10-diy-development-boards-iot-prototyping/>

Naylampmechatronics. *Tutorial Módulo GPS con Arduino*. (2016, June 25). <https://naylampmechatronics.com/blog/18_Tutorial-M%C3%B3dulo-GPS-con-Arduino.html>

Solera, E. (2018, 27 agosto). *Modulación LoRa: Long Range Modulation - Exploración de aplicaciones usando tecnología LoRa*. Medium. <https://medium.com/pruebas-de-laboratorio-de-la-modulaci%C3%B3n-lora/modulaci%C3%B3n-lora-4ad74cabd59e>

Raftery, H. (2018, 26 febrero). *LoRaWAN: OTAA or ABP?* NewieVentures. <https://www.newieventures.com.au/blogtext/2018/2/26/lorawan-otaa-or-abp>

VOLOV. Alexander, V. (2016, June 13). *Advanced Message Queuing Protocol (AMQP) | Alexander Volov’s homepage*. <http://alexvolov.com/2016/06/amqp/>

*What is Arduino?* (2018, 5 febrero). Arduino cc. <https://www.arduino.cc/en/Guide/Introduction>

El Espectador -Redacción Bogotá - bogota@elespectador.com. (2020, July 2). *Listos los contenedores, solo falta saber dónde ubicarlos*. https://www.elespectador.com/noticias/bogota/listos-los-contenedores-solo-falta-saber-donde-ubicarlos/

# **Anexos**

# https://bogota.gov.co/sites/default/files/u3881/infografiacontenedores.jpg

**Figura 13-1:** infografía funcionamiento de los nuevos contenedores en Bogotá,tomada de (ksestupinan, 2018)

1. Compañía estado unidense proveedora de semiconductores y equipo avanzado de infraestructura de comunicaciones y equipo industrial [↑](#footnote-ref-1)
2. LASER: light amplification by stimulated emission of radiation [↑](#footnote-ref-2)
3. Lenguaje de programación basado en java con una sintaxis simplificada y orientado a la programación de equipos electonicos. [↑](#footnote-ref-3)
4. In-Circuit Serial Programming [↑](#footnote-ref-4)
5. Especificación de la interfaz serial de comunicación síncrona usada para comunicaciones de corta distancia en sistemas embebidos [↑](#footnote-ref-5)
6. National Marine Electronics Asociation [↑](#footnote-ref-6)
7. Full dúplex es una especificación que permite comunicación bidireccional gracias a que los dispositivos pueden emitir y recibir señal simultáneamente y no deben interrumpir la transmisión usando el canal para emitir. [↑](#footnote-ref-7)
8. Unidades administradoras especiales de servicios públicos. [↑](#footnote-ref-8)