

Diseño de un Nodo LoRa-GPS para Localización de Bicicletas Implementado en una  
Red LoRaWAN

Jairo Andrés Rodríguez Sánchez

Jhoiner Smith Rojas González

UNIVERSIDAD SANTO TOMÁS  
FACULTAD DE INGENIERÍA ELECTRÓNICA  
BOGOTÁ D.C.

2021

Diseño de un Nodo LoRa-GPS para Localización de Bicicletas Implementado en una  
Red LoRaWAN

Jairo Andrés Rodríguez Sánchez

Jhoiner Smith Rojas González

Proyecto de grado presentado como requisito para optar al título de  
INGENIERO ELECTRÓNICO

Director:

Dario Alejandro Segura Torres, Msc

UNIVERSIDAD SANTO TOMÁS  
FACULTAD DE INGENIERÍA ELECTRÓNICA  
BOGOTÁ D.C.

2021

NOTA DE ACEPTACIÓN

---

---

---

---

FIRMA DEL DIRECTOR DEL PROYECTO

---

FIRMA DEL JURADO 1

---

FIRMA DEL JURADO 2

## **Dedicatoria**

*Dedico este trabajo de grado a mi madre Martha Sánchez y mi hermana Ana María Rodríguez, quienes día a día me dio su apoyo incondicional para culminar la universidad y en todo momento estaban para brindarme sus consejos, amor y motivación.*

*Jairo Andrés Rodríguez Sánchez*

*Dedico este trabajo de grado primeramente a Dios por darme fortaleza y ganas de siempre salir adelante, a mi madre Anafloreba Gonzalez, mi padre Salvador Rojas y mis dos hermanitas Karenth e Isabela, quienes día a día me motivan no solo a ser un buen estudiante y un buen profesional, sino siempre ser una mejor persona.*

*Jhoiner Smith Rojas González*

## **AGRADECIMIENTOS**

Agradecemos a nuestro director de proyecto de grado, el Ingeniero Dario Alejandro Segura Torres, por su apoyo y acompañamiento constante para llevar a cabo la terminación del proyecto de grado. Agradecemos también a todas las personas que hicieron parte del proceso entre familiares y amigos por su constante ánimo, consejos y apoyo que recibido.

# CONTENIDO

RESUMEN .....	12
INTRODUCCIÓN .....	13
1 Problema .....	14
2 Antecedentes.....	16
3 Justificación .....	19
4 Objetivos.....	21
4.1 General .....	21
4.2 Específicos.....	21
5 Marco Teórico.....	22
5.1 LoRaWAN .....	22
5.1.1 Topología.....	23
5.1.2 Clases.....	24
5.1.3 Seguridad .....	24
5.1.4 Tasas de transferencia de datos.....	25
5.1.5 Bandas .....	25
5.2 Cifrado AES (Advanced Encryption Standard).....	26
5.2.1 Pasos de cifrado AES.....	27
5.3 Dragino LoRa/GPS Shield.....	31
5.4 Dragino LG-01.....	32
5.5 ATmega328P .....	33
5.6 GPS.....	35
5.7 RFM95 .....	37
5.7.1 Estructura de envío de paquetes LoRa .....	37
6 Diseño Metodológico .....	39
7 Desarrollo .....	41
7.1 Requerimientos del sistema .....	41
7.1.1 Requerimientos Técnicos .....	41
7.1.2 Requerimientos Funcionales .....	45
7.2 Ambiente de pruebas .....	47
7.2.1 Arquitectura de red LoRaWAN .....	47

7.2.2	Configuración del ambiente de pruebas .....	48
7.3	Diseño e implementación del nodo LoRa-GPS .....	63
7.3.1	Diseño electrónico y PCB .....	63
7.3.2	Carga de gestor de arranque de Arduino y programación del nodo LoRa-GPS	72
7.3.3	Diseño estructural.....	73
8.	Resultados.....	76
9.	Conclusiones y trabajo futuro .....	86
10.	Glosario .....	89
11.	Referencias bibliográficas .....	91
12.	Anexos .....	94
1.	Código cliente nodo LoRa-GPS.....	94
2.	Código servidor gateway LoRa.....	98
3.	Función de descifrado en la aplicación web .....	99
4.	Dockerfile para la creación de imagen contenerizada .....	100

## Lista de Figuras

Figura 1 Arquitectura LoRaWAN [23] .....	22
Figura 2 Topología estrella de estrellas de la red LoRaWAN [Autor] .....	23
Figura 3 Etapas paso a paso para cifrado y descifrado AES [27]. .....	27
Figura 4 Adición de round key para cifrado AES [27]. .....	28
Figura 5 Generación de nueva llave partiendo de la llave principal [27]. .....	29
Figura 6 Rotación de Bytes en la operación de desplazamiento de filas [27]. .....	30
Figura 7 Mezcla de columnas para incrementar la confusión [27]. .....	31
Figura 8 Nodo Lora/GPS Shield para Arduino [29]. .....	32
Figura 9 Gateway Dragino LG-01 [30]. .....	33
Figura 10 ATmega328P para montaje de superficie [32]. .....	34
Figura 11 Ilustración de satélites visibles por un GPS utilizando trilateración [34]. ...	36
Figura 12 Dispositivo GPS L-80 [35]. .....	36
Figura 13 Diagrama del diseño metodológico. [Autor]. .....	39
Figura 14 Arquitectura implementada en la solución [Autor]. .....	47
Figura 15 Acceso al servidor del gateway [Autor]. .....	49
Figura 16 Panel de control inicial del servidor del gateway [Autor]. .....	50
Figura 17 Configuración de conexión a red de internet [Autor]. .....	51
Figura 18 Configuración de Firewall dentro del gateway [Autor] .....	52
Figura 19 IDE de Arduino para configuración del puerto de programación del Gateway [Autor]. .....	53
Figura 20 Diagrama de flujo del algoritmo del gateway [Autor] .....	54
Figura 21 Diagrama de flujo del algoritmo del nodo LoRa-GPS [Autor] .....	55
Figura 22 Página de registro para el servidor ThingSpeak [Autor] .....	56
Figura 23 Configuración del canal para recibir datos [Autor]. .....	57
Figura 24 Dashboard de API Keys para el canal [Autor] .....	57
Figura 25 Visualización de los datos desde la aplicación web [Autor]. .....	58
Figura 26 Re-direccionamiento a la interfaz de Google Maps, con el dato de la posición seleccionado en la aplicación web [Autor]. .....	59
Figura 27 Listado de archivos fuente de la aplicación [Autor]. .....	59
Figura 28 Estructura Dockerfile para la construcción de imagen Docker [Autor]. .....	60
Figura 29 Comando y proceso de construcción de imagen Docker [Autor]. .....	61
Figura 30 Nombre de imagen aceptado por la nube de Google [Autor]. .....	61
Figura 31 Comando para subir la imagen Docker a la nube de Google [Autor]. .....	61
Figura 32 Comando de creación del deployment de kubernetes con la imagen Docker deseada [Autor]. .....	62
Figura 33 Comando para exponer a internet publico la aplicación [Autor]. .....	62
Figura 34 Interfaz gráfica de la nube de Google y cluster de kubernetes [Autor]. .....	63
Figura 35 Circuito electrónico del nodo LoRa-GPS [KiCad – Autor]. .....	64
Figura 36 Circuito electrónico del módulo GPS [KiCad – Autor]. .....	65



Figura 37 Circuito electrónico y conexiones del ATMEGA328p [KiCad – Autor].....	66
Figura 38 Circuito electrónico y conexiones del módulo transceptor LoRa Rfm95 [KiCad – Autor].....	67
Figura 39 Pines de entrada y salida del nodo LoRa-GPS [KiCad – Autor].....	67
Figura 40 Circuito electrónico de regulador de voltaje a 3.3v [KiCad – Autor] .....	68
Figura 41 Diseño del circuito impreso [KiCad – Autor].....	68
Figura 42 Diseño de diseño impreso capa superior [KiCad – Autor].....	69
Figura 43 Diseño de diseño impreso capa inferior [KiCad – Autor].....	70
Figura 44 Modelo en 3D del circuito impreso capa superior [KiCad – Autor] .....	70
Figura 45 Modelo en 3D del circuito impreso capa inferior [KiCad – Autor] .....	71
Figura 46 Circuito impreso soldado, capa superior [Autor].....	72
Figura 47 Circuito impreso soldado, capa inferior [Autor].....	72
Figura 48 Diseño estructural completo para el nodo [Autor]. .....	74
Figura 49 Diseño estructural por partes para el nodo [Autor]. .....	74
Figura 50 Estructura impresa en 3D para el nodo [Autor]. .....	75
Figura 51 Nodo LoRa-GPS integrado a una bicicleta [Autor]. .....	75
Figura 52 Prueba #1 - Recorrido alrededor de la Universidad Santo Tomas con los datos obtenidos por el nodo LoRa-GPS diseñado [40]. .....	76
Figura 53 Prueba #1 - Densidad de predios cerca de la Universidad Santo Tomas donde se realiza el recorrido Fuente: Mapas de Infraestructura de Datos Espaciales de Bogotá (IDECA) [41].....	77
Figura 54 Prueba #1 - Medida del punto más lejano de comunicación del nodo LoRa-GPS con el gateway [40]......	78
Figura 55 Prueba #2 - Recorrido de prueba con los datos obtenidos por el nodo LoRa-GPS diseñado [40]. .....	79
Figura 56 Prueba #2 - Densidad de predios en la zona de Villa Mayor donde se realiza el recorrido Fuente: Mapas de Infraestructura de Datos Espaciales de Bogotá (IDECA) [41]. .....	80
Figura 57 Prueba #2 - Medida del punto más lejano de comunicación del nodo LoRa-GPS con el gateway a línea de vista [40]......	81
Figura 58 Prueba #2 - Medida del punto más lejano de comunicación del nodo LoRa-GPS con el gateway con obstáculos [40]. .....	81
Figura 59 Prueba #3 – Recorrido de prueba (Calle 51, Bogotá) con los datos obtenidos por el nodo LoRa-GPS diseñado en línea de vista [40]......	82
Figura 60 Prueba #3 - Medida del punto más lejano de comunicación del nodo LoRa-GPS con el gateway a línea de vista [40].....	83
Figura 61 Consumo de corriente del nodo LoRa-GPS en estado de reposo y envió de datos [Autor]......	84
Figura 62 Medición de la potencia en la frecuencia de 914.9 MHz al enviar de un paquete LoRa 5 metros de distancia entre el gateway y el nodo LoRa-GPS con obstáculos [Autor].....	85

Figura 63 Medición de la potencia en la frecuencia de 914.9MHz al enviar de un paquete LoRa 20 centímetros de distancia entre el gateway y el nodo LoRa-GPS sin obstáculos [Autor]..... 85

## Lista de Tablas

Tabla 1 Parámetros importantes ATmega328P. ....	34
Tabla 2 Características de las tecnologías de geolocalización [38]. ....	43
Tabla 3 Reglas configuradas en KiCad para el diseño de la PCB [Autor] .....	69
Tabla 4 Conexiones para carga de gestor de arranque y programación del Nodo LoRa-GPS [Autor] .....	73

## RESUMEN

En el presente proyecto de grado se desarrolló un dispositivo de rastreo para poder ser implementado en una bicicleta y de esta manera añadir un nivel de seguridad adicional, utilizando los conceptos teóricos y prácticos de IoT, computación en la nube y diseño electrónico. Para alcanzar satisfactoriamente los objetivos planteados, se propuso trabajar en varias etapas. Inicialmente, fue necesario identificar la necesidad y el problema y a partir de este buscar y plantear una solución práctica para la mayoría de las personas que tienen una bicicleta como medio de transporte. Se establecieron los objetivos y el diseño metodológico, se realizó la selección de las tecnologías a utilizar, seleccionando diferentes dispositivos que mejor se acoplan a la solución.

Se implementó un ambiente de pruebas basado en la tecnología LoRa desarrollado con un gateway, un dispositivo de prueba comercial y una página web desarrollada en Angular para la visualización de los datos obtenidos. Se realizó el diseño y construcción del dispositivo final con métricas y diseño propio, para cumplir el objetivo principal del proyecto, pero a su vez pueda ser mimetizado en la bicicleta. Se estableció la conexión entre el sistema embebido y el gateway que son los encargados de procesar y enviar los datos a la nube, en este caso el servidor escogido fue ThingSpeak y, finalmente se verificó el correcto funcionamiento de la aplicación web que permite visualizar los datos almacenados en la nube.

Los resultados finales mostraron que el sistema presenta una respuesta acorde a los objetivos planteados, ya que se desarrolló con éxito un dispositivo de rastreo para bicicletas con una precisión y alcance similar a la que nos brinda un nodo LoRa-GPS comercial, pero con la ventaja de que el dispositivo desarrollado es mucho más pequeño y fácil de mimetizar e integrar en una bicicleta. Además se logró diseñar aplicación web que permite visualizar las diferentes coordenadas obtenidas por el dispositivo y enviadas al servidor por medio de una comunicación con cifrado seguro de datos.

## INTRODUCCIÓN

En el siglo XXI, la tecnología y las comunicaciones brindan la posibilidad de conectar y centralizar información que sea útil y tenga alguna consecuencia a futuro, de esta manera surge una tendencia constante de conectar todo tipo de objetos a internet, lo que hoy en día se conoce como, internet de las cosas. En este proyecto de grado se pusieron en práctica conocimientos de la tecnología LoRaWAN, con el diseño de un Nodo LoRa-GPS para la localización de bicicletas en la ciudad de Bogotá, con la cual aumentar la posibilidad de recuperar una bicicleta robada.

Para lograr este objetivo fue necesario identificar las características técnicas y funcionales de un sistema de rastreo para bicicletas, que permitieran obtener los lineamientos electrónicos y físicos del dispositivo, para su integración tanto a una red LoRaWAN como a la bicicleta. En la primera fase del proyecto, se desarrolló una red LoRaWAN con un Nodo comercial, el cual se utilizó como un ambiente de pruebas que luego se utilizaría para integrar el dispositivo diseñado y comprobar su funcionamiento, además se realizó una aplicación web que le permite ver de manera gráfica la información de geolocalización recolecta por el nodo al usuario.

En la segunda fase se llevó a cabo la implementación de Nodo LoRa-GPS, siguiendo los lineamientos encontrados en la fase uno. En primer lugar, se realizaron las pruebas de funcionamiento y validación de los diferentes módulos que componen el Nodo siguiendo los lineamientos, posteriormente se diseñó con un software de código abierto el circuito electrónico y la PCB o circuito impreso que soporta y lleva la conexión de los componentes electrónicos.

Por último, en la tercera fase se integró el Nodo desarrollado al ambiente de pruebas, para validar su funcionamiento en campo, dando una aproximación de su uso cotidiano y finalmente se obtuvieron los resultados presentados más adelante en este documento al igual que las conclusiones que por el diseño, construcción y funcionamiento resaltaron.

# 1 PROBLEMA

En Bogotá, se ha venido popularizando la bicicleta como medio de transporte, un indicador de esto, es que entre el año 2011 al 2015, aumentó su uso en un 38.5% con cifras de 635.431 viajes diarios para el 2015 y en el año 2019, fueron aumentando hasta llegar a 880.367 viajes diarios que corresponden al 44% de aumento, época en la que se registra que aproximadamente un 5% de los bogotanos la utilizan como medio de transporte alternativo [1].

Los principales motivos por los que los bogotanos han elegido la bicicleta como medio de transporte alterno son: por tener aproximadamente 500 kilómetros de vías construidas para ciclistas; porque según cita textual del distrito capital se cuenta con *“6.921 cupos de CicloParqueaderos habilitados por el Distrito, de estos 4.819 están en portales y estaciones del sistema Transmilenio”* [2], y por último, pero para algunos el más importante, por reducir costos de movilización. Estas razones indirectamente ayudan a los biciusuarios a tener un estilo de vida saludable, y a reducir el daño ambiental causado por el uso de combustibles fósiles por parte del parte automotor.

Lamentablemente las bicicletas en la actualidad son un blanco fácil para los delincuentes, ya que la seguridad es muy precaria, presentando reportes diarios de 19 hurtos, de los cuales algunos han terminado en tragedia. Así por ejemplo, durante el año 2018 se registraron 33 ciclistas muertos por hurtos o accidentes viales [3], lo cual causa indignación entre los usuarios y la comunidad: *“Para algunos líderes del ciclismo urbano en Bogotá, lo que está demostrado es que cada vez son más apetecidos estos vehículos, por dos razones específicas: las bicicletas son más fáciles de comercializar en el mercado ilegal que un celular y el apoyo judicial y estatal es insuficiente para evitar la expansión de ese mercado negro”* [4]. Estas bicicletas robadas aumentan en gran medida el índice de ventas ilegales en la ciudad, las cuales mueven gran cantidad de dinero, como lo expresa Andrés Nieto: *“El mercado negro de bicicletas robadas genera al mes casi \$320 millones, adicionales a los de partes de bicicleta, que puede mover hasta \$160 millones”* [5]. experto en seguridad ciudadana de la Universidad Central.

Los delincuentes causan daños a la integridad física, psicológica y monetaria de las víctimas, con el fin de lucrarse ya sea a través de ventas ilegales de repuestos, o mediante la reventa de las bicicletas, debido a que muy pocos ciudadanos que pierden su bicicleta pueden llegar a recuperarla. Esto genera temor al uso de este

medio de transporte, además de gastos para una nueva bicicleta, o afectaciones por las heridas que pueda causar el victimario, llegando incluso a la muerte.

Ahora, si se hace referencia a los dueños de bicicletas o específicamente a los biciusuarios que usan a diario este vehículo como medio de transporte, ellos no adquieren dispositivos de localización, la mayoría de veces por el costo tan elevado que presentan, y además por el hecho que casi todos estos dispositivos se basan en tecnología GSM, la cual requiere un módulo de suscripción de identificación (Sim-Card) con datos, para lograr activar la funcionalidad, que además del servicio de geolocalización, genera un costo adicional que se sufraga en cuotas, generando una renta mensual adicional, que para muchos de los usuarios les resulta muy difícil o incluso imposible de pagar. Por otra parte, adquirir un seguro para la bicicleta, es otra opción, pero solo se puede obtener si el costo de la bicicleta es mayor a \$500.000 pesos colombianos.

Por todo lo anterior es importante saber ¿Qué características debe tener un sistema de localización que permita incrementar la probabilidad de recuperar una bicicleta hurtada, en la ciudad de Bogotá?

## 2 ANTECEDENTES

La red de LoRaWAN es usada para múltiples aplicaciones y cumple con los requerimientos característicos de Internet de las cosas, inclusive para uso específico de localización. En los últimos años han surgido pequeñas empresas prestadoras de servicios en esta tecnología, nuevas investigaciones que compraran la diversidad de aplicaciones y promueven el uso de LoRaWAN para los próximos años, caracterizándose por el bajo consumo de potencia y largo alcance, características que dan gran visión a esta tecnología.

En primera instancia en la Universidad de Malmö se desarrolló para esta ciudad, una comparación entre la tecnología Lora y GPS, en donde enviaban la información de la localización actual entre cada una. Por su parte el GPS mostró un rango de localización de 10m de la posición actual, superando a LoRa, que en su caso se realizaron pruebas con triangulación, donde envían mensajes a referencias con TDoA, RSSI y TTN, cada una de estas referencias, mostró un rango de 200-280m de distancia de la posición original. Pero a pesar de este rango tan grande, los beneficios de bajo consumo, y la duración de la batería, son un beneficio que lleva esta tecnología [6].

Dicho esto, se ha llevado a pruebas y prototipos de localización en exteriores e interiores, varios de ellos presentados en conferencias o tesis de grado. Por ejemplo en undécima conferencia internacional sobre computación móvil y red ubicua (ICMU) de 2018, se presentó *“Compression Method of Position Information for IoT-based Bus Location System Using LoRaWAN”*, en el cual se usa la red LoRaWAN para enviar una estimación del tiempo de recorrido y llegada del bus, por medio de GPS instalados en cada bus, realizando un cálculo según los datos tomados por el GPS en un base de datos, que luego en cada parada de autobuses, se transmite la información a través de RF por proximidad entre el autobús y el paradero, de esta manera informa la localización estimada y la duración de llegada al paradero, con el fin de no estar transmitiendo la localización en todo momento y comprimiendo los datos para no subir datos a la nube [7].

Pero hay casos donde estar enviando datos de la localización en tiempo real, es necesario al mismo tiempo guardándolo en un servidor para tener un histórico, información que será necesaria según cada aplicación, a esto se hace referencia un trabajo presentado en la Conferencia Internacional Anual de la Sociedad de Ingeniería en Medicina y Biología, en la cual se presentó un rastreador LoRa GPS para pacientes con demencia, con el cual se busca tener la localización exacta de cada paciente a



3Km de distancia, por medio de una manilla la cual tiene una duración de 40 horas continuas enviando la localización cada 60 segundos [8].

Sistema de información de apoyo a las decisiones en la agricultura de precisión basado en redes de sensores remotos y herramientas de sistemas de información geográfica (SIG), es un trabajo final de graduación presentado en la Universidad Siglo 21 en el año 2020 en Argentina, donde el autor del documento utiliza las tecnologías disponibles de redes de sensores remotos LoRa, caracterizados por un mínimo consumo de energía y poder funcionar bajo condiciones adversas, para realizar un monitoreo y análisis de las variables meteorológicas y fisiológicas de los cultivos de su entorno tomando decisiones que permitan aumentar la productividad de los cultivos, logrando una implementación para el acceso a los productores agrícolas de la provincia de Mendoza, desde grandes empresas, pymes y empresas familiares para los que su único sustento es la agricultura [9].

Implementar un sistema de seguimiento de Internet de las cosas basado en el protocolo LoRa, es un trabajo publicado en 2018, en donde se concluye que, en la vía urbana, el alcance que tiene esta red, con los paquetes enviados y recibidos correctamente, versus una carretera, es mucho mayor, dado que en carretera se pierden demasiados paquetes de datos, debido al entorno y diferentes cambios de alturas proporcionadas en las carreteras. En el estudio: *“Un resultado del experimento es la detección de un gran rendimiento del protocolo LoRa cuando se enfrentan a las características de las cosas en movimiento”* [10].

Por otro lado, Dragino es una empresa, que ofrece tecnología por medio de IoT con protocolo LoRaWAN, algunos productos que ofrece van destinados a la localización con dispositivos de rastreo GPS o LoRa Geolocalización, agricultora o estaciones meteorológicas, además de gateway's para LoRaWAN de uno o varios canales [11]. Otras empresas como Catsensors o Duolink venden sensores LoRaWAN específicos para otras aplicaciones como medición de temperatura, nivel, velocidad humedad, o servicios de monitoreo de los sensores mencionados anteriormente incluyendo también el monitoreo geotécnico [12] , [13].

Aunque LoRaWAN es un protocolo de red que no está tan expandido como SigFox, en los últimos años se vienen desarrollando diversos proyectos entorno a LoRaWAN para aplicaciones de IOT, dando más reconocimiento a este protocolo de red, que se espera llegue a estar tan implementado como SigFox. Por ejemplo, Relay Cat Board, es uno de los más recientes dispositivos creados para la industria, este es un dispositivo compatible con LoRaWAN, está especialmente creado para interactuar

con dispositivos con lógica de relevadores como PLCs que permite el control de motores, actuadores, luces y cualquier otro componente de la industria [14].

En la Universidad Distrital Francisco José de Caldas en Bogotá Colombia, se desarrolló un proyecto en el programa Ingeniería Electrónica el cual tenía como objetivo *“Implementar una red de sensores inalámbricos LPWAN mediante el protocolo LoRa para el monitoreo de los parámetros de calidad del agua (CE, pH, TDS, T).”* [15]. Un año más tarde en 2018 en la misma universidad, se desarrolló un proyecto en la facultad de tecnología en el programa de telecomunicaciones, el cual tenía como objetivo *“Diseñar e implementar un prototipo de solución IoT con tecnología “LoRa” para el monitoreo de cuatro variables físicas (humedad, temperatura, pH y radiación UV) con un aplicativo web en un cultivo de fresas.”* [16].

En 2019 en la Universidad Santo Tomás en la Facultad de Ingeniería de Telecomunicaciones, se llevó a cabo un sistema de apoyo a la prevención del abigeato de ganado bovino utilizando tecnologías LoRa, IoT y Cloud. Este sistema captura y monitorea datos en tiempo real de la ubicación de los animales que se encuentran en una zona limitada mediante una cerca virtual, generando alarmas si estos salen de la propiedad y dar aviso a los ganaderos, contribuyendo a reducir el factor de abigeato en las zonas ganaderas de Une, Cundinamarca [17].

Por último, en el año 2020 en la sede de Bucaramanga de la Universidad Santo Tomás, se presentó un proyecto de grado para optar por el título de Magister en Redes y Sistemas de Comunicación. Dicho proyecto de grado se encuentra bajo el título de *“Solución IoT para la optimización del proceso de piscicultura en el Centro de Desarrollo Agroalimentario El Limonal”*, cuyo objetivo era monitorear en tiempo real y de manera remota diferentes variables del proceso, para de esta manera poder tomar decisiones que ayuden a mejorar los índices en la multiplicación de los peces. La conclusión a la que se llegó al finalizar el proyecto es que bajo una utilización de LoRa tendrían una implementación óptima para llevar este proyecto a fincas más lejanas [18].

### 3 JUSTIFICACIÓN

En las bicicletas, la utilización de los candados, cerraduras y cadenas sigue siendo necesario y muy eficiente al momento de parquear el vehículo, lo que brinda un grado de seguridad al estar detenido, pero al momento de moverse estos métodos anteriores no aplican. Además, en una ciudad como Bogotá es inevitable que los bicisuarios sigan siendo víctimas de hurto y la probabilidad de recuperar la bicicleta es mínima, porque solo el 3.58%<sup>1</sup> de las bicicletas que se hurtan en la ciudad de Bogotá son recuperadas, principalmente por no tener un sistema de rastreo que permita su localización, generando incertidumbre en el propietario al no saber dónde ha sido llevada por el ladrón [19] , [20].

Con el desarrollo de este proyecto se crea y se pone en funcionamiento un sistema de localización, mediante la implementación de internet de las cosas, el cual dentro de sus diversas aplicaciones incluye el campo de la localización, adicionando el protocolo de comunicación LoRaWAN, el cual tiene una gran visión respecto a la comunicación eficiente por tener características de bajo consumo y largo alcance. Logrando implementar, desarrollar y colocar en funcionamiento un dispositivo integrado a una bicicleta, con el fin de enviar su geolocalización sin necesidad de estar conectado a internet o sin tener acceso a una señal GSM.

La red LoRaWAN implementada, permite enviar la posición de la bicicleta a un servidor de red (Nube), que almacena y permite extraer esta información, para analizarla, procesarla y visualizarla en una aplicación web. Se escoge un GPS, ya que resulta más preciso que la triangulación realizada por el GSM, que trabaja a un rango de localización de aproximadamente 50m en el mejor de los casos, mientras que la de un GPS llega a ser menor a 10m [21] y por otro lado también supera a la LoRa Geolocalización por medio de que llega a tener una precisión error mínimo de 73,8 metros mientras que el error máximo fue de 178,2 metros, por medio de TDOA [22].

Al momento del hurto de una bicicleta que cuente con un sistema de localización GPS, la probabilidad en cuanto a opción de localizarla y recuperarla es más alta, ya que, si se conoce la ubicación del vehículo, es obvio que aumentan las posibilidades de recuperarla, o se tiene un punto de referencia para dar con su ubicación actual.

---

<sup>1</sup> Hallado del número de bicicletas robadas y las que se han sido recuperadas en el 2018

Además, por medio del sistema, se busca salvaguardar la integridad del bicusuario para que no sea afectada tanto física, como económicamente, puesto que al momento del delito el dueño de la bicicleta no opondrá resistencia, cediendo a entregarla, dado que el dispositivo le ofrece más posibilidades de recuperarla y no verse obligado a invertir en la compra de una nueva bicicleta.

Paralelamente a la solución general, el nodo LoRa-GPS desarrollado en este proyecto, es factible aplicarlo en otros ámbitos de localización, ya que la tecnología no tiene limitaciones de uso; por ejemplo, el sistema se puede utilizar para el monitoreo de envíos a domicilio, para logística de ubicación de vehículos en una empresa de transporte o cualquier implementación que requiere rastrear y tener un informe de geolocalización.

## 4 OBJETIVOS

### 4.1 General

Diseñar e implementar un Nodo LoRa GPS de localización para incrementar la posibilidad de recuperar una bicicleta robada.

### 4.2 Específicos

- Identificar las diferentes características técnicas y funcionales de un sistema de rastreo para bicicletas, para obtener los lineamientos del diseño electrónico que debe tener un nodo que integre este sistema.
- Diseñar un nodo LoRa-GPS para cumplir las características identificadas.
- Desarrollar un nodo LoRa-GPS propio bajo unas características técnicas y funcionales que permitan implementarlo en el rastreo de bicicletas.
- Implementar un ambiente de pruebas que conste de una red LoRaWAN con nodos comerciales.
- Desarrollar una aplicación para el ambiente de pruebas, que permita la visualización de los datos de geolocalización, proporcionados por los nodos.
- Integrar el nodo diseñado al ambiente de pruebas para validar su funcionamiento.

## 5 MARCO TEÓRICO

A continuación, se presenta los soportes teóricos necesarios para desarrollo del proyecto.

### 5.1 LoRaWAN

Es un protocolo LPWA (Low Power Wide Area), diseñado para conectar "cosas" de forma inalámbrica a Internet en redes regionales, nacionales o mundiales. Esta red se dirige a requisitos clave del Internet de las cosas, como la comunicación bidireccional, la seguridad de extremo a extremo, la movilidad y los servicios de localización. Su arquitectura esta descrita en la

Figura 1 [23].

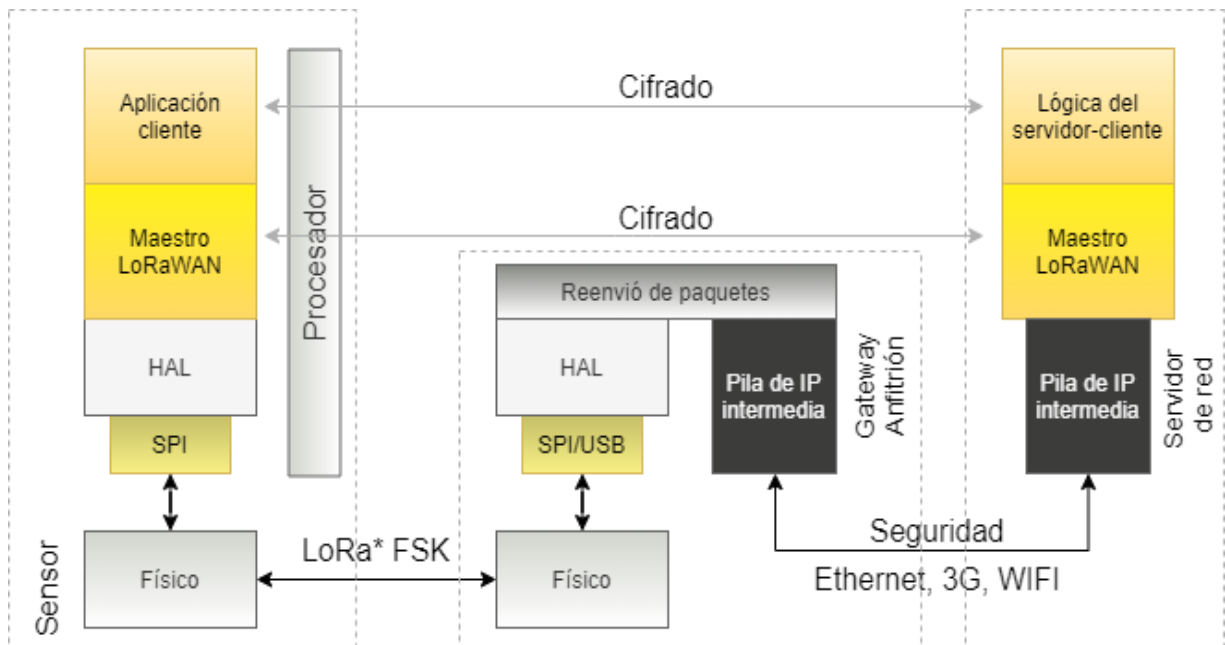


Figura 1 Arquitectura LoRaWAN [23]

### 5.1.1 Topología

La arquitectura de una red LoRaWAN que se muestra en la

Figura 1 se despliega en una topología de estrella de estrellas como se ilustra en la Figura 2, en donde cada gateway, retransmite los mensajes de cada nodo final a un servidor de red (nube).

Cada gateway está conectado al servidor de red a través de una conexión IP estándar, que realiza el puente para convertir los paquetes de RF en paquetes IP y viceversa [23].

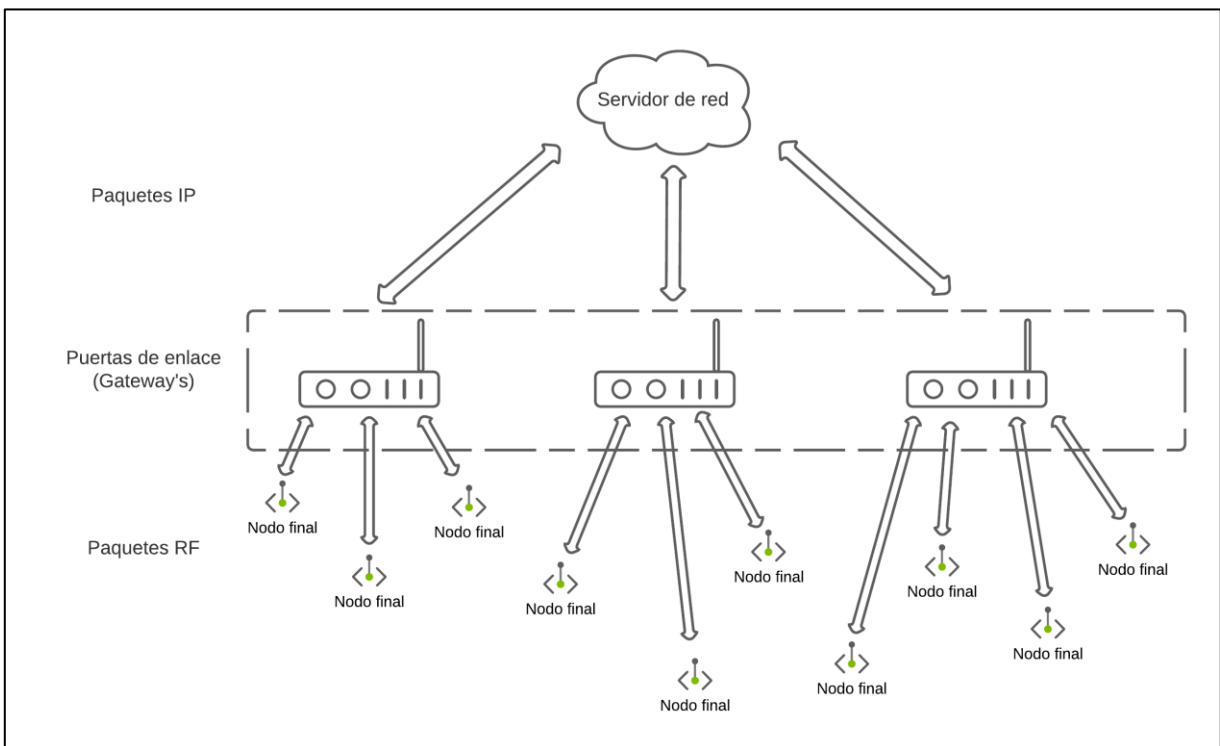


Figura 2 Topología estrella de estrellas de la red LoRaWAN [Autor]

### 5.1.2 Clases

LoRaWAN cuenta con tres clases diferentes de dispositivos con los cuales brinda la oportunidad de tener una amplia gama de aplicaciones, estas clases y sus características son:

- **Clase A:** Esta clase la soportan todos los nodos de LoRaWAN, la comunicación de esta clase es asíncrona y el encargado de iniciarla es el mismo nodo. Esta comunicación ascendente abre dos ventanas de recepción en las cuales puede recibir información en cualquier momento, lo que permite que la comunicación sea bidireccional.

Otra ventaja es que el nodo puede entrar en un estado de reposo cuyo objetivo es tener un bajo consumo de corriente durante el tiempo definido.

- **Clase B:** Además de lo mencionado con respecto a las ventanas de recepción en un dispositivo clase A, los dispositivos de clase B se conectan y realizan la sincronización con la red mediante balizas periódicas que abren "franjitas de ping" para poder programar el envío de información a determinadas horas.

Además, esta clase permite tener una latencia programable de hasta 128 segundos lo que la hace tener la capacidad de adaptarse a distintas aplicaciones.

- **Clase C:** Además de la estructura de clase A, los dispositivos clase C reducen la latencia en la conexión de enlace descendente ya que dejan abierto el canal del nodo a toda hora, incluso cuando el nodo no está transmitiendo y por esto, la clase C es la que más consumo de energía tiene [23].

### 5.1.3 Seguridad

La seguridad es una de las características más importantes en la transmisión de información mediante LoRa, ya que se tiene la opción de cifrar los datos con claves de



hasta 256 bytes y este cifrado se realiza mediante algoritmos AES que proporcionan autenticación e integridad de los paquetes al servidor de red y cifrado de extremo [23].

#### **5.1.4 Tasas de transferencia de datos**

La comunicación entre los diferentes nodos y los gateway que se hace por paquetes, permite ajustar la velocidad con la que se transfieren los datos y esto mediante la utilización en una variable.

Para aumentar al máximo la autonomía de la batería de los nodos y la capacidad total de la red, el servidor o gateway de red LoRaWAN ajusta la velocidad de transmisión de los datos y la potencia de transmisión hacia cada nodo mediante un ADR (velocidad de datos adaptativa). Estas velocidades de LoRaWAN en la transmisión puede ir desde 0,3 kbps hasta 50 kbps [23].

#### **5.1.5 Bandas**

La tecnología LoRaWAN permite la utilización de 64 canales para realizar una conexión de enlace ascendente de 125kHz de 902,3MHz a 914,9MHz en saltos de 200kHz. Se tienen ocho canales que permiten un enlace ascendente de 500kHz adicionales en saltos de 1.6MHz de 903MHz a 914,9 MHz. Los ocho canales de enlace descendente tienen un ancho de 500 kHz a partir de 923,3 MHz a 927,5 MHz. La potencia de salida máxima en la banda de América del Norte 902-928MHz es +30dBm, pero para la mayoría de los dispositivos + 20dBm es suficiente. Bajo FCC no hay limitaciones de ciclo, pero hay un tiempo máximo de permanencia de 400 ms por canal.

Además, LoRaWAN trabaja en diferentes bandas y canales a nivel global, por lo tanto, cada país regula las bandas a utilizar, Según LoRa Alliance en Colombia están avaladas dos bandas para el uso de esta red, 433-434.79 MHz (EU433) y 915-928MHz (AU915-928) [24]. Sin embargo, la Agencia Nacional del Espectro (ANE) entidad que vigila y controla el espectro radioeléctrico, en el territorio colombiano, establece en la RESOLUCIÓN N° 000711 DEL 11 OCT. 2016, las bandas de radio industriales, científicas y médicas (ICM) libres, donde se establece la banda de 915-928MHz para uso de LoRaWAN [25].

## 5.2 Cifrado AES (Advanced Encryption Standard)

El estándar de cifrado avanzado (AES) especifica el algoritmo Rijndael, un cifrado de bloques simétrico que puede procesar bloques de datos de 128 bits, utilizando claves de cifrado con longitudes de 128, 192 y 256 bits, El algoritmo puede utilizarse con las tres longitudes de clave diferentes denominadas: "AES-128", "AES-192" y "AES-256" [26].

El algoritmo AES no sólo tiene un gran conjunto de claves ( $2^{128}$  claves posibles), sino que también es seguro ante algoritmos de criptoanálisis como, el criptoanálisis diferencial, la integración multiconjunto y similares; considerado el más seguro frente a todos los ataques conocidos hasta el momento [27] , [28]. El flujo del algoritmo AES se muestra en la Figura 3. La mitad izquierda de la figura muestra el flujo de cifrado y la mitad derecha el de descifrado.

Para desglosar la Figura 3 se puede describir el proceso de cifrado en 5 pasos secuenciales:

- Añadir Round Key
- Sustituir bytes
- Desplazar filas
- Mezclar columnas
- Añadir Round Key

De igual manera, el proceso de descifrado también tiene 5 pasos secuenciales:

- Añadir Round Key
- Mezcla inversa de columnas
- Desplazamiento inverso de filas
- Sustitución inversa de bytes
- Añadir Round Key

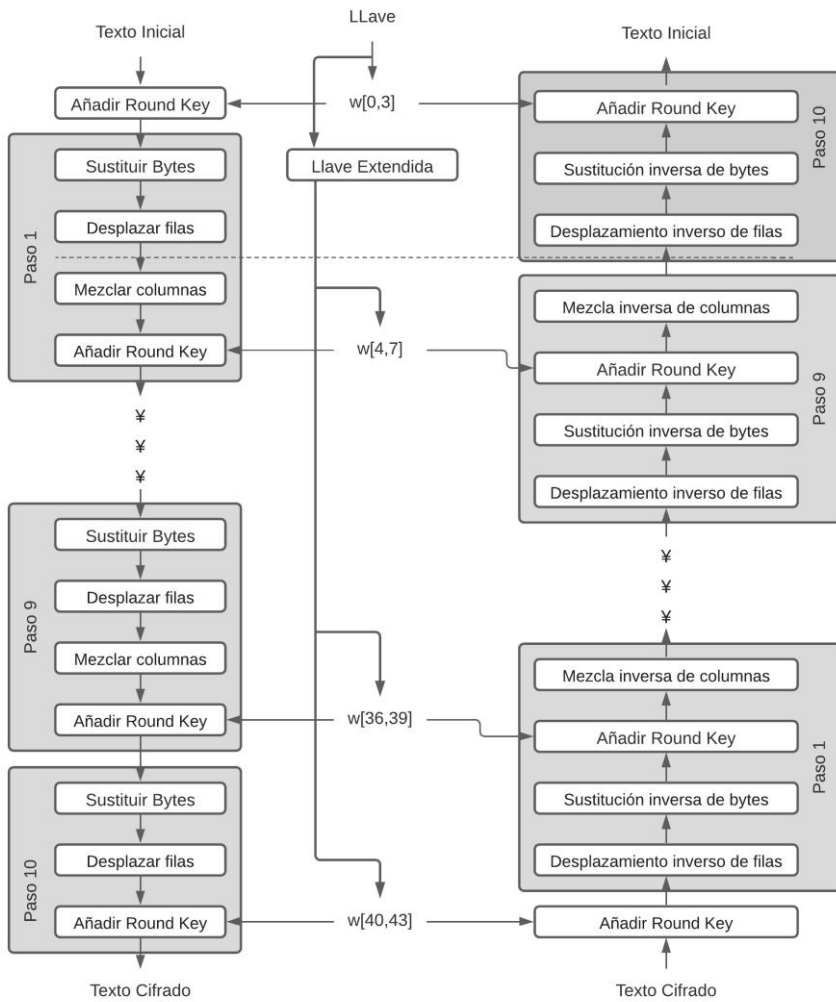


Figura 3 Etapas paso a paso para cifrado y descifrado AES [27].

### 5.2.1 Pasos de cifrado AES

A continuación, se van a explicar cada uno de los pasos por separado para el proceso total del cifrado con AES.

### 5.2.1.1 Añadir Round Key

La suma de la *Round Key* se muestra en la Figura 4. Es la suma XOR de las cadenas binarias del mensaje y la clave. La condición para esta suma es que la longitud de la cadena de mensajes y la clave deben ser iguales. Además, el cifrado AES mapea el mensaje en una matriz cuadrada (4x4 bytes o 32x32 bits).

En la operación de adición de claves, a los primeros 32 bits del mensaje se realizan una operación XOR con la primera columna de la matriz de claves. La adición de la clave da el texto cifrado de la sub-ronda. En la Figura 5 se muestra la generación del Round Key. Para el primer paso, se utiliza una llave de entrada, y la clave para el resto de las rondas se calculan a partir de la llave del paso anterior, el mensaje de texto plano y la semilla (cualquier palabra de 4 bytes o cadena binaria de 32 bits).

La llave del paso anterior se asigna en una matriz cuadrada de elementos de 4x4 bytes. En el siguiente paso, cada columna de la matriz de llaves se combina con la columna respectiva de la palabra.



Figura 4 Adición de round key para cifrado AES [27].

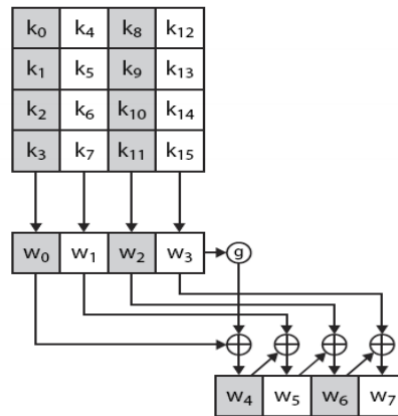


Figura 5 Generación de nueva llave partiendo de la llave principal [27].

### 5.2.1.2 Sustituir Bytes

Tras calcular la llave del siguiente paso y la suma de la llave para el texto cifrado (o texto plano para el primer paso), cada byte es sustituido por la respectiva tabla de Rijndael (una tabla estándar de 256 valores) para aumentar la confusión.

### 5.2.1.3 Desplazamiento de filas

Tras la sustitución, los 16 bytes se distribuyen para construir una matriz cuadrada de 4x4. En la matriz resultante, se aplica la operación de rotación de filas, la cual se muestra en la Figura 6 y se explica a continuación:

- La primera fila de la matriz no se modifica.
- La segunda fila de la matriz se gira un byte a la izquierda.
- La tercera fila se gira dos bytes a la izquierda.
- La cuarta fila se gira a la izquierda en tres bytes (o se gira a la derecha en un byte).

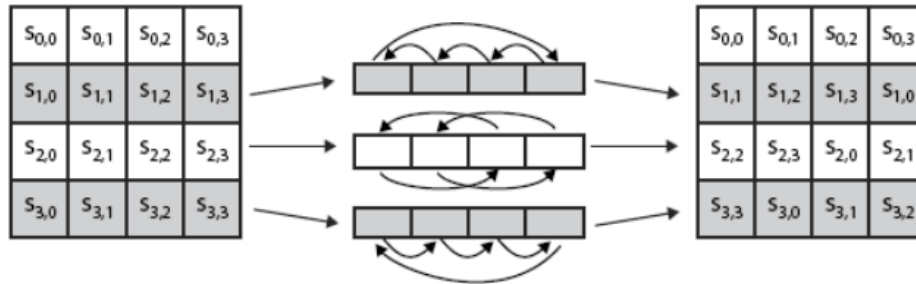


Figura 6 Rotación de Bytes en la operación de desplazamiento de filas [27].

#### 5.2.1.4 Mezcla de columnas

Como se muestra en la Figura 7, la multiplicación matricial por la izquierda se aplica utilizando una matriz 4x4 sobre los resultados de la operación de desplazamiento de filas. Para obtener los resultados de la multiplicación por dos sin pérdida de información se consigue en dos pasos. Aplicar el desplazamiento a la izquierda sobre el valor binario de los datos y añadir un 0 en el lado LSB. El segundo paso depende de si el MSB era 0 o 1. El segundo paso tiene dos formas dependiendo del valor MSB:

- Si el MSB de los datos era 0, entonces el resultado final de la multiplicación de la columna  $S_{x,2}$  sigue siendo el mismo que el del desplazamiento a la izquierda en un bit.
- Si el MSB de los datos era 1, entonces se obtiene el resultado final de la multiplicación de  $S_{x,2}$  calculando la suma XOR de los resultados del desplazamiento a la izquierda y (00011011).

La multiplicación de los datos es bastante sencilla, es la suma XOR de los  $S_{x,2}$  y los datos.

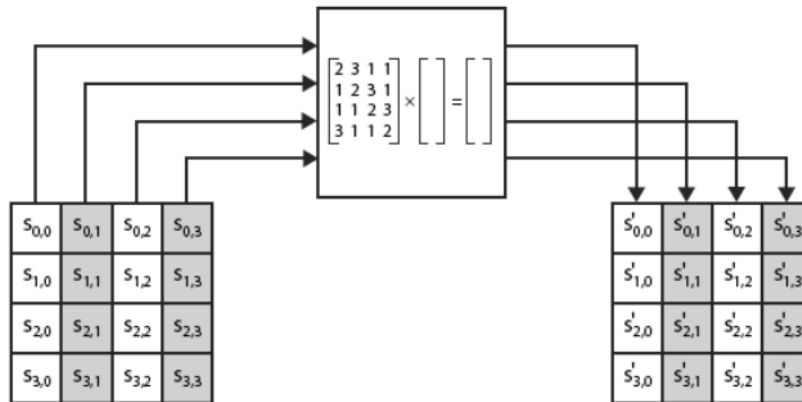


Figura 7 Mezcla de columnas para incrementar la confusión [27].

### 5.3 Dragino LoRa/GPS Shield

El Dragino *LoRa/GPS Shield* (Figura 8), es una placa de expansión para LoRa/GPS para usar con Arduino. El *LoRa/GPS Shield* está compuesto por la placa madre *LoRa/GPS Shield* y Lora BEE. Los nodos *LoRa/GPS Shield* cuentan con la capacidad de conectarse a un gateway de largo alcance LoRa que proporciona una comunicación de alto espectro con largo alcance y una alta inmunidad sobre las interferencias al mismo tiempo que minimiza el consumo de energía en forma de corriente.

Ahora bien, el dispositivo GPS L80 está diseñado para distintas aplicaciones que utilizan un GPS conectado a través de los puertos serie de Arduino. Este módulo GPS está diseñado para predecir y calcular órbitas automáticamente utilizando datos almacenados en la memoria interna (flash) de hasta 3 días, permitiendo al nodo *LoRa/GPS Shield* fijar la posición satelital con un consumo de energía bajo.

La tecnología *AlwaysLocate*, del nodo Lora/GPS puede ajustar el tiempo de encendido y el tiempo de apagado logrando un equilibrio entre el consumo de energía y la precisión, todo esto teniendo consideraciones del movimiento del dispositivo y las condiciones ambientales [29].

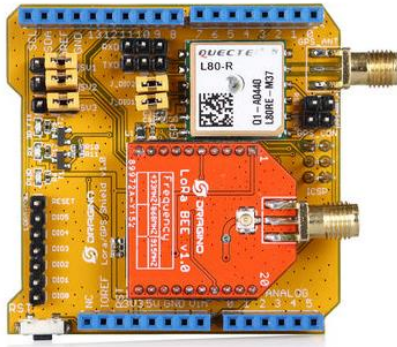


Figura 8 Nodo Lora/GPS Shield para Arduino [29].

Dentro de las características principales y ventajas tecnológicas se tienen.

- Bandas de frecuencia de 868 MHz, 433 MHz y 915 MHz.
- Bajo consumo de energía.
- Compatibilidad con Arduino Leonardo, Mega y Uno.
- Modulación por FSK, GFSK, MSK, GMSK, LoRa y OOK.
- Conmutación automática de GPS entre antena de parche interna y antena activa externa.
- Tasa de bits programable hasta 300 kbps.
- Entre otras.

#### 5.4 Dragino LG-01

El gateway LG-01 (Figura 9), muestra un gateway LoRa de un solo canal mediante el cual se puede conectar la red LoRa a una red IP en conexión inalámbrica WiFi, red por cable Ethernet, o red mobile ya sea 3G o 4G con el fin de servir como un enrutador de red para recibir datos de los nodos y subirlos a un servidor. La red LoRa generada, permite el envío de datos a largos rangos de transmisión con bajos costos energéticos.

Dentro de las especificaciones técnicas y características más importantes se tienen las siguientes:

- Posee un procesador de 400MHz, 24K MIPS.
- Tiene una memoria flash de 16MB y una memoria RAM de 64MB.
- Sistema *OpenWrt* el cual es de código abierto.



- Bajo consumo de energía.
- Banda LoRa disponible a 433/868/915/920 MHz
- Servidor web incorporado
- Conexión a Internet a través de LAN, Wifi, 3G o 4G
- Compatible con el IDE de Arduino.
- Máximo rango LoRa: 5~10 km. Densidad de área: >500m

Estas son solo algunas de las ventajas y características que se pueden encontrar en un gateway LoRa [30].



*Figura 9 Gateway Dragino LG-01 [30].*

## 5.5 ATmega328P

El ATmega328P (Figura 10) es un microcontrolador AVR de 8 bits de alto rendimiento y bajo consumo que es capaz de lograr la máxima ejecución en un solo ciclo de reloj de 131 instrucciones gracias a su avanzada arquitectura RISC. Es habitual encontrarlo como procesador en placas Arduino como Arduino Fio y Arduino Uno.

El ATmega328P es compatible en cuanto a conexión con diferentes herramientas para desarrollo de software o sistemas que incluyan: Compiladores C, depuradores, simuladores de programas o ensambladores [31].

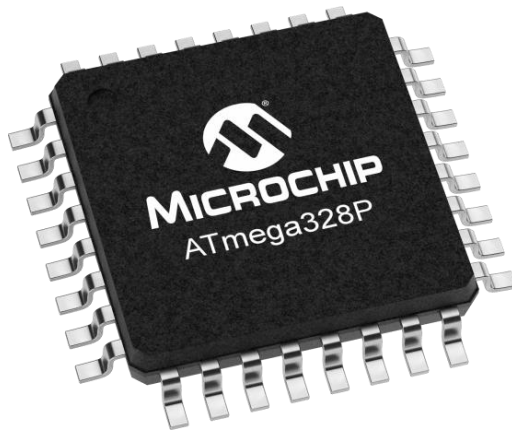


Figura 10 ATmega328P para montaje de superficie [32].

Dentro de sus características principales se pueden encontrar las siguientes:

- Segmentos de memoria no volátil de alta resistencia.
- Un temporizador/contador de 16 bits con pre-escalador independiente con modo de comparación y modo de captura.
- Dos temporizadores/contadores de 8 bits con pre-escalador separado, modo de comparación.
- Seis modos de reposo: Inactividad, reducción de ruido ADC, ahorro de energía, apagado, espera y espera prolongada.

Y los parámetros para tener en cuenta son:

Tabla 1 Parámetros importantes ATmega328P.

Tipo de memoria programable	Flash
Tamaño de memoria programable	32 bytes
Velocidad CPU (MIPS/DMIPS)	20
SRAM	2048 KB
Datos EEPROM/HEF	1024 bytes
Contadores	2 x 8-bit, 1x 16 bit
Comparadores	1
Rango de temperatura	-40 a 85 °C
Rango de voltaje operativo	1.8 a 5.5 V
Numero de pines	32

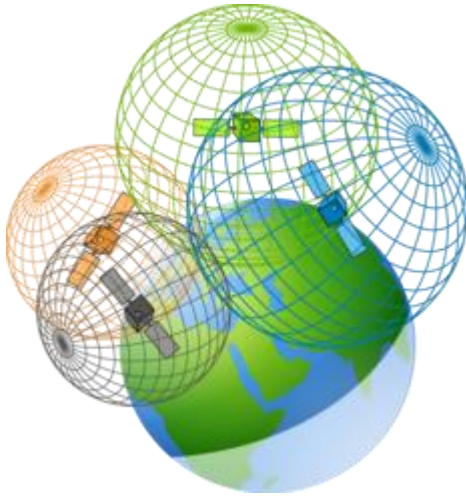
## 5.6 GPS

El sistema de posicionamiento global o GPS es un sistema GNSS, que permite determinar en toda la tierra la posición de cualquier objeto (ej.: una persona o un vehículo) con una precisión de entre 5 y 10 metros para un GPS convencional.

El sistema fue desarrollado, instalado y empleado por el departamento de defensa de estados unidos. Para determinar la posición satelital en cualquier parte de la tierra, el sistema GPS utiliza la trilateración la cual consiste en la medición de la distancia a los satélites para de esta manera determinar la posición del objeto que tiene el GPS en la tierra [33].

Para entender un poco más el funcionamiento de un GPS, es importante saber lo siguiente:

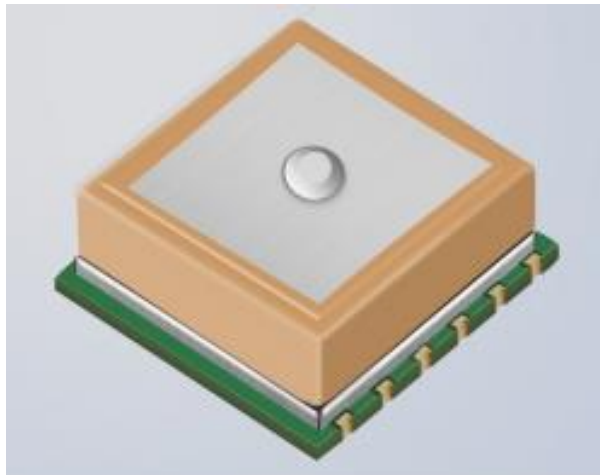
- Un receptor GPS recibe señales en onda de radio provenientes de 4 o más satélites (Figura 11).
- Cada señal contiene información acerca del momento en que fue transmitida desde el satélite y la posición en la que se encontraba dicho satélite al momento de enviar la señal.
- Como toda onda electromagnética, la señal de radio para los GPS viaja a la velocidad de la luz (299.792,458 kilómetros por segundo).
- El receptor calcula su posición comparando las distancias en línea recta hasta cada uno de los satélites, teniendo en cuenta que se toma la posición del satélite al momento de enviar la señal.



*Figura 11 Ilustración de satélites visibles por un GPS utilizando trilateración [34].*

El GPS permite saber dónde se encuentra un objeto en el instante que se quiera, pero también permite calcular rutas y distancias entre varios objetos.

Para poder utilizar esta tecnología se debe disponer de un dispositivo GPS, por ejemplo, el que se muestra en la (Figura 12).



*Figura 12 Dispositivo GPS L-80 [35].*

## 5.7 RFM95

Este tipo de transceptores cuentan con el módem de largo alcance LoRa que le permite tener una comunicación con uno o más gateway LoRa, teniendo un envío de datos rápido y con alta inmunidad a las interferencias con un bajo consumo de corriente.

Utilizando la técnica de modulación LoRa, el transceptor RFM95 puede alcanzar una sensibilidad de más de -148dBm utilizando un cristal de bajo coste y una lista de materiales ideales. La alta sensibilidad combinada con el amplificador de potencia integrado de +20 dBm, permite un presupuesto de enlace líder en la industria lo que lo hace óptimo para cualquier aplicación que requiera alcance o robustez [36].

### 5.7.1 Estructura de envío de paquetes LoRa

Los paquetes de envío LoRa contienen tres elementos principales:

- **Un preámbulo:** Esté, se utiliza para sincronizar el gateway con el flujo de datos entrante. Por defecto el paquete está configurado con una secuencia de 12 caracteres de longitud, pero al tratarse de una variable programable, el preámbulo puede ampliarse.
- **Una Cabecera (Opcional):** Esté, permite configurar la transmisión de los datos.
- **Los datos:** La información que se quiere enviar.

Para el envío de datos se deben tener en cuenta los siguientes tres parámetros: el primero, el ancho de banda o bandwidth, ya que da la posibilidad de tener un rango de transmisión más amplio, pero se debe elegir un ancho de banda más pequeño sacrificando velocidad de transferencia. El segundo, el factor de difusión o spreading factor que además de dar un rango de transmisión más alto, ayuda a tener mejoras de procesamiento y codificación de datos, y por último el tercer parámetro hace referencia

a la tasa de codificación o coding rate el cual, describe la relación entre los datos reales y los datos de corrección de errores.

Un envío de datos por LoRa, emplea dos tipos de formato de envío paquetes, explícito e implícito, mediante los cuales realiza el envío de los datos.

**Modo Explícito de envío:** Este es el modo de operación por defecto, en este caso, la cabecera proporciona información sobre la carga útil (datos).

- La longitud de la carga útil en bytes.
- Porcentaje de corrección de errores.
- La presencia de un CRC opcional de 16 bits para la carga útil (datos).

La cabecera se transmite con un código de corrección de errores máximo (4/8). También tiene su propio CRC para que el receptor pueda descartar las cabeceras no válidas.

**Modo Implícito de envío:** Se presenta, cuando se tiene un valor fijo para la carga útil, el CRC y la tasa de codificación o por lo menos son valores conocidos, la ventaja que se tiene sobre el modo explícito es que la transmisión se puede hacer más rápido. A diferencia del modo explícito en este, se elimina la cabecera de paquete. Por otro lado, los valores de configuración para la carga útil, el CRC y la tasa de codificación de errores deben colocarse a los dos lados del radioenlace de manera manual [36].

## 6 DISEÑO METODOLÓGICO

A continuación, se presenta el diseño metodológico para el desarrollo del proyecto, teniendo en cuenta las etapas que se establecen en la Figura 13 y donde se evidencia la organización de estas para el cumplimiento de los objetivos planteados en el presente documento.

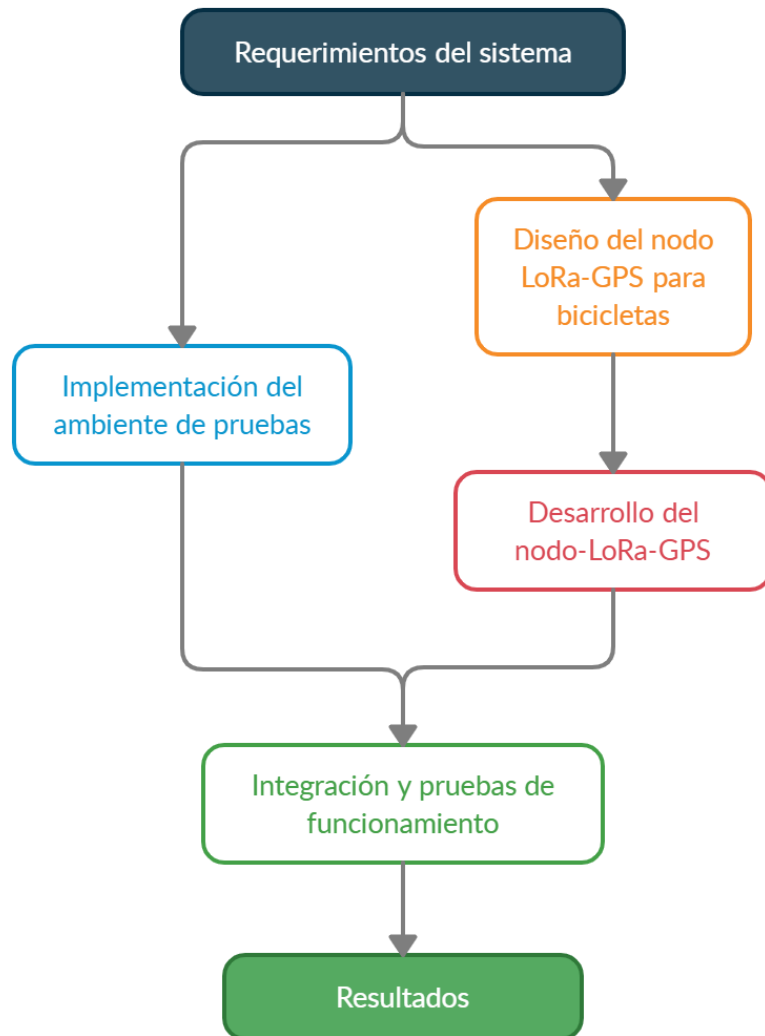


Figura 13 Diagrama del diseño metodológico. [Autor].

- **Requerimientos de sistema:** Se buscan y analizan los requerimientos técnicos y funcionales para el correcto funcionamiento del ambiente de pruebas y diseño del dispositivo nodo-LoRa.
- **Implementación del ambiente de pruebas:** Se construye un ambiente de pruebas básico, basado en una red LoRaWAN con un nodo GPS comercial (Shield de Arduino LoRa-GPS), una puerta de enlace (LG01-P IoT gateway), un servidor de red (ThingSpeak) y una aplicación web. Con los anteriores dispositivos, se puede enviar datos de geolocalización de nodos LoRa-GPS, comprobar el correcto funcionamiento de la red LoRaWAN y visualizar los datos al usuario final.
- **Diseño del nodo propio:** Partiendo de los aspectos técnicos y funcionales encontrados anteriormente en los requerimientos del sistema, se diseña un nodo LoRa-GPS que se adapte a una bicicleta.
- **Desarrollo del nodo propio:** Se implementa y desarrolla un nodo LoRa-GPS, para un consumidor final, basado en el diseño para un consumidor final.
- **Integración y pruebas de funcionamiento:** Se integra al ambiente de pruebas, el desarrollo del nodo propio (Lora-GPS) y se comprueba el correcto funcionamiento dentro de la red LoRaWan. Posteriormente, se adapta el nodo LoRa-GPS a una bicicleta para poder hacer un seguimiento de su localización y los aspectos funcionales.
- **Resultados:** Se integra el nodo LoRa-GPS a una bicicleta, donde se recopila información que evidencie su correcto funcionamiento, tomando datos en campo para visualizar los resultados en la aplicación web, además de medir la autonomía, distancia y veracidad de los datos obtenidos.



## 7 DESARROLLO

### 7.1 Requerimientos del sistema

Para el desarrollo de un dispositivo o producto final, es necesario tener en cuenta los requerimientos técnicos y funcionales que permiten obtener los lineamientos de diseño electrónico y físico básicos además de una idea general para completar su desarrollo a cabalidad.

#### 7.1.1 Requerimientos Técnicos

A continuación, se describen los requerimientos técnicos para el ambiente de pruebas y el diseño del Nodo LoRa-GPS, que adapta características y requerimientos de un sistema de IoT con LoRaWAN.

- **Tecnología LoRaWAN:** La tecnología LoRaWAN es la tecnología sobre la cual va a estar estructurado el dispositivo de rastreo. Esta tecnología se escogió por las ventajas y características mencionadas en la sección 5.1 del marco teórico.

Para la implementación del proyecto se debe tener en cuenta:

- La banda por usar para el ambiente de pruebas debe estar entre (Gateway y nodo comercial) y el dispositivo a diseñar debe ser de 914.9MHz.
  - Se debe implementar una topología estrella en el ambiente de pruebas ya que solo se tendrán dos nodos conectados a la red.
  - Implementar seguridad a la comunicación mediante el cifrado de datos por AES.
  - El dispositivo por diseñar debe ser de clase A, para que la comunicación se realice con la potencia más baja, para una mayor autonomía.
- 
- **Precisión:** La precisión (o error de localización) suele ser uno de los requisitos más importantes en este tipo de sistemas. Por lo general, se toma como métrica

de rendimiento el error de distancia media, que está dado por la diferencia entre la posición estimada y la real. Cuanto mayor es la precisión, se puede afirmar que el sistema de localización es mejor.

Comparando diferentes tecnologías de localización con los sistemas de posicionamiento global (GPS), se puede encontrar una diferencia importante en la precisión de cada una, en primer lugar, las tecnologías de localización como GSM, Cell-Id, AOA, TOA, TDOA o RSS, su precisión es mayor en zonas urbanas, donde llega ser menor de 50 metros, en zonas rurales puede aumentar, por otro lado, la precisión con sistemas de posicionamiento global (GPS) llega a ser menor de 15 metros y su precisión no se ve afectada en zonas rurales [37].

Ahora bien, si se quiere tener una precisión que permita encontrar una bicicleta cuyo tamaño es de máximo 2 metros, se necesita un dispositivo de rastreo que brinde un margen de precisión acorde al objeto a rastrear. Por esta razón, la tecnología de localización a utilizar es un sistema de posicionamiento global y para ver cuál de estas tecnologías se acoplaba mejor al proyecto se plasmó en la Tabla 2, las principales tecnologías de geolocalización y sus características:

Tabla 2 Características de las tecnologías de geolocalización [38].

	<b>GPS</b>	<b>GLONASS</b>	<b>GALILEO</b>	<b>BeiDou/ COMPASS</b>
<b>Fuente</b>	24 satélites	24 satélites	30 satélites	30 satélites
<b>Precisión</b>	7.8 m Público (SPS) 5.4 m Militar (PPS)	7.4 m Público (SP) 4.5 m Militar (HP)	1 m Público 0.01 m Avanzado	10 m Público 0.1 Militar
<b>Cobertura</b>	Global	Global	Global	China para BeiDou-1 Global para BeiDou-2/COMPASS
<b>Planos Orbitales</b>	6 planos	3 planos	3 planos	3 planos
<b>Satélites por plano</b>	4 satélites	8 satélites	10 satélites	9 satélites
<b>Altura órbita</b>	26650 km	19100 km	23222 km	21150 km

Teniendo en cuenta las características mencionadas en la Tabla 2, el tamaño de una bicicleta promedio y el presupuesto previsto para el desarrollo del proyecto, se opta por utilizar la tecnología GPS ya que brinda una cobertura de geolocalización global, tiene una precisión aceptable para la implementación, es uno de los dispositivos más comerciales.

- **Autonomía:** El consumo total del Nodo-LoRa depende del consumo de los componentes electrónicos que sean necesarios para su correcto funcionamiento generando a su vez una mayor autonomía para el usuario, si el consumo total es bajo.

En Bogotá el tiempo de desplazamiento promedio por viaje en bicicleta es de 39 minutos, según la encuesta de movilidad de 2019, además un poco menos de la mitad (48,2%) de estos viajes son realizados por motivos obligatorios de trabajo y estudio [1], tomando en cuenta que el horario laboral establecido en

Colombia es de máximo 8 horas diarias según el Artículo 161 del código sustantivo del trabajo, en un día laboral, una bicicleta está en la calle aproximadamente 9,5hr diarias, por lo cual, puede ocurrir un robo en cualquier momento del trayecto; suponiendo que se realice al final de la jornada el usuario tiene que armar un plan de búsqueda desde este instante, contando con el tiempo muerto de buscar a las autoridades competentes para que la integridad del usuario no sea afectada y del tiempo que tome la recuperación.

Tomando en cuenta lo anterior, el dispositivo debería cumplir las siguientes especificaciones:

- El consumo total para el nodo Lora-GPS a diseñar debe ser bajo, para lograr una mayor autonomía, para ello la mayoría de los componentes que lo conformen deben ser de baja potencia.
- Contando con el tiempo de búsqueda, el dispositivo deberá tener una autonomía de batería mínima de 14 horas, para lograr el objetivo de aumentar la posibilidad de recuperar la bicicleta.
- **Ubicación:** El nodo a diseñar, es un dispositivo de rastreo y su ubicación dentro de la bicicleta es un factor importante por razones de seguridad, por un lado, para que no se identifique con facilidad que hay un dispositivo de rastreo y, por otro lado, si se logra identificar, que su desactivación o remoción no sea convencional, para aumentar el tiempo de búsqueda al biciusuario.

La bicicleta tiene diferentes tamaños y formas, para las cuales buscar una general sería ideal, tomando en cuenta que la estructura de la bicicleta no se vea afectada y que pueda causar accidentes, daños físicos o daños visuales a la bicicleta. La empresa Skybike GPS Traker contempla ubicaciones para sus dispositivos dentro de faroles, marco o debajo del sillín; Velocate, una empresa alemana, mimetiza su rastreador GPS en la luz trasera de la bicicleta y la empresa Sherlock que oculta sus rastreadores en el manubrio de la bicicleta.

El nodo Lora-GPS deberá ubicarse en un lugar estratégico, para el cual cumpla con algunas especificaciones, como:

- La posición del dispositivo: tendrá que ser estructuralmente fuerte y estar bien sujetado.

- El tamaño y forma del dispositivo: se tendrá que adaptar a la ubicación escogida.
- El dispositivo deberá optar por pasar lo más desapercibido posible o por lo menos no demostrar ser un sistema de rastreo.

### 7.1.2 Requerimientos Funcionales

A continuación, se describen los requerimientos funcionales para el diseño del Nodo LoRa-GPS, que se relaciona con la experiencia de usuario para cumplir a cabalidad el objetivo para el que es diseñado.

En un producto se puede observar distintos factores funcionales y conceptos fundamentales, que tienen relación con diseño del proyecto al usuario final [39]:

- **Usabilidad e Interacción:** Es la facilidad de uso, que el biciusuario encuentre del sistema, teniendo en cuenta el diseño, la satisfacción y la facilidad de aprendizaje o adaptación para su correcto uso y su interacción. El nodo LoRa-GPS al ser un producto para consumidor final, tiene tres interacciones con el usuario: la instalación del dispositivo por parte de usuario, la recarga de la batería y por último el uso de la aplicación, donde se visualiza los datos proporcionados por el dispositivo.

Tomando en cuenta la interacción que el usuario tiene con el sistema, se tiene en cuenta los siguientes aspectos de usabilidad:

- La instalación del dispositivo debe ser sencilla para el usuario, sin requerir herramientas difíciles de conseguir.
- El cambio o recarga de la batería debe ser de fácil acceso para el usuario y deberá contar con las herramientas necesarias para realizarlo, sin requerir el transporte de la bicicleta a un punto de carga o esfuerzos adicionales.
- La aplicación pedirá el id del dispositivo y proporciona latitud, longitud, fecha y hora de cada uno de los datos de geolocalización, para tener un conocimiento aproximado de la ubicación de la bicicleta y su recorrido.

- **Accesibilidad:** Es un atributo que se refiere a la posibilidad de usar un producto sin tanta dificultad, por el mayor número de usuarios posibles, independientemente de las limitaciones (contexto o usuario). Sin embargo, no significa que el diseño deba ser universal y para todo el mundo, ya que se cuentan con unos requerimientos mínimos que el usuario debe contemplar:
  - Tener conocimientos mínimos de aplicaciones web.
  - Contar con un dispositivo celular o computadora con acceso a internet, para visualizar los datos de geolocalización en tiempo real.
  
- **Relación Esfuerzo-Beneficio:** Aunque la usabilidad genera una satisfacción al usuario, no es realmente hasta que se consigue un beneficio en donde encuentra una motivación para completar las tareas en las que se interactúa, y por esto cada interacción debería generar un beneficio, para que el usuario realice los esfuerzos necesarios, en el caso del nodo Lora-GPS son los siguientes:
  - La instalación se realiza una sola vez en la bicicleta, ya que el dispositivo no se mueve o instala en cada uso.
  - Al estar fijo el dispositivo, la recarga de la batería es más complicada si se realiza directamente en la bicicleta, ya que su transporte al punto de carga genera un esfuerzo grande, por este motivo la batería debería ser removible para su carga. Además, si el usuario tiene varias baterías puede intercambiarlas sin necesidad de esperar a la carga completa para seguir utilizando el sistema.
  - Una aplicación web puede generar un esfuerzo menor, ya que los datos pueden revisarse desde un celular, tablet o computadora, sin tener problemas de compatibilidad o accesibilidad.
  - El envío de datos de geolocalización constantes sin necesidad de activar el dispositivo, genera mayor seguridad y menos esfuerzo al usuario, ya que no tendrá que estar pendiente si encendió el dispositivo o si está funcionando correctamente.

## 7.2 Ambiente de pruebas

Con la finalidad de corroborar el correcto funcionamiento del nodo comercial y el nodo desarrollado, se implementó un ambiente sobre el cual se puedan tomar métricas y parámetros de funcionamiento.

Este ambiente se realizó en dos etapas: contemplando la arquitectura de red LoRa y la configuración de ambiente de pruebas.

### 7.2.1 Arquitectura de red LoRaWAN

La arquitectura de una red LoRaWAN, se ajusta a la Figura 14, y es la implementada en el desarrollo del proyecto.

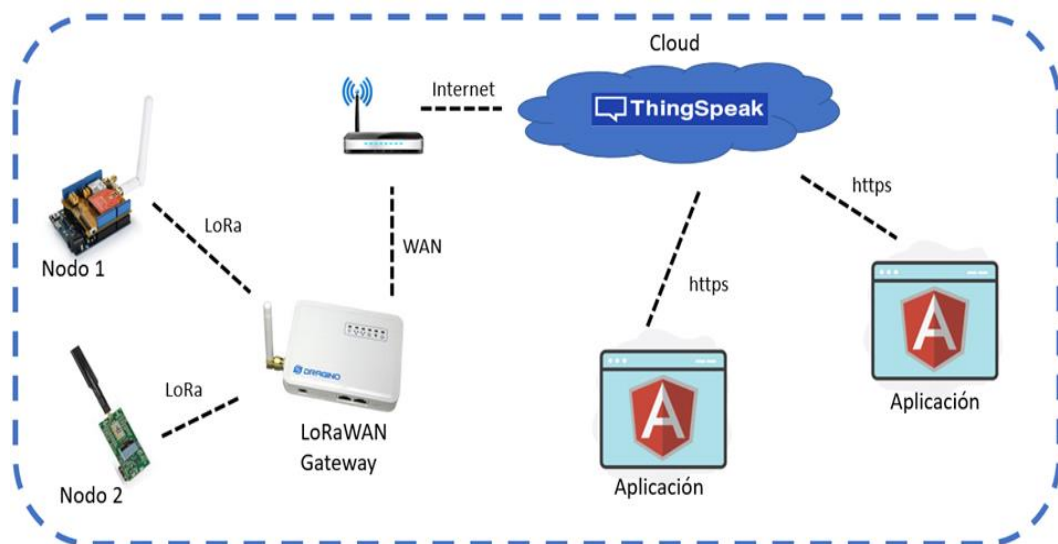


Figura 14 Arquitectura implementada en la solución [Autor].

Esta arquitectura implementada, está compuesta por:

- Nodos: Se tienen dos nodos, uno comercial y el nodo que se desarrolló con el fin de adquirir las componentes de latitud y longitud que unidas brindan la posición satelital de la bicicleta.

- Gateway: Para este desarrollo, la configuración del gateway se realizó en dos etapas: la configuración de red y la programación de este para que recibiera datos de los nodos y los envié al servidor elegido (ThingSpeak).
- Servidor: Dentro de los diferentes servidores gratuitos que se encuentran en internet, se decidió utilizar ThingSpeak ya que ofrece un margen alto de almacenamiento de datos, pero a su vez, da la facilidad para consumir estos datos por medio de diferentes API's.
- Aplicación: Con el fin de mostrar de una manera más agradable los datos de posición satelital de la bicicleta, se desarrolló una aplicación en el framework Angular.

## **7.2.2 Configuración del ambiente de pruebas**

Para poder utilizar de manera satisfactoria la arquitectura que se muestra en la Figura 14 se deben preparar y programar tanto el gateway como el nodo, para esto se llevaron cabo los siguientes pasos:

### **1. Verificar la versión del Firmware del gateway**

Para el correcto funcionamiento del gateway, el proveedor recomienda mantenerlo actualizado a la versión más reciente del firmware y en el momento del desarrollo del proyecto se tenía instalada la versión 4.3.7, la cual salió al público como versión estable en septiembre 2019.

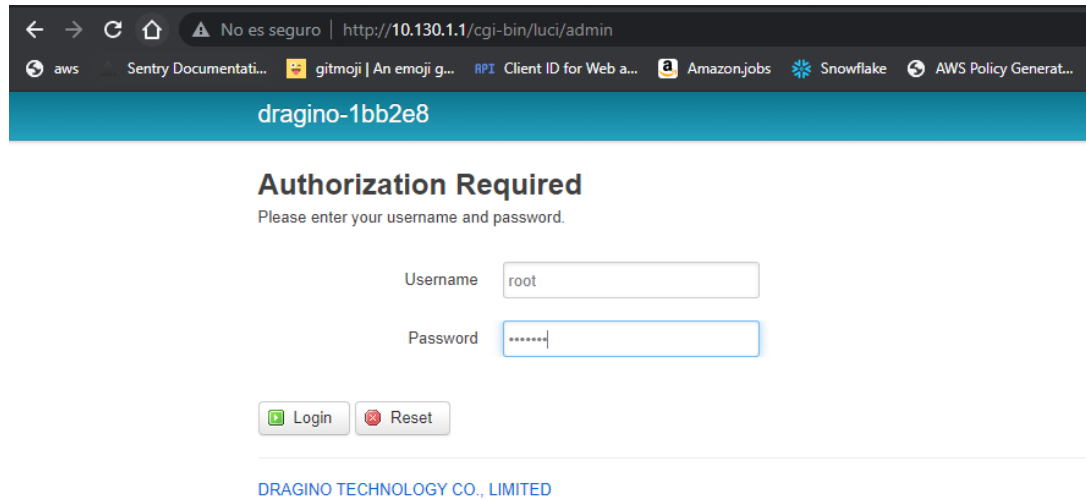
### **2. Configuración del entorno de red del gateway**

El proceso de configuración del nodo para su correcto funcionamiento es el siguiente:

- Inicialmente, se accede a la red wifi que genera el gateway y de esta manera se tiene acceso al servidor y sus configuraciones tal cual como se observa en la Figura 15.



Este servidor se encuentra en la dirección IP 10.130.1.1 la cual se configura previamente para que sea una IP fija.



*Figura 15 Acceso al servidor del gateway [Autor].*

- Al realizar el proceso de inicio de sesión al servidor de configuraciones con las credenciales de acceso, se tuvo el control total de la adecuación, las métricas y también los registros del estado del gateway Figura 16.

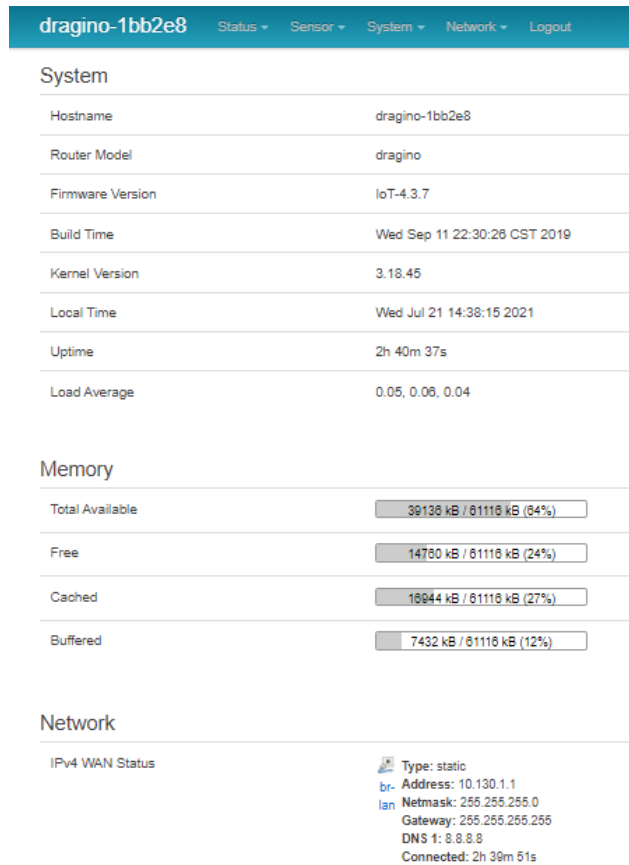


Figura 16 Panel de control inicial del servidor del gateway [Autor].

- Para que el gateway pudiera cumplir a cabalidad su función de recibir los datos y posteriormente enviarlos al servidor, una configuración indispensable es conectar el gateway a una red de internet (LAN/WAN) y en este caso se usó la red WIFI provista por un punto de acceso de un celular y esta configuración se ve reflejada en la Figura 17.

## Small Enterprise-Campus Network

### Internet Access


Access Internet Via	WiFi Client ▾
SSID	iPhone de Smith
Encryption	WPA2 ▾
Password	..... 
Way to Get IP	DHCP ▾
<u>Display Net Connection</u>	Domain or IP
	<input checked="" type="checkbox"/> Continusely Check Net Connection

Figura 17 Configuración de conexión a red de internet [Autor].

- Al tener una implementación de una red privada, es importante también configurar la opción de cortafuegos que trae el gateway con el fin de darle un mayor grado de seguridad a la implementación, y para lograr esto es necesario configurarlo como se ilustra en la Figura 18.

dragino-1bb2e8    Status ▾    Sensor ▾    System ▾    Network ▾    Logout

General Settings    Port Forwards    Traffic Rules    Custom Rules

### Firewall - Zone Settings

The firewall creates zones over your network interfaces to control network traffic flow.

#### General Settings

Enable SYN-flood protection

Drop invalid packets

Input: accept ▾

Output: accept ▾

Forward: reject ▾

#### Zones

Zone ⇒ Forwardings	Input	Output	Forward	Masquerading	MSS clamping	
lan: lan: ⇒ wan	accept ▾	accept ▾	reject ▾	<input type="checkbox"/>	<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Delete</a>
wan: wan: ⇒ REJECT	accept ▾	accept ▾	reject ▾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a> <a href="#">Delete</a>

[Add](#)

[Save & Apply](#) [Save](#) [Reset](#)

DRAGINO TECHNOLOGY CO., LIMITED

Figura 18 Configuración de Firewall dentro del gateway [Autor]

Así, se cumplen todos los pasos de configuración necesarios para el correcto funcionamiento del gateway en la implementación del ambiente de pruebas y posteriormente para un ambiente productivo.

### 3. Programación del gateway

Para realizar este proceso se usó el IDE de Arduino para programar las funciones específicas de enviar los datos al servidor de la nube contemplando las especificaciones técnicas que requiere la red LoRaWAN. El gateway se programa por medio OTA, programando el equipo sin ningún tipo de cable, en su lugar se utiliza una red Wifi, empleando un puerto de red como se puede observar en la Figura 19:

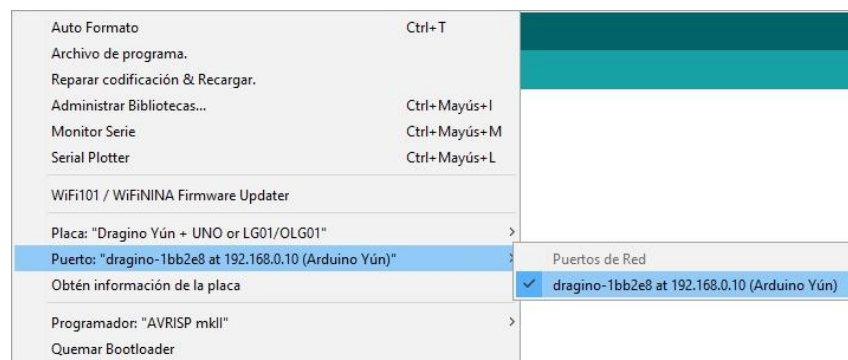


Figura 19 IDE de Arduino para configuración del puerto de programación del Gateway [Autor]

En la Figura 20, se muestra el diagrama de flujo que describe la programación del gateway (se toma en cuenta que el gateway debe estar conectado a la red Wifi para su correcto funcionamiento y envió satisfactorio al servidor de red). Además, los datos que recibe el gateway vienen cifrados, por lo tanto, no tienen visibilidad para el operador de red, ni para otra red LoRaWAN.

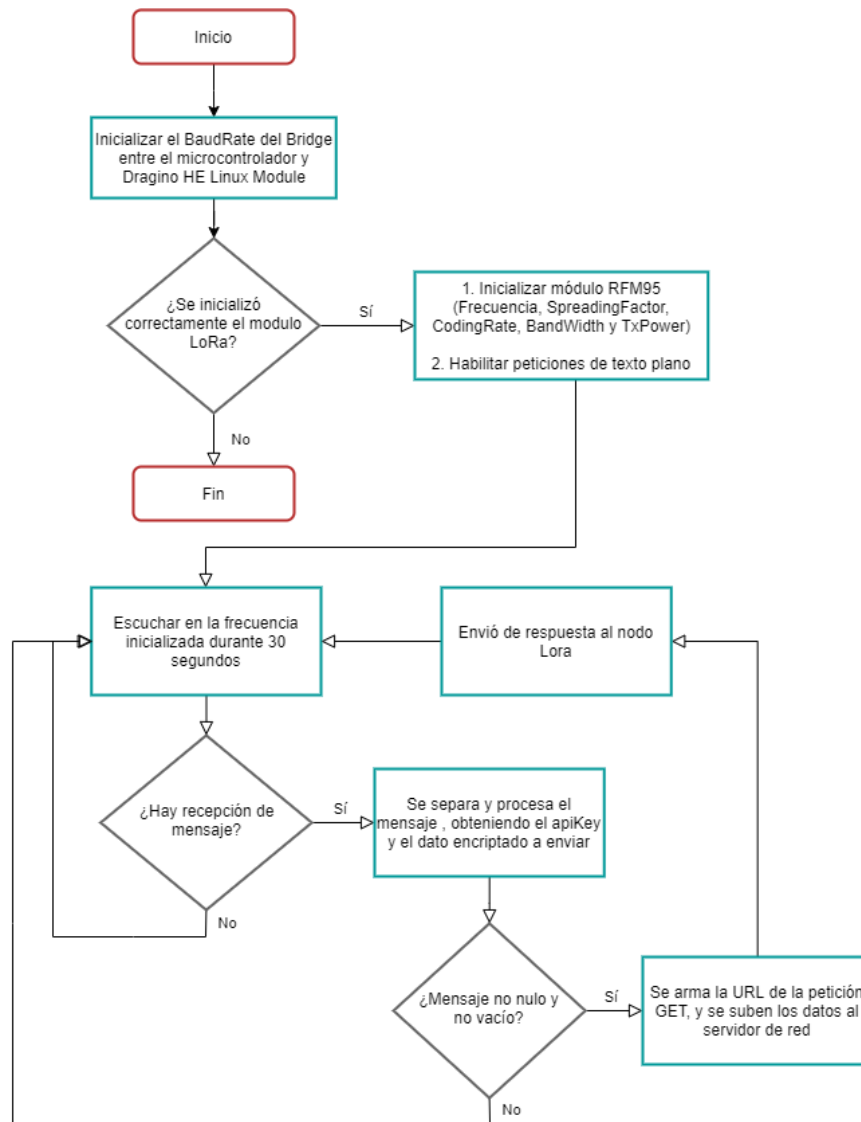


Figura 20 Diagrama de flujo del algoritmo del gateway [Autor]

#### 4. Programación del nodo LoRa-GPS

Para realizar este proceso al igual que con el gateway, se usó el IDE de Arduino para programar las funciones específicas de obtener y cifrar los datos de geolocalización por medio del módulo GPS y enviarlos junto a la clave de sesión de red (apiKey) pre configurada al subir el programa ya que se activan por medio de ABP,

además del modo suspensión del Arduino y el módulo RFM95 para bajar el consumo total. En la Figura 21, se muestra el diagrama de flujo que describe la programación del Nodo LoRa-GPS y el funcionamiento que tiene dentro del sistema.

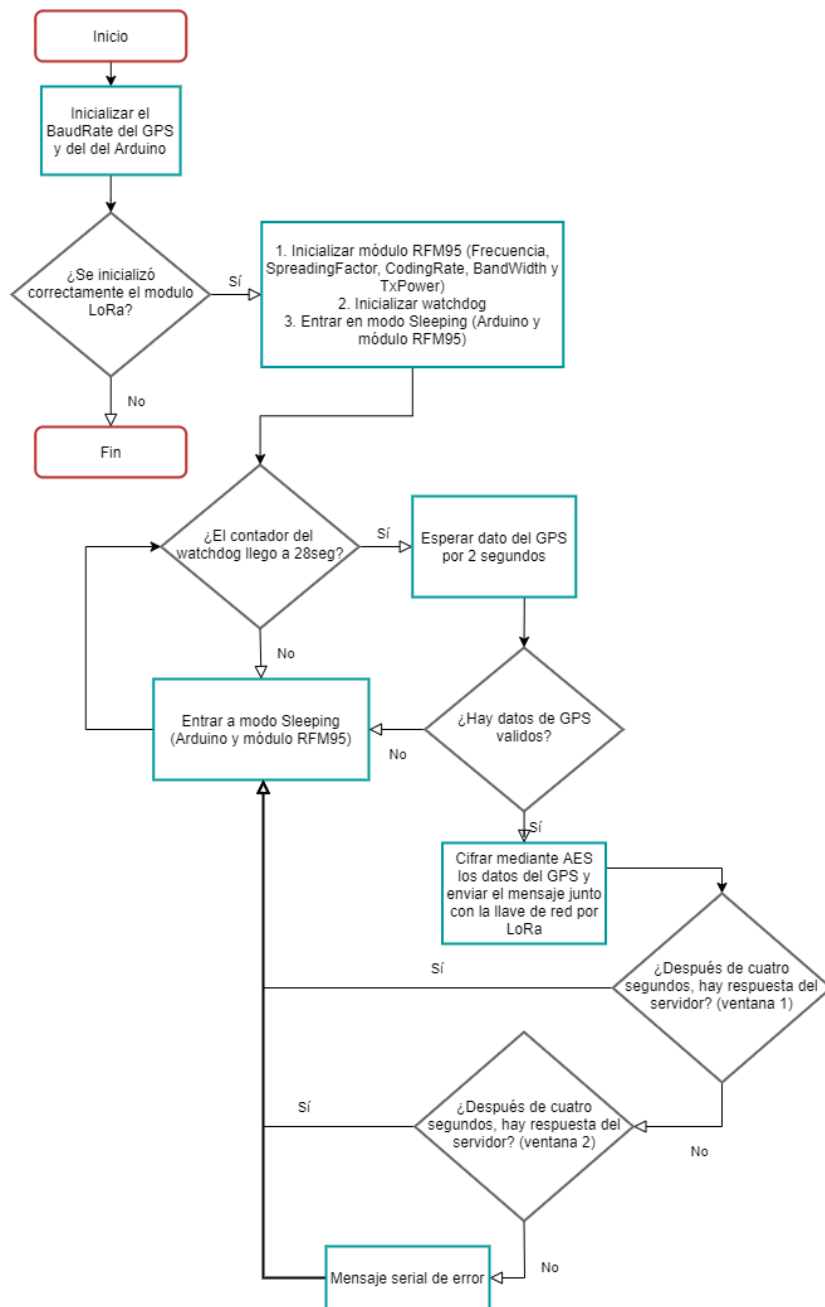


Figura 21 Diagrama de flujo del algoritmo del nodo LoRa-GPS [Autor]

## 5. Registrar el gateway en el servidor

Ya funcionando el gateway y el nodo en conjunto en el ambiente de pruebas, el paso a seguir es almacenar los datos recibidos en un servidor, para lo cual se eligió el servidor ThingSpeak y en el momento de integrarlo a la solución fue necesario:

- El registro gratuito con la cuenta de MathWorks, este servidor brinda la posibilidad de recibir tres millones de mensajes al año en cuatro diferentes canales, en la Figura 22 se visualiza la página de registro de MathWorks.

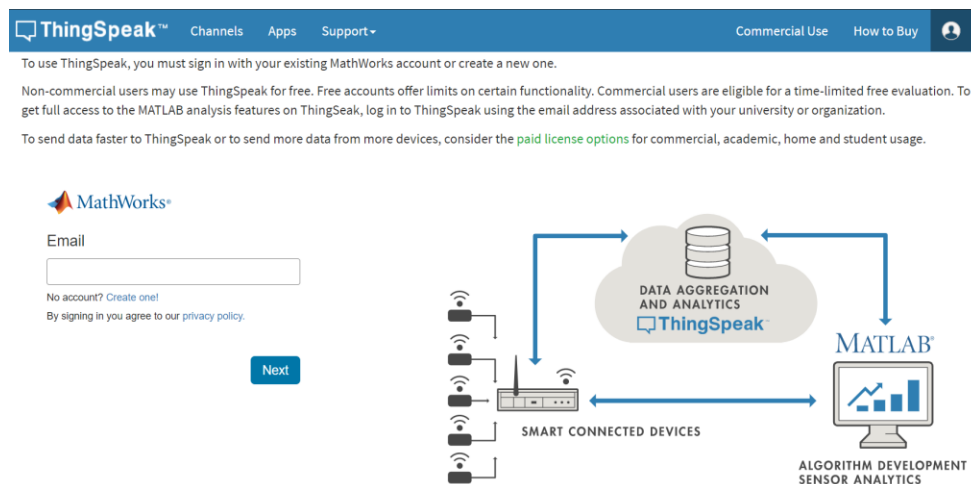


Figura 22 Página de registro para el servidor ThingSpeak [Autor]

- Una vez se tiene acceso al servidor, el paso siguiente es crear un canal en el cual se almacenan los datos enviados desde el gateway por medio de un Write API-Key que brinda ThingSpeak, la creación y configuración de este canal se realizó como se ve en la Figura 23:



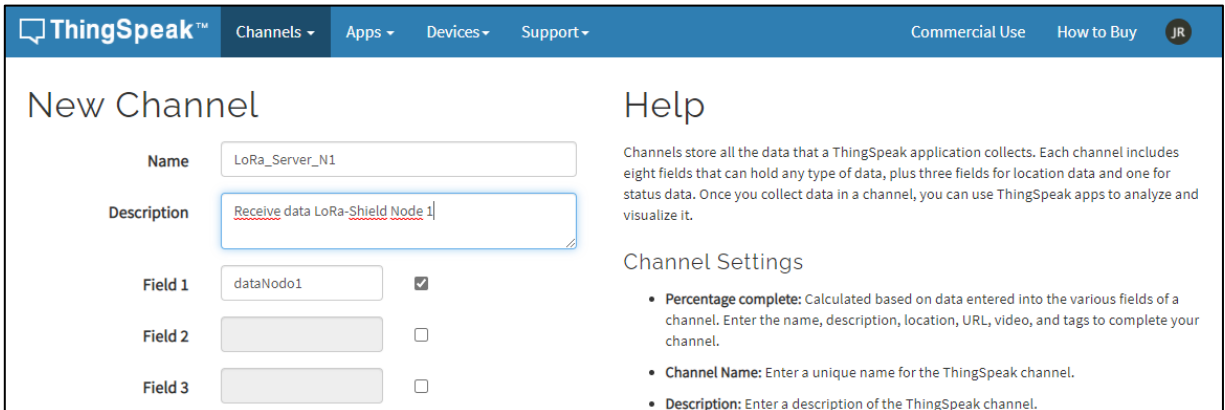


Figura 23 Configuración del canal para recibir datos [Autor]

- Una vez creado el canal, el campo nombrado como “dataNodo1” corresponde a las variables que se usan en la geolocalización (latitud y longitud) pero cifrado, se generan también las API-keys y API Requests como se muestra Figura 24.

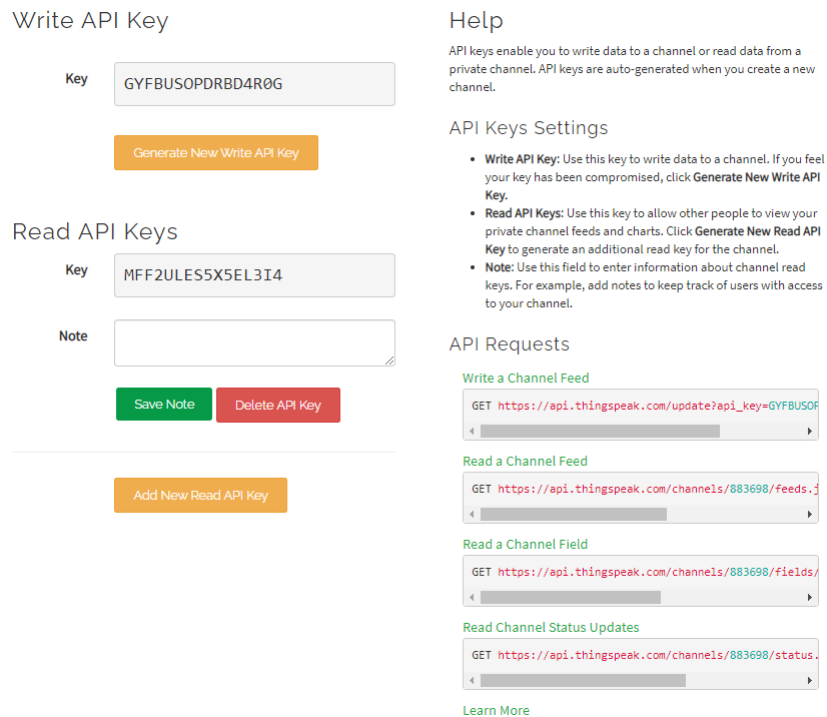
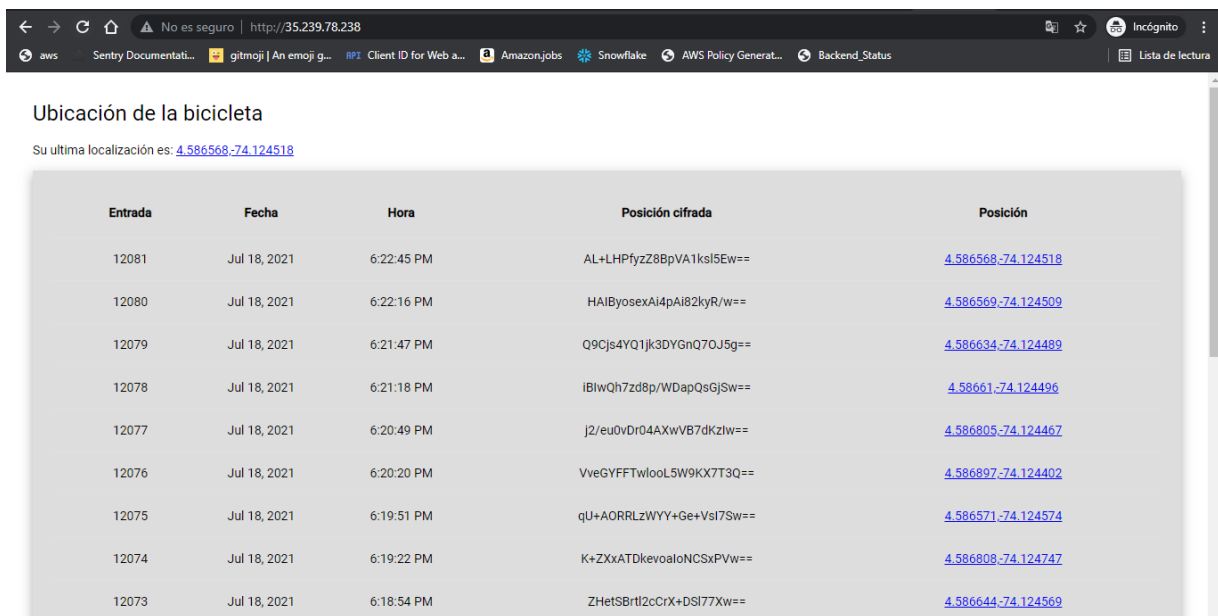


Figura 24 Dashboard de API Keys para el canal [Autor]

## 6. Aplicación Web

Con el objetivo de mostrar al usuario final los datos adquiridos por el nodo LoRa, se desarrolló una aplicación en el framework Angular como se muestra en la Figura 25, con el fin de consultar los datos del servidor y extraerlos mediante un API REST que brinda el servidor de ThingSpeak.



Ubicación de la bicicleta

Su última localización es: [4.586568-74.124518](#)

Entrada	Fecha	Hora	Posición cifrada	Posición
12081	Jul 18, 2021	6:22:45 PM	AL+LHPfyzZ8BpVA1ksl5Ew==	<a href="#">4.586568-74.124518</a>
12080	Jul 18, 2021	6:22:16 PM	HAIByosexAi4pAi82kyR/w==	<a href="#">4.586569-74.124509</a>
12079	Jul 18, 2021	6:21:47 PM	Q9Cjs4YQ1jk3DYGnQ70J5g==	<a href="#">4.586634-74.124489</a>
12078	Jul 18, 2021	6:21:18 PM	IBlWqh7zd8p/WDapQsGjSw==	<a href="#">4.58661-74.124496</a>
12077	Jul 18, 2021	6:20:49 PM	j2/eu0vDr04AXwVB7dKzlw==	<a href="#">4.586805-74.124467</a>
12076	Jul 18, 2021	6:20:20 PM	VveGYFFTWl0oL5W9KX7T3Q==	<a href="#">4.586897-74.124402</a>
12075	Jul 18, 2021	6:19:51 PM	qU+AORRLzWYy+Ge+Vsl7Sw==	<a href="#">4.586571-74.124574</a>
12074	Jul 18, 2021	6:19:22 PM	K+ZXxATDkevoaloNCSxPVw==	<a href="#">4.586808-74.124747</a>
12073	Jul 18, 2021	6:18:54 PM	ZHetSBrtI2cCrX+DSI77Xw==	<a href="#">4.586644-74.124569</a>

Figura 25 Visualización de los datos desde la aplicación web [Autor]

En la Figura 25 se muestra la tabla de los últimos datos tomados por el nodo LoRa-GPS, en donde se encuentra la fecha, hora, posición cifrada y la posición descifrada. El dato de la posición tiene un hipervínculo a Google Maps para mostrarle al usuario la posición en el mapa con se muestra en la Figura 26.

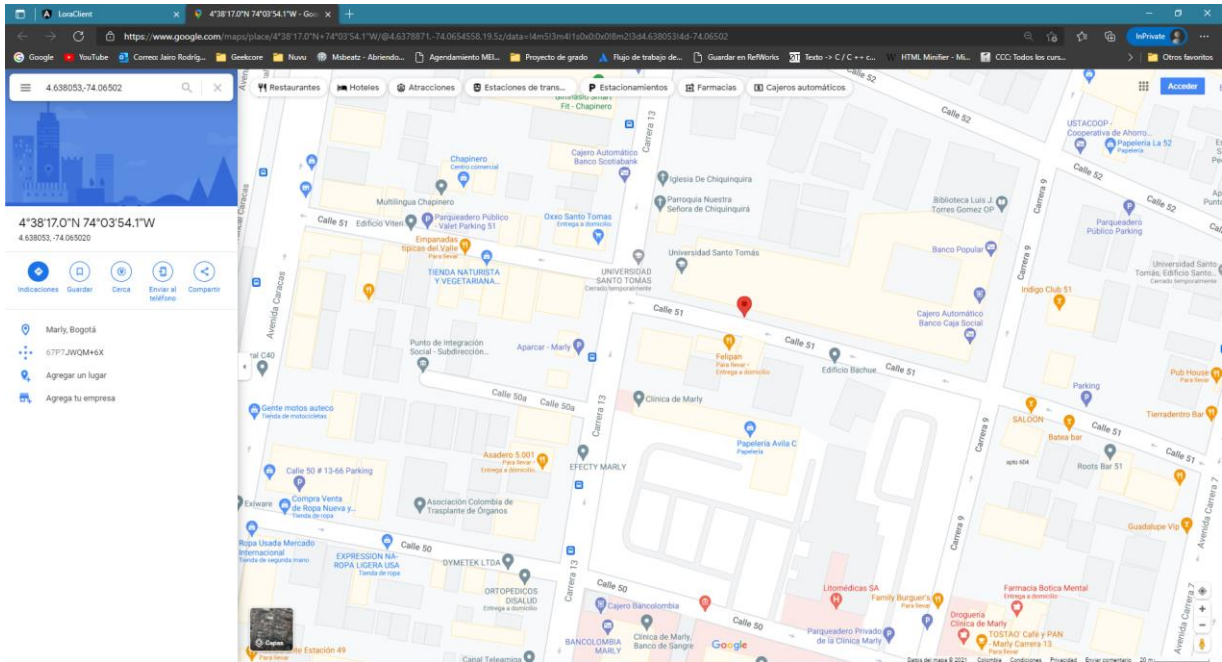


Figura 26 Re-direccionamiento a la interfaz de Google Maps, con el dato de la posición seleccionado en la aplicación web [Autor].

Adicional a esto, y con el objetivo que el usuario final pudiera consultar los datos en cualquier momento, se contenerizó la aplicación y se desplegó en un clúster de kubernetes en la nube de Google (GCP).

Los pasos para contenerizar y desplegar la aplicación fueron los siguientes:

- Se debe clonar el repositorio o estar en la carpeta raíz del proyecto. En la Figura 27 se pueden ver los archivos principales del desarrollo de la aplicación y las instrucciones para la construcción de la imagen Docker para posteriormente desplegarla.

```
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$ ls
Dockerfile      e2e              package.json     tsconfig.json
README.md       karma.conf.js    src              tsconfig.spec.json
angular.json     package-lock.json tsconfig.app.json tslint.json
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$
```

Figura 27 Listado de archivos fuente de la aplicación [Autor].

- Para la construcción de la imagen Docker, se debe hacer mediante instrucciones en un Dockerfile es cual debe contener la sintaxis que se ve en la Figura 28.

```
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$ cat Dockerfile
#Primera Etapa
FROM node:10-alpine as build-step

RUN mkdir -p /app
WORKDIR /app
COPY package.json /app
RUN npm install
COPY . /app
RUN npm run build --prod

#Segunda Etapa
FROM nginx:1.17.1-alpine

COPY --from=build-step /app/dist/LoraClient /usr/share/nginx/html
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$
```

Figura 28 Estructura Dockerfile para la construcción de imagen Docker [Autor].

- Teniendo construido el Dockerfile, se debe correr el comando de consola para crear la imagen Docker. El comando y el proceso de la creación de la imagen Docker se pueden ver en la Figura 29.

```
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$ sudo docker build -t lora-web-app .
Sending build context to Docker daemon 838.1kB
Step 1/9 : FROM node:10-alpine as build-step
--> aa67ba258e18
Step 2/9 : RUN mkdir -p /app
--> Using cache
--> 465d75274196
Step 3/9 : WORKDIR /app
--> Using cache
--> ec365700cae4
Step 4/9 : COPY package.json /app
--> Using cache
--> 2b24b2f721a8
Step 5/9 : RUN npm install
--> Using cache
--> d10de247b4b7
Step 6/9 : COPY . /app
--> Using cache
--> 1d7ae54e5f76
Step 7/9 : RUN npm run build --prod
--> Using cache
--> 01db332efa3a
Step 8/9 : FROM nginx:1.17.1-alpine
--> ea1193fd3dde
Step 9/9 : COPY --from=build-step /app/dist/LoraClient /usr/share/nginx/html
--> Using cache
--> f75393fb70b2
Successfully built f75393fb70b2
Successfully tagged lora-web-app:latest
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$
```

Figura 29 Comando y proceso de construcción de imagen Docker [Autor].

- Una vez construida la Imagen Docker, se debe cambiar el nombre a la sintaxis aceptada por la nube que se está utilizando, en este caso particular y al estar utilizando la nube de Google, la sintaxis se puede ver en la Figura 30.

```
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$ sudo docker tag f75393fb70b2 us-east1-docker.pkg.dev/lora-client-repo/lora-web-app:latest
```

Figura 30 Nombre de imagen aceptado por la nube de Google [Autor].

- Para poder subir la imagen a un clúster de kubernetes en la nube de Google, se debe tener la imagen construida en la misma región y para subir esta imagen a la nube se debe hacer utilizando el comando que se ve en la Figura 31.

```
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$ sudo docker push us-east1-docker.pkg.dev/starlit-tube-317523/lora-client-repo/lora-web-app
Using default tag: latest
The push refers to repository [us-east1-docker.pkg.dev/starlit-tube-317523/lora-client-repo/lora-web-app]
459a884013df: Pushed
fbc0fc9bcf95: Layer already exists
f1b5933fe4b5: Layer already exists
latest: digest: sha256:730589c4633022f1993d51f18cfaa3103e649f12c609daf8155366681ba2ee74 size: 950
```

Figura 31 Comando para subir la imagen Docker a la nube de Google [Autor].

- Partiendo del hecho que ya se tiene la imagen Docker en un repositorio en la nube de Google, ya se puede iniciar con el despliegue de la aplicación en el clúster de kubernetes y el primer paso de este despliegue es crear un deployment con la imagen Docker que se quiere. Este deployment se crea mediante la utilización del comando que se muestra en la Figura 32.

```
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$ kubectl create deployment lora-web-app
--image=us-east1-docker.pkg.de/starlit-tube-317523/lora-client-repo/lora-web-app
deployment.apps/lora-web-app created
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$ kubectl get deployments
kubectl: command not found
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$ kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
cliente-lora	1/1	1	1	33d
cliente-lora-web	0/1	1	0	2m52s
cliente-lora2	1/1	1	1	30d
hello-server	1/1	1	1	33d
lora-client	0/1	1	0	33d
lora-web-app	0/1	1	0	47s
tomcat	1/1	1	1	33d

Figura 32 Comando de creación del deployment de kubernetes con la imagen Docker deseada [Autor].

- Una vez creado el deployment en el clúster, se debe exponer el servicio asociado a este y se realiza bajo la sintaxis mostrada en la Figura 33. Acá es importante tener en cuenta el puerto en el cual se quiere exponer la aplicación a desplegar.

```
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$ kubectl expose deployment lora-web-app
--type LoadBalancer --port 80 --target-port 80
service/lora-web-app exposed
jairopc@LAPTOP-024CDUUL:~/Lora2/LoraClient$
```

Figura 33 Comando para exponer a internet publico la aplicación [Autor].

- Al exponer el servicio de la aplicación, se habilitará una ip en la cual estará corriendo la aplicación en internet público y para conocer dicha ip es necesario acceder al clúster de kubernetes mediante la Interfaz gráfica de la nube utilizada, en este caso particular y como se puede ver en la Figura 34, se ingresó al clúster de kubernetes en la nube de Google.

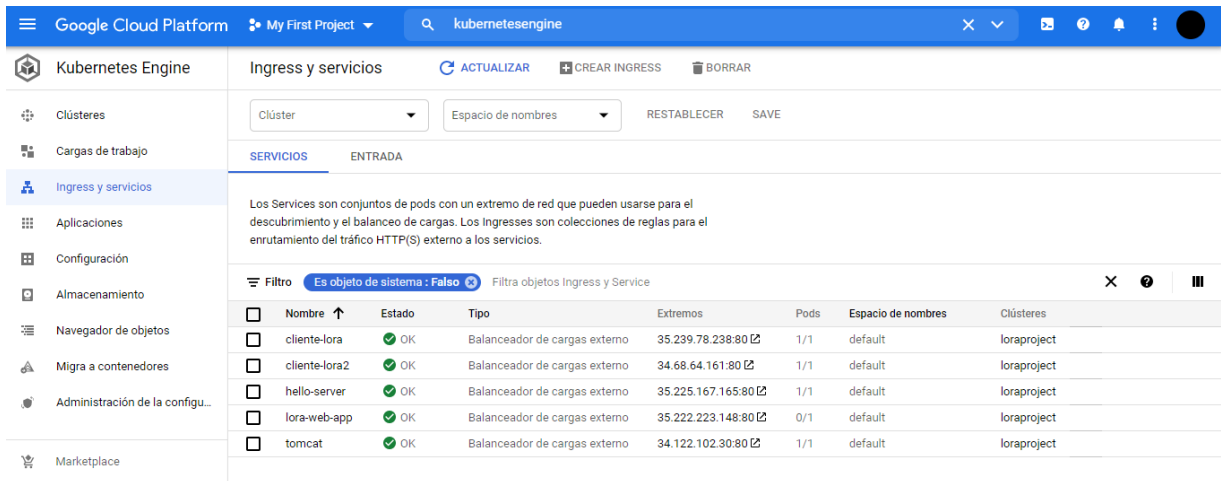


Figura 34 Interfaz gráfica de la nube de Google y cluster de kubernetes [Autor].

### 7.3 Diseño e implementación del nodo LoRa-GPS

Para un nodo LoRa-GPS es necesario realizar un diseño electrónico y estructural que cumpla con los requerimientos mínimos para su correcto funcionamiento del dispositivo dentro de la bicicleta.

#### 7.3.1 Diseño electrónico y PCB

El diseño electrónico se realiza, tomando en cuenta los módulos que componen el funcionamiento principal del nodo tales como el GPS, un microcontrolador y un transmisor LoRa con los cuales se lee, procesa y envía la geolocalización por medio del transmisor LoRa al gateway. Para este diseño se toma como parámetro un voltaje de funcionamiento estándar de 3.3v, para disminuir componentes electrónicos dentro del circuito y generar una comunicación entre los módulos sin necesidad de convertidores lógicos o reguladores de voltaje adicionales al general; que en consecuencia agranda el tamaño del diseño y su posible consumo.

El software de código abierto para la automatización del diseño electrónico utilizado es KiCad, con el cual se realiza el diseño del circuito electrónico, circuito impreso y archivos Gerber para la descripción de la PCB.

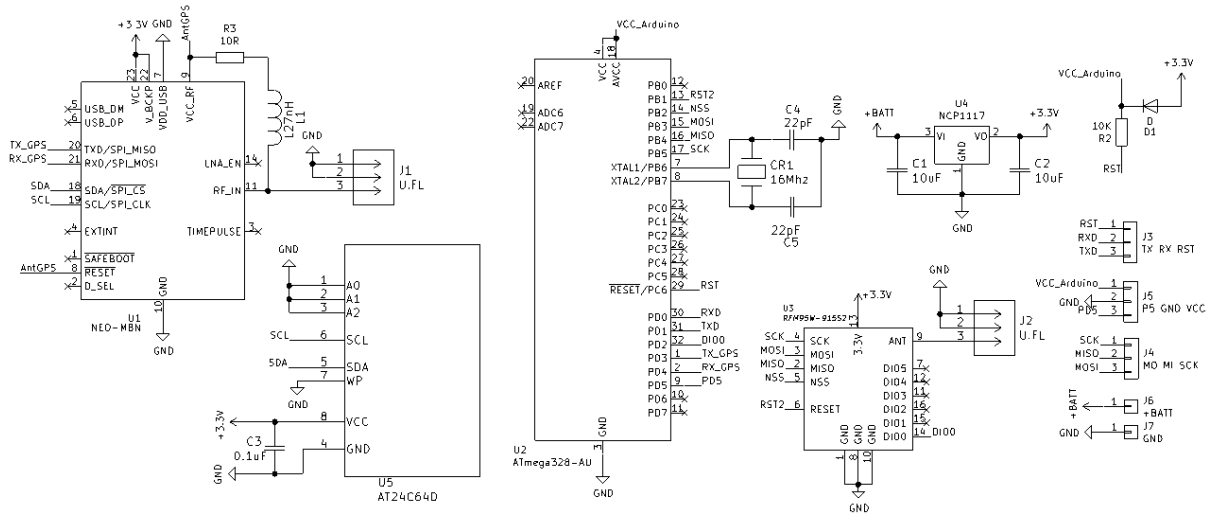


Figura 35 Circuito electrónico del nodo LoRa-GPS [KiCad – Autor]

En la Figura 35, se muestra el circuito electrónico completo para el nodo LoRa-GPS diseñado, en donde la mayoría de los símbolos y huellas de los componentes fueron extraídos de Ultra Librarian, una biblioteca CAD gratuita que utiliza estándares a nivel mundial de IPC-7351B para huellas de PCB y ANSI Y32.2-1975 para símbolos esquemáticos.

Para el módulo GPS se utiliza el chip NEO-6M de Ublox, principalmente por ser escalable en su diseño para mejoras futuras, utilizando las versiones más recientes como el NEO-M8N que maneja además otros sistemas de navegación global con satélites como el GLONASS o BeiDou. Por otro lado, la comunicación con este chip se puede realizar por UART, USB, I2C o SPI a diferencia de otros módulos o chips en el mercado como el L80 o L86 de Quectel que tienen únicamente comunicación UART. En la Figura 36 se observa las conexiones del chip NEO-6M acompañado de una memoria EEPROM (ATMEL 24C64D) que se comunica a través de I2C para retener configuraciones y últimos datos de posición del NEO-6M, también un conector U.FL para conectar la antena externa para recibir la señales de geolocalización y la conexión UART a 9600bps para la comunicación con el ATMEGA 328p.



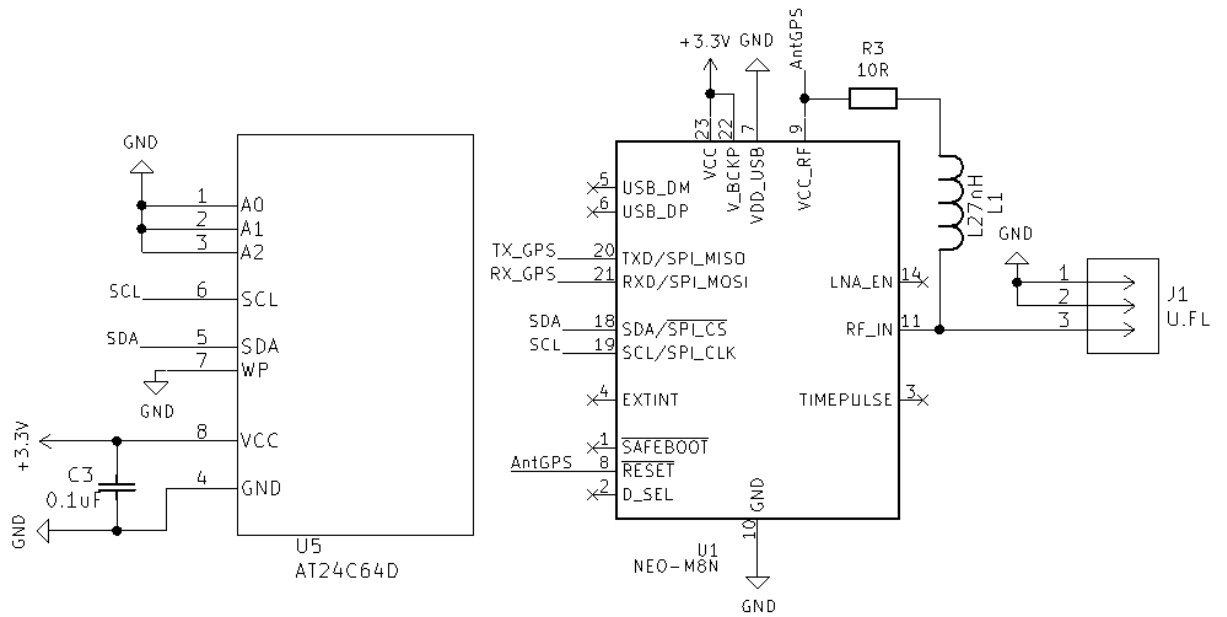


Figura 36 Circuito electrónico del módulo GPS [KiCad – Autor]

El microcontrolador utilizado es un ATMEGA328p que junto con el gestor de arranque de Arduino (Minicore) proporciona el AVR genérico por la cual funcionara la comunicación con los demás módulos y su programación mediante el IDE de Arduino. En la Figura 37 se muestra las conexiones del ATMEGA328p mediante SPI con el módulo transceptor LoRa Rfm95; comunicación UART con el módulo GPS; alimentación de 3.3v con un reloj externo de 16Mhz y por último una salida digital, conexión a pines UART y SPI para su programación.

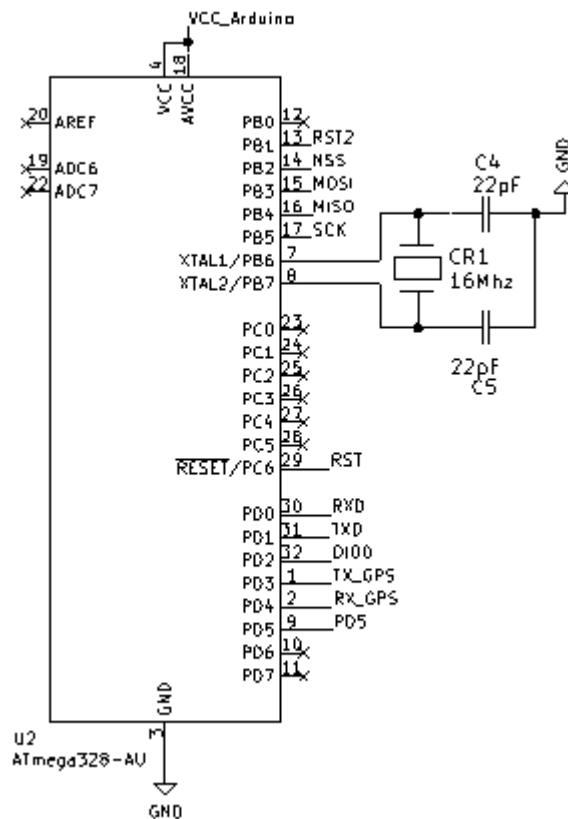


Figura 37 Circuito electrónico y conexiones del ATMEGA328p [KiCad – Autor]

En la Figura 38 se encuentran las conexiones del módulo transceptor LoRa Rfm95 basado en el chip Sx1276 a una frecuencia de 915Mhz y alimentación de 3.3v siendo el chip más comercial y utilizado para la modulación LoRa, el circuito cuenta con una conexión U.FL la conexión de la antena y la comunicación SPI con el ATMEGA328p.

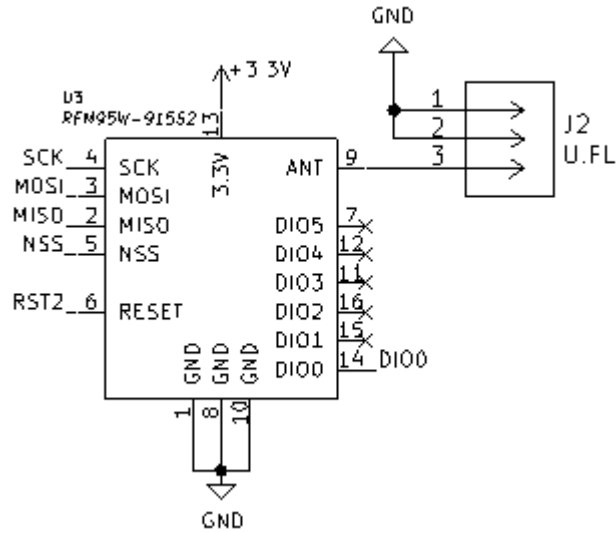


Figura 38 Circuito electrónico y conexiones del módulo transceptor LoRa Rfm95 [KiCad – Autor]

En la Figura 39 se muestran los pines de entrada y salida que se utiliza para realizar la programación del ATmega328P y pruebas de funcionamiento.

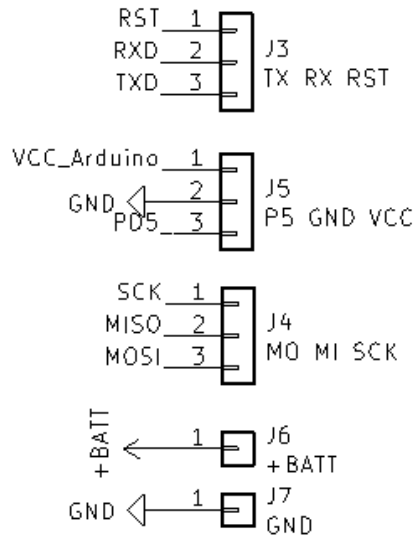


Figura 39 Pines de entrada y salida del nodo LoRa-GPS [KiCad – Autor]

Por último, la Figura 40 muestra un LDO (Regulador de baja caída) NCP1117 de 3.3v, para la alimentación completa del circuito, este regulador puede alcanzar una corriente salida de 1A con una entrada de 4.8v hasta 18v proporcionados por baterías.

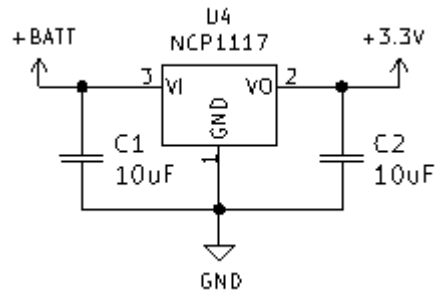


Figura 40 Circuito electrónico de regulador de voltaje a 3.3v [KiCad – Autor]

El diseño del circuito impreso se realiza teniendo en cuenta que el dispositivo va a estar en la parte inferior del sillín de una bicicleta, sus dimensiones son de aproximadamente 67x33mm, para poder acondicionar el dispositivo en la parte larga del sillín, como se puede observar en la Figura 41.

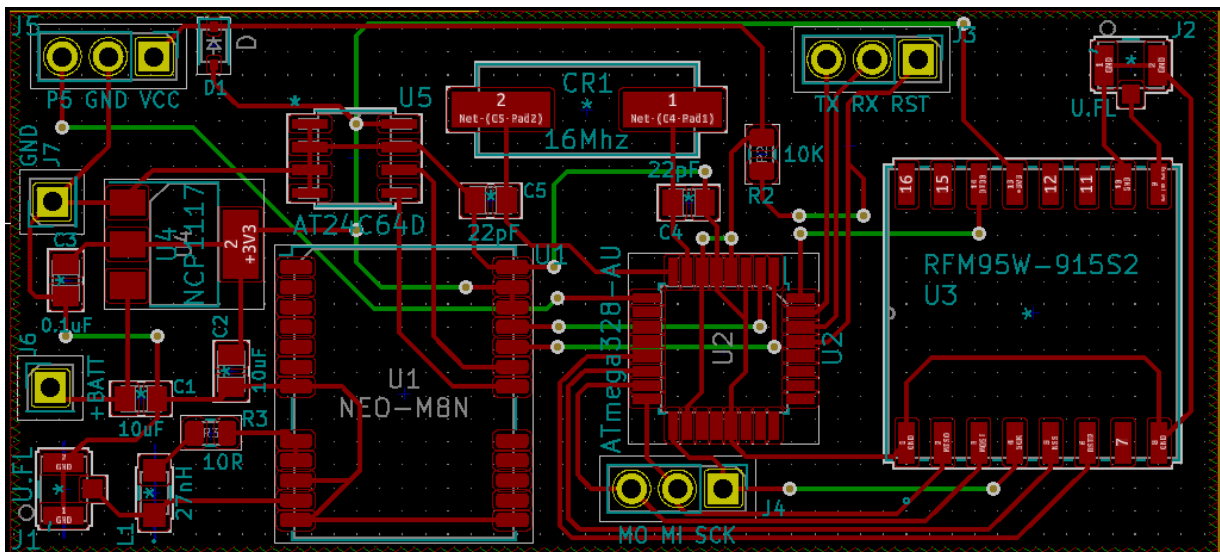


Figura 41 Diseño del circuito impreso [KiCad – Autor]

El diseño de la PCB está descrito por algunas reglas generales configuradas en KiCad para el ancho mínimo de las pistas, vías de taladros y agujeros, con las que se asegura que la manufactura no sea especializada y con un presupuesto elevado, en la Tabla 3 se muestra las reglas asignadas.

Tabla 3 Reglas configuradas en KiCad para el diseño de la PCB [Autor]

Regla de diseño	Tamaño mínimo (mm)
Ancho de pista	0.127
Diámetro de vía	0.5
Taladro de vía	0.4
Distancia entre agujeros	0.25

Aunque estas reglas están descritas en el software, el ancho de pista mínimo utilizado es de 0.254mm y la separación de áreas rellenas es de 0.508mm. El diseño utiliza dos capas, la primera capa (capa superior) se encuentra los componentes electrónicos de superficie como se ilustra en la Figura 42 , por otro lado, la segunda capa se utiliza para interconexiones de componentes como se ilustra en la Figura 43.

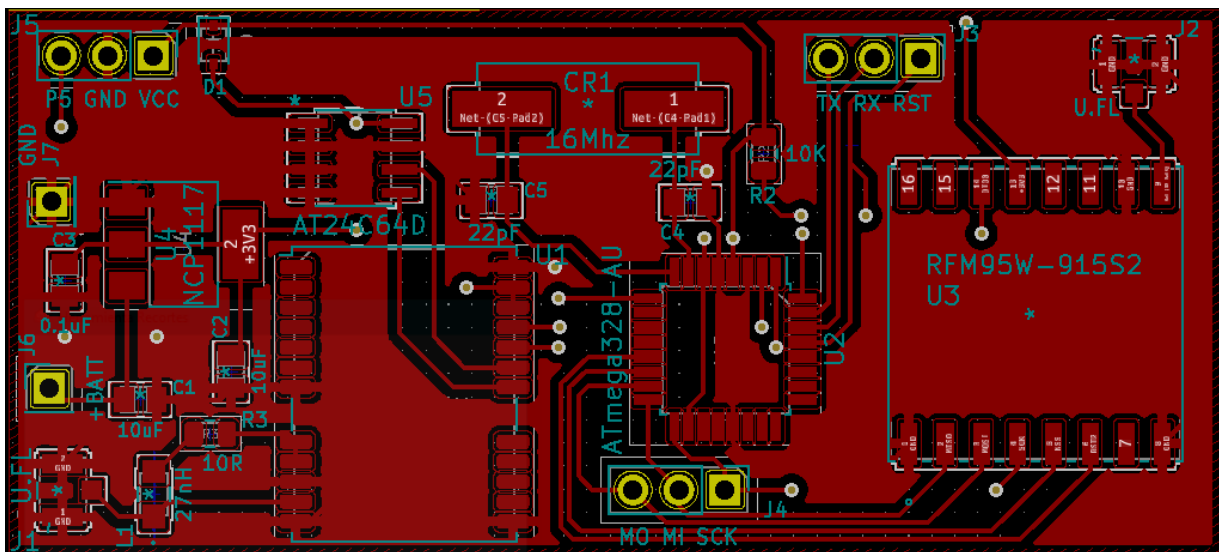


Figura 42 Diseño de diseño impreso capa superior [KiCad – Autor]

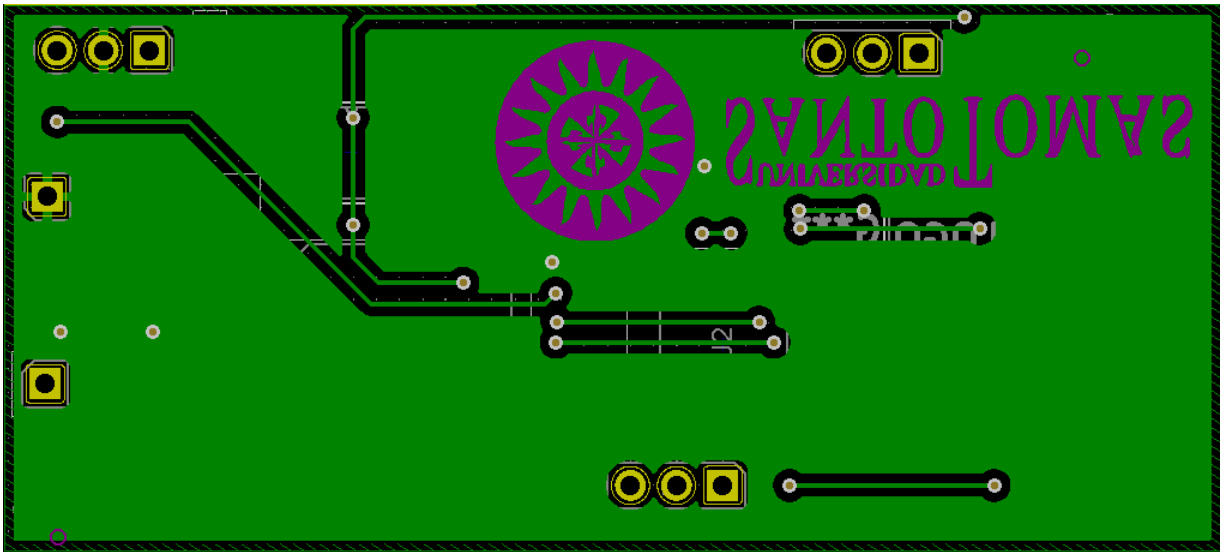


Figura 43 Diseño de diseño impreso capa inferior [KiCad – Autor]

En el diseño se tiene en cuenta que ninguna señal de las antenas este muy cerca de otras señales que puedan generar ruido, como lo advierte el fabricante, por último, se generan los archivos gerber para su manufactura. Kicad ofrece una modelo 3D en el cual se puede observar una vista muy similar a la PCB física como se ilustra en la Figura 44 y Figura 45.

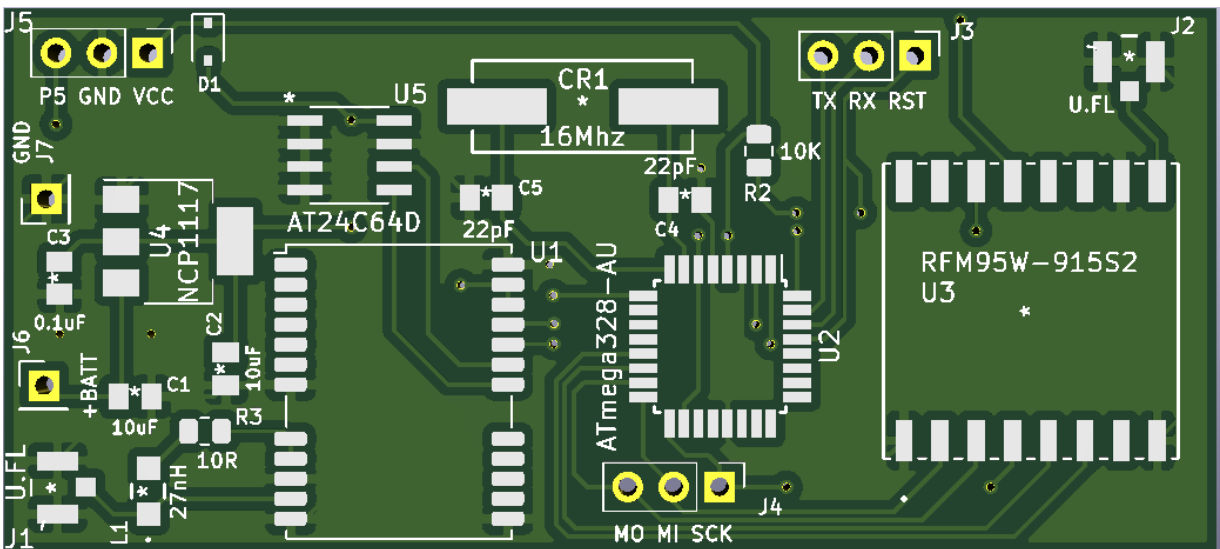
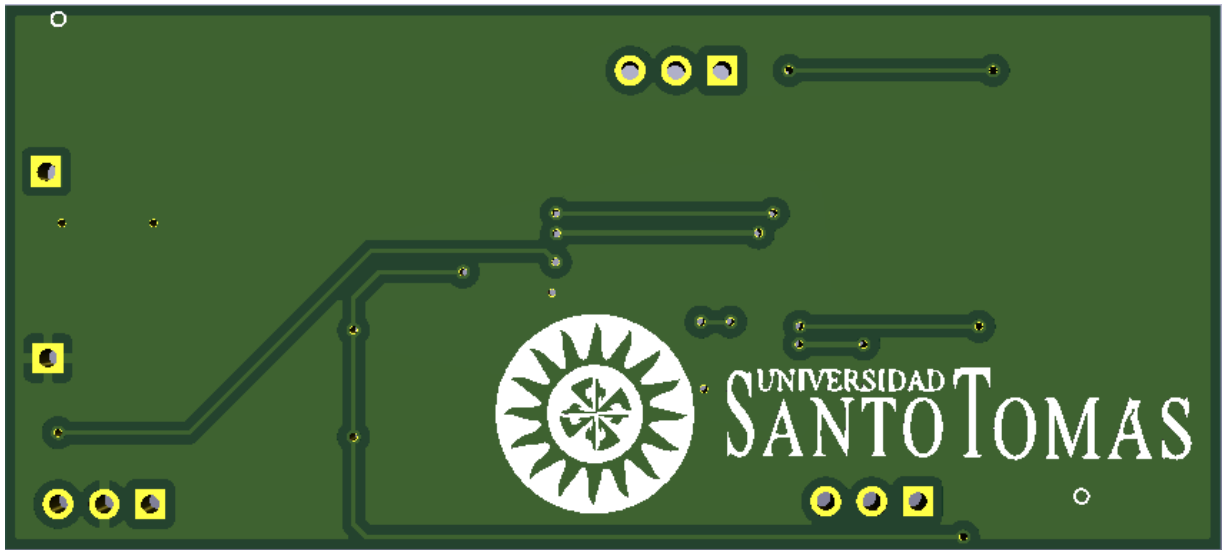


Figura 44 Modelo en 3D del circuito impreso capa superior [KiCad – Autor]



*Figura 45 Modelo en 3D del circuito impreso capa inferior [KiCad – Autor]*

Para el montaje de los componentes en la PCB, se tiene en cuenta las temperaturas y tiempos especificados por el fabricante de cada chip, módulo y/o componente electrónico para la correcta soldadura, además de las advertencias de descargas electrostáticas o daños que se puedan generar durante la soldadura. Luego se procede a acomodar los componentes con unas pinzas de precisión antiestáticas y se soldan utilizando la estación de calor Yaxun-703d.

El módulo RFM95 se debe soldar después de realizar la carga del gestor de arranque y del programa al ATmega328P como se describe en la sección 7.3.2 para que no exista un conflicto por medio de la programación mediante SPI. El montaje final del circuito impreso con los componentes soldados se puede observar en la Figura 46 y Figura 47.



Figura 46 Circuito impreso soldado, capa superior [Autor]



Figura 47 Circuito impreso soldado, capa inferior [Autor].

### 7.3.2 Carga de gestor de arranque de Arduino y programación del nodo LoRa-GPS

La carga del gestor de arranque de Arduino al ATmega328P se debe realizar sin que el módulo RFM95 este soldado a la placa, ya que la programación se realiza



por SPI y el módulo RFM95 se comunica de la misma forma con el ATmega328P generando un conflicto por la definición de esclavos y maestro, además el voltaje para utilizar en la carga del gestor de arranque y programa, se realiza a 5V, que puede dañar el módulo RFM95. La carga del gestor se realiza por medio de un Arduino conectado al nodo mediante los pines que se muestran en la Figura 39 y su conexión se especifica en la Tabla 4.

*Tabla 4 Conexiones para carga de gestor de arranque y programación del Nodo LoRa-GPS [Autor]*

<b>Arduino (Mega-Uno-Nano)</b>	<b>Nodo Lora-GPS</b>
<b>Vcc 5V</b>	VCC
<b>Gnd</b>	Gnd
<b>MISO</b>	MISO
<b>MOSI</b>	MOSI
<b>SCK</b>	SCK
<b>SS</b>	RST

Luego de conectado el nodo con el Arduino, se carga el gestor de arranque con el IDE de Arduino en la opción “Herramientas -> Quemar Bootloader”, que permite después de cargado, programar el ATmega328P por medio de ISP con las mismas conexiones, cargando el mismo programa creado para el de nodo comercial explicado en el diagrama de flujo de la Figura 21, editando únicamente la clave de sesión de red.

### **7.3.3 Diseño estructural**

Para cumplir con el objetivo de integrar el nodo LoRa en una bicicleta fue necesario diseñar e imprimir en 3D una estructura con capacidad suficiente para llevar el nodo y la batería de alimentación.

La estructura diseñada fue una caja (Figura 48) con dimensiones de 110mm de larga, 90mm de ancha y 30mm de alta. Esta caja aparte de tener la capacidad de llevar el nodo y la batería se diseñó para poder integrarla y asegurarla a la bicicleta con tornillos.

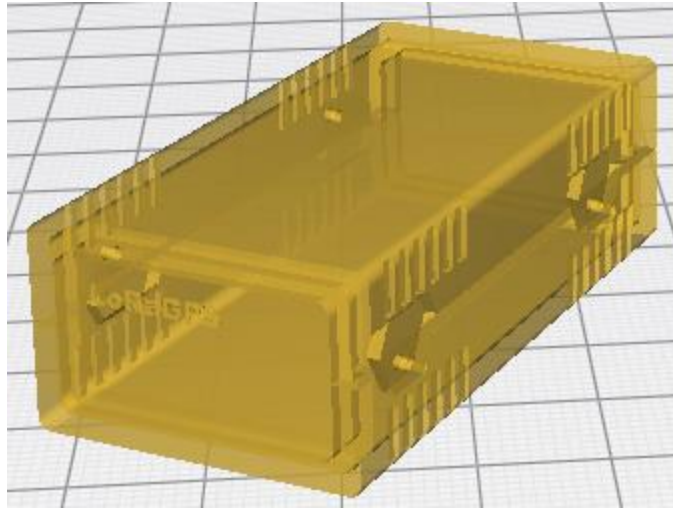


Figura 48 Diseño estructural completo para el nodo [Autor].

La caja se diseñó de una manera que fuera fácil armar y desarmar para que el usuario final pueda acceder al nodo y extraer la batería para poder recargarla. La Figura 49 muestra los componentes de la estructura diseñada en el software de diseño 3D Ultimaker.

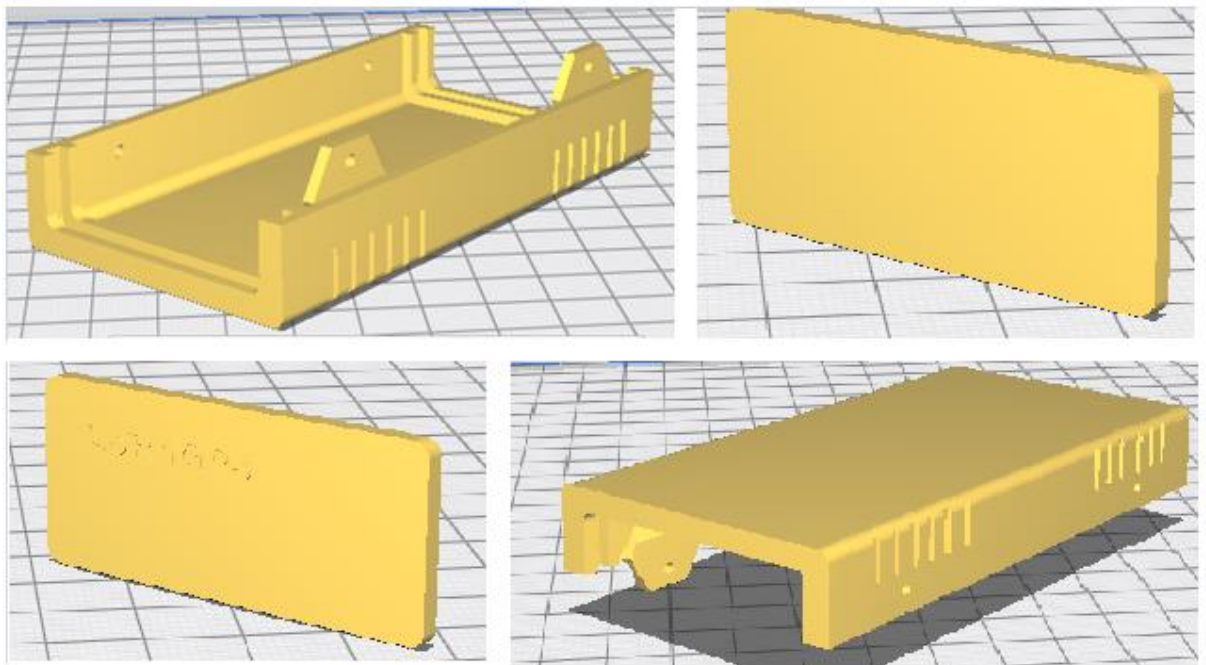


Figura 49 Diseño estructural por partes para el nodo [Autor].

La impresión en 3D de los componentes de la estructura mostrada en la Figura 49, dio como resultado una caja en color negro que se muestra en la Figura 50 y en la Figura 51 su montaje en una bicicleta.



*Figura 50 Estructura impresa en 3D para el nodo [Autor].*



*Figura 51 Nodo LoRa-GPS integrado a una bicicleta [Autor].*

## 8. RESULTADOS

Implementado e integrado el dispositivo LoRa-GPS a una bicicleta se realiza la comprobación de funcionamiento en el area urbana de Bogotá con el ambiente de pruebas, en donde el Gateway se ubica en el ultimo piso de la biblioteca de la Universidad Santo Tomás para realizar un registro de un recorrido en las cercanias de la universidad, como se observa en la Figura 52, se demarca el recorrido que se realizo (color verde), la ubicación del gateway (triángulo amarillo) y los puntos registrados por el nodo LoRa-GPS (marcadores azules).

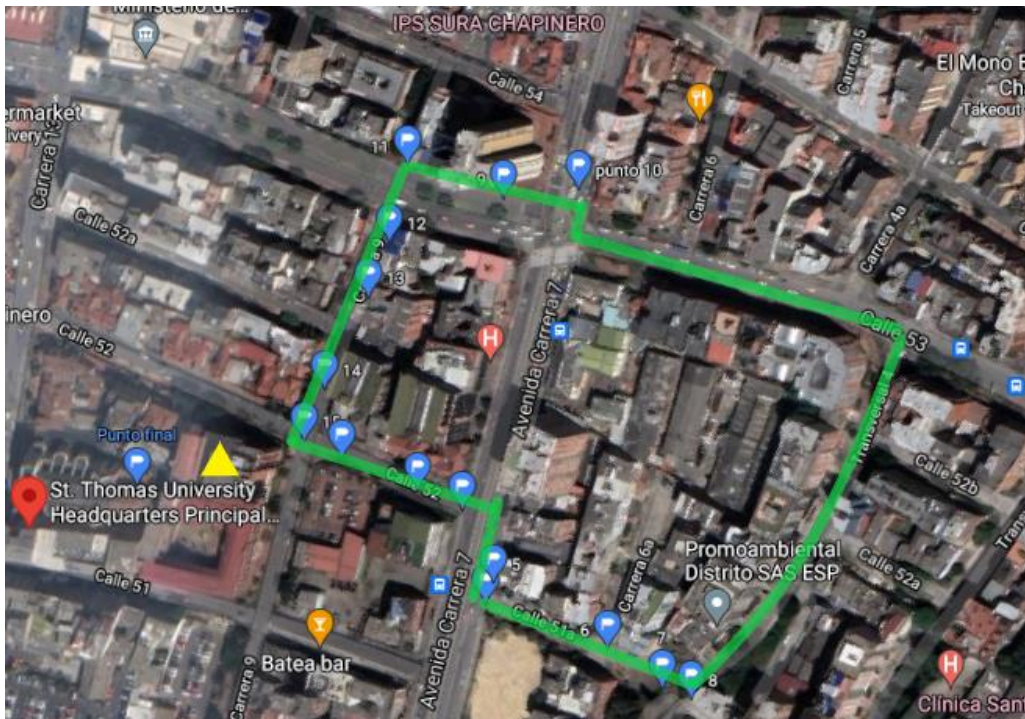


Figura 52 Prueba #1 - Recorrido alrededor de la Universidad Santo Tomas con los datos obtenidos por el nodo LoRa-GPS diseñado [40].

Como se puede observar en la Figura 52 los datos obtenidos no comprenden todo el recorrido, ya que la densidad de área que generan las edificaciones deteriora la señal enviada por el nodo LoRa-GPS, generando que él envié del paquete LoRa al gateway no se realice. Además, el gateway de pruebas usado (Sección 5.4 ) advierte



que en una densidad de área elevada se puede reducir su rango a menos de 500m. En la Figura 53 se puede observar que la densidad de predios cercana a la Universidad Santo Tomas es alta, tomando en cuenta que el recorrido (color verde) y la ubicación del gateway (triángulo amarillo) están en ubicados en esta zona.

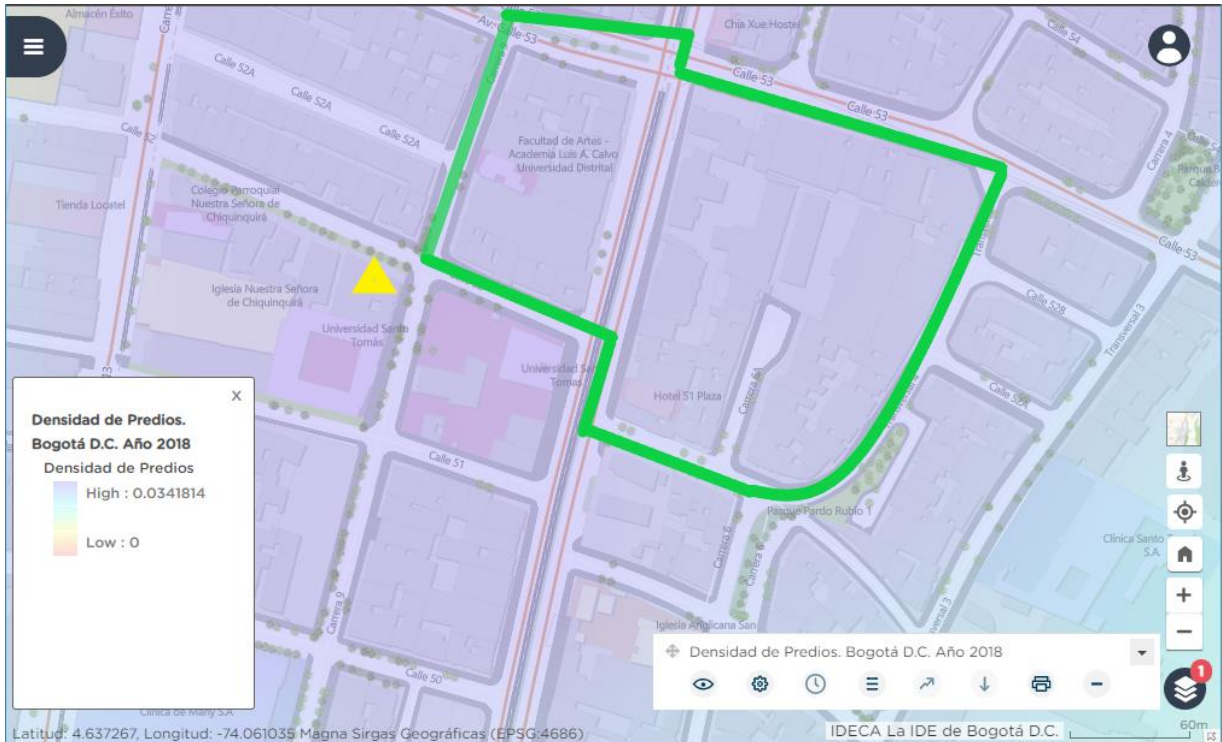


Figura 53 Prueba #1 - Densidad de predios cerca de la Universidad Santo Tomas donde se realiza el recorrido Fuente: Mapas de Infraestructura de Datos Espaciales de Bogotá (IDECA) [41]

La mayor distancia en donde el nodo LoRa-GPS tuvo comunicación con el gateway en el recorrido fue de aproximadamente 240 metros como se puede observar en la Figura 54.

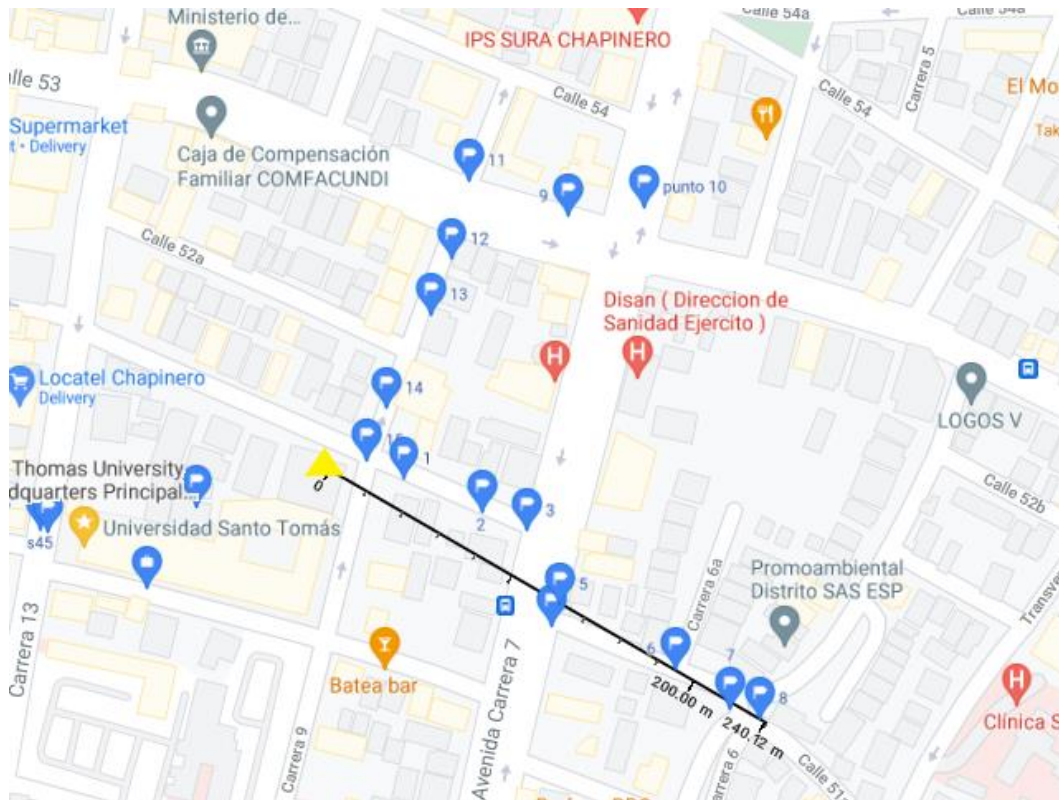


Figura 54 Prueba #1 - Medida del punto más lejano de comunicación del nodo LoRa-GPS con el gateway [40].

Se realizó otro recorrido en una zona urbana con menos densidad de predios al sur de Bogotá, para comprobar el alcance. Se puede observar en la Figura 55, el gateway (triángulo rojo) que se ubica en el tercer piso de una edificación, las ubicaciones registradas por el nodo LoRa-GPS (marcadores azules) y el desplazamiento por la zona (color azul claro).

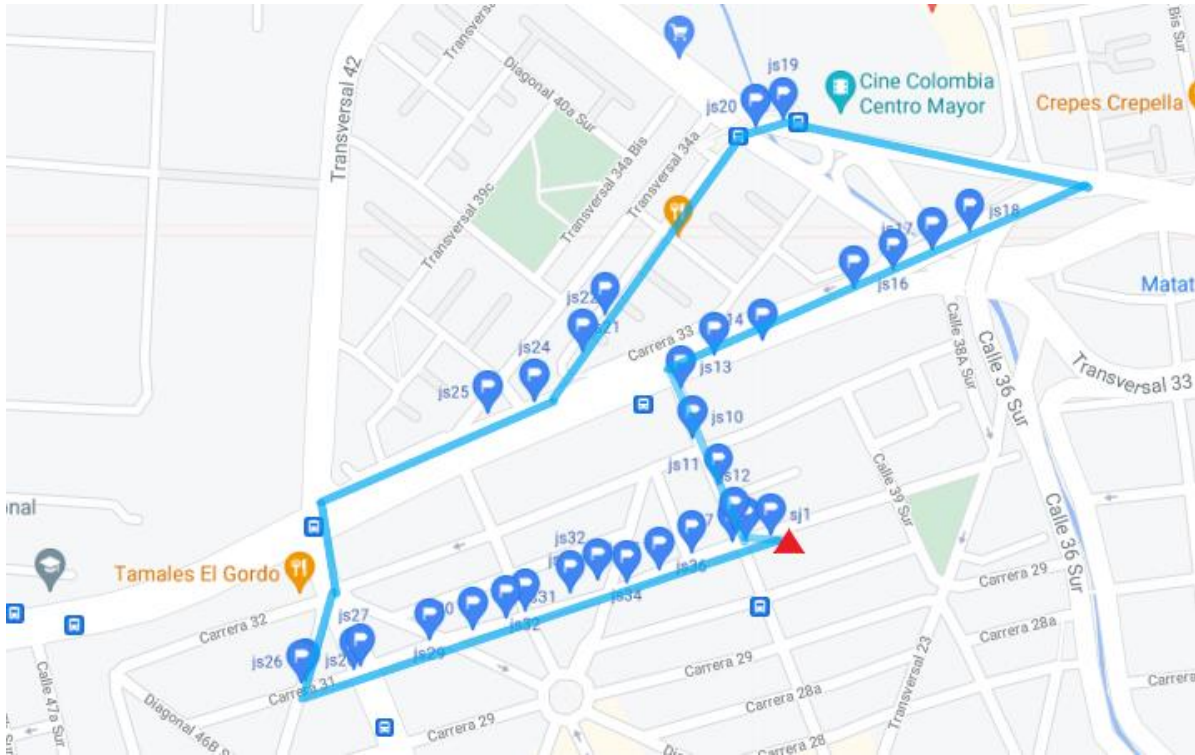
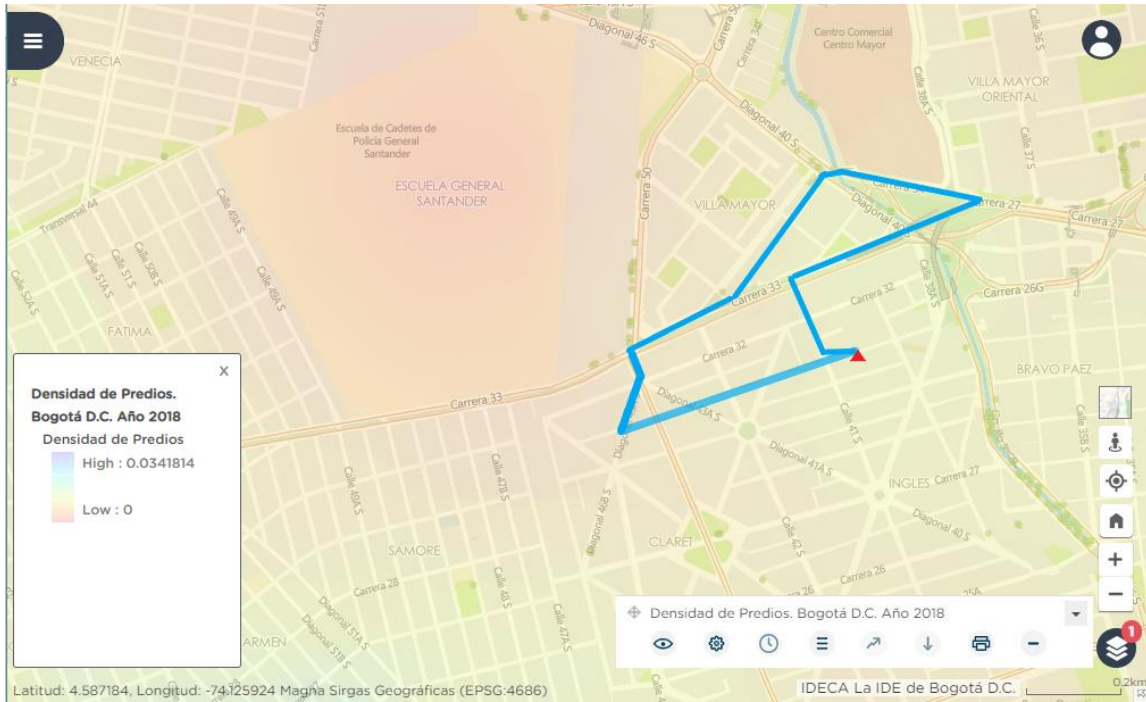


Figura 55 Prueba #2 - Recorrido de prueba con los datos obtenidos por el nodo LoRa-GPS diseñado [40].

En la Figura 56 se puede observar que la densidad de predios cercana a zona de Villa Mayor, Bogotá es más baja, tomando en cuenta que el recorrido (color azul claro) y la ubicación del gateway (triángulo rojo) están en ubicados en esta zona.



*Figura 56 Prueba #2 - Densidad de predios en la zona de Villa Mayor donde se realiza el recorrido Fuente: Mapas de Infraestructura de Datos Espaciales de Bogotá (IDECA) [41].*

En la Figura 57 y en la Figura 58 se puede observar un mismo recorrido con mediciones de distancia, que corresponden a la mayor distancia obtenida en línea de vista y medición con obstáculos como edificios respectivamente. Como se puede observar, cuando el recorrido es en línea de vista, se alcanza una recepción de datos mejor (aproximadamente 477 metros) superando la mayor distancia obtenida con obstáculos (aproximadamente 390 m).



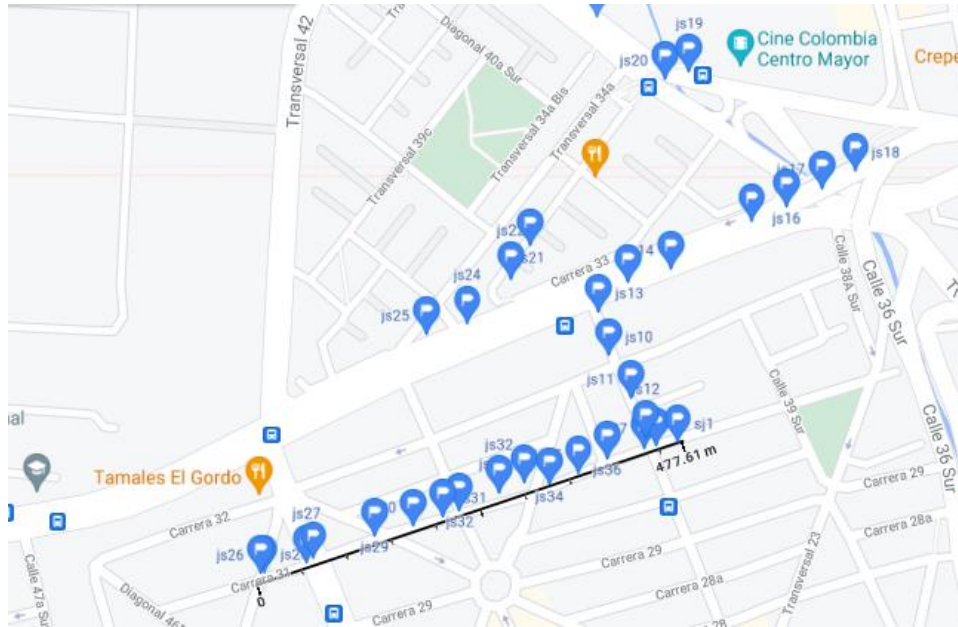


Figura 57 Prueba #2 - Medida del punto más lejano de comunicación del nodo LoRa-GPS con el gateway a línea de vista [40].

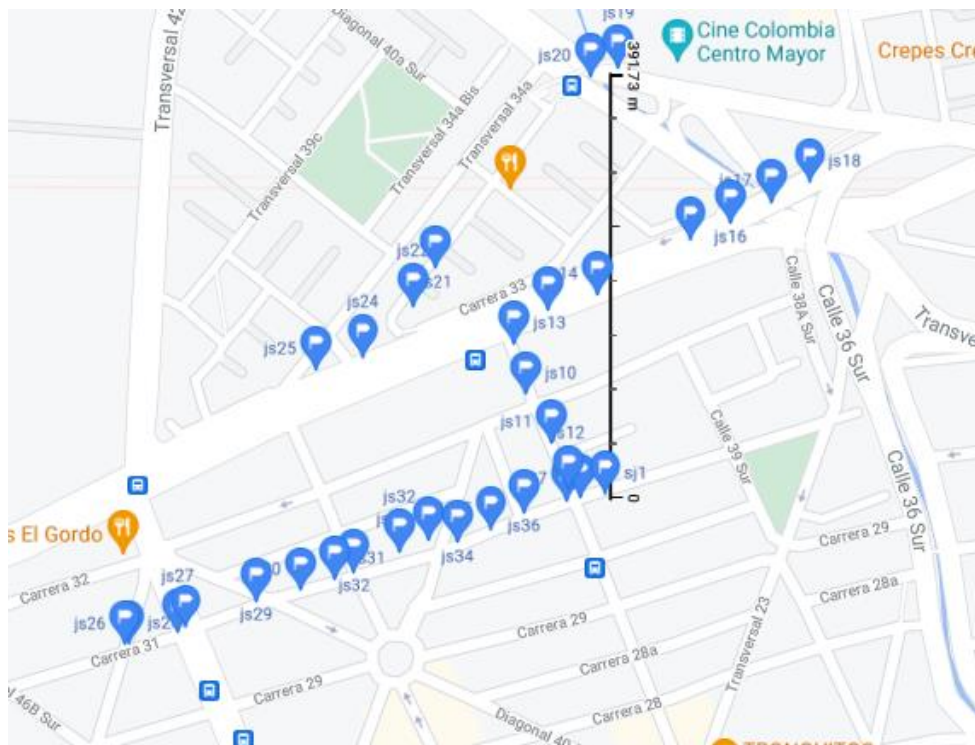


Figura 58 Prueba #2 - Medida del punto más lejano de comunicación del nodo LoRa-GPS con el gateway con obstáculos [40].

Se realiza un tercer recorrido a línea de vista, tomando como punto de inicio los laboratorios de la Universidad Santo Tomás ya que desde ellos se tiene una línea de vista por toda la Calle 51 hasta la NQS. Como se observa en la Figura 59 los marcadores azules son los datos tomados por el nodo LoRa-GSP enviados al servidor.

Al tratarse de una prueba en línea de vista, el recorrido que fue truncado por una zona gubernamental a la cual no se podía acceder sin un permiso, por lo cual el recorrido máximo de esta prueba no es el alcance máximo del dispositivo.

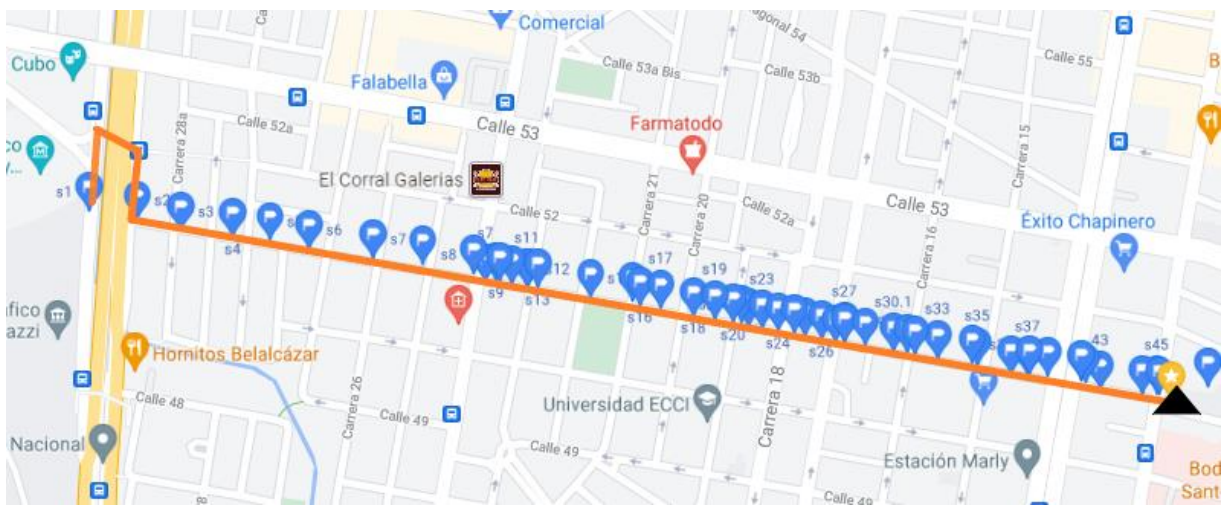


Figura 59 Prueba #3 – Recorrido de prueba (Calle 51, Bogotá) con los datos obtenidos por el nodo LoRa-GPS diseñado en línea de vista [40].

La mayor distancia en línea de vista en donde el nodo LoRa-GPS desarrollado tuvo comunicación con el gateway en el recorrido fue de aproximadamente 1600 metros como se puede observar en la Figura 60.

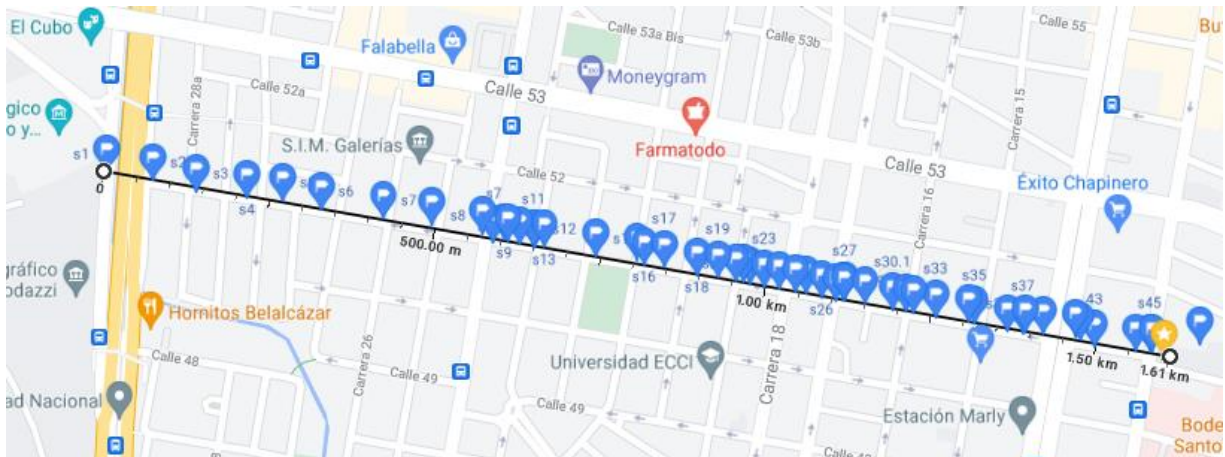


Figura 60 Prueba #3 - Medida del punto más lejano de comunicación del nodo LoRa-GPS con el gateway a línea de vista [40]

Una de las funcionalidades adicionales que se le implementaron al proyecto con el fin de que el tiempo de funcionamiento fuera lo máximo posible, es la capacidad de apagar la mayor cantidad de componentes y solo encenderlos al momento de enviar los datos de ubicación. En la Figura 61, se puede observar el consumo de corriente que está teniendo el nodo LoRa-GPS cuando este está en estado de reposo y cuando se activa para enviar los datos, con un envío aproximadamente cada 8 segundos.

En el tiempo de reposo en el cual el consumo de corriente es menor, únicamente está activo el módulo GPS que busca continuamente datos, para que al momento de encender el resto de los componentes y necesitar la geolocalización, la respuesta sea casi instantánea, ahorrando energía y brindando más horas de funcionamiento al dispositivo. Por esta razón el nodo final envía datos en un intervalo de 30 segundos.

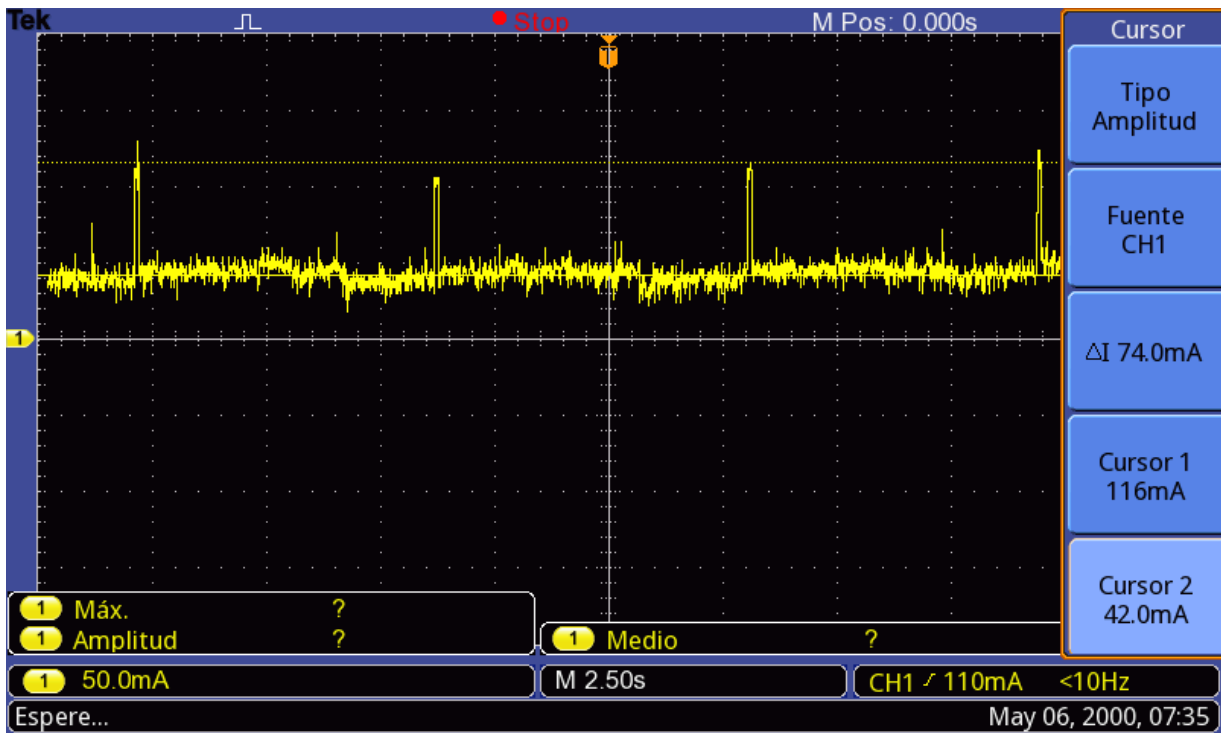


Figura 61 Consumo de corriente del nodo LoRa-GPS en estado de reposo y envió de datos [Autor].

Al desarrollar el proyecto con LoRaWAN y estar localizados en Colombia, se utilizó la banda permitida y programada la cual es 914.9 MHz y se comprobó que efectivamente es esa banda mediante un analizador de espectro, teniendo resultados como los que se pueden observar en la Figura 62 y en la Figura 63. En estas imágenes se puede ver que el momento en el cual se envía un dato desde el nodo LoRa-GPS con la frecuencia programada y a su vez el pico acompañado de la potencia de la señal en decibelios dBm.

La potencia que sale en la Figura 62 de -23dBm a diferencia de la potencia de la Figura 63 de -7.8dBm es más baja y la razón de este cambio es porque el nodo LoRa-GPS estaba alejado del gateway por aproximadamente 5 metros de distancia y con obstáculos cuando se realizó esta prueba.





Figura 62 Medición de la potencia en la frecuencia de 914.9 MHz al enviar de un paquete LoRa 5 metros de distancia entre el gateway y el nodo LoRa-GPS con obstáculos [Autor].



Figura 63 Medición de la potencia en la frecuencia de 914.9MHz al enviar de un paquete LoRa 20 centímetros de distancia entre el gateway y el nodo LoRa-GPS sin obstáculos [Autor].

## 9. CONCLUSIONES Y TRABAJO FUTURO

Del estudio de los requerimientos necesarios para la implementación del sistema, se identificó que las características más relevantes que debe tener un nodo LoRa-GPS para ser implementado como sistema de rastreo en una bicicleta, estas características son: trabajar sobre una red LoRaWAN cumpliendo con todos los lineamientos de la tecnología LoRa para un sistema de rastreo, hacer un correcto análisis y selección de dispositivos que brinden una precisión en la obtención de coordenadas para un objeto pequeño como una bicicleta, tener en cuenta que el dispositivo debe estar funcionando sin interrupción por cierta cantidad de tiempo y para esto es necesario hacer un análisis sobre el consumo y la autonomía, las cuales fueron aplicadas, integradas y puestas en funcionamiento en el nodo LoRa-GPS, probando que en conjunto dan como resultado un sistema de rastreo para bicicletas funcional y ofrecer una experiencia de usuario para su uso.

Se evidencia que un ambiente de pruebas conformado por: nodos LoRa, ya sea el desarrollado en este proyecto o comerciales; un gateway que genere una red LoRaWAN, sobre la cual se realiza la conexión; un servidor de nube y una aplicación de visualización, es suficiente para la validación funcional del nodo LoRa-GPS, tal como se realizó en este trabajo, teniendo en cuenta, que esta implementación no es viable en un ambiente de producción en el mercado, ya que los requerimientos de una red LoRaWAN son mucho más robustos y con equipos de mayor capacidad tanto en potencia y conexión.

La aplicación desarrollada realiza la obtención de los datos cifrados de latitud y longitud del servidor mediante el llamado de un API de lectura, luego de eso mediante la ejecución se realiza el desciframiento estos datos y finalmente los muestra al usuario final, permitiéndole acceder mediante un hipervínculo al marcador de Google Maps con la posición exacta del nodo LoRa-GPS. En conjunto, este proceso es suficiente para tener una validación del funcionamiento actual del sistema.

Teniendo en cuenta que LoRaWAN se preocupa por la seguridad en una implementación IoT, se agregaron las capas de cifrado mediante un protocolo AES y con dos llaves de 128 bits se cifraban los datos recogidos de latitud y longitud antes de enviarlos. Este proceso adicional forzó a utilizar la memoria SRAM para guardar variables constantes del programa como las llaves de cifrado (La llave de sesión de la aplicación y la llave de red) al igual que los seriales logs del programa de

Arduino, ya que consumía más recursos de los que brinda la memoria RAM y al ejecutar el cifrado de la información el Arduino se quedaba sin recursos para su ejecución.

Para el desarrollo de la aplicación web, se seleccionó el framework Angular principalmente porque es uno de los más utilizados comercialmente según las estadísticas de la cuota de mercado mundial, porque tiene un alto nivel de integración entre componentes, posee amplia disponibilidad de bibliotecas y por supuesto porque tiene el respaldo de Google. Al hacer una revisión de la aplicación se puede observar que cumple con los requerimientos de diseño establecidos, ofreciendo al usuario final una interfaz gráfica amigable y entendible, pero sobre todo que cumple con su propósito el cual es poder visualizar en todo momento los datos que están llegando de la posición GPS de la bicicleta mediante un hipervínculo hacia Google Maps.

Durante el desarrollo del proyecto se evidenció que a mayor distancia entre el nodo Lora-GPS y el gateway, el tiempo de recepción de datos aumenta, lo que genera que los datos no se envíen cada 30 segundos como está definido en el código, generando que en el informe de la aplicación web, los datos puedan llegar a demorarse hasta el doble del tiempo establecido, generando un tiempo muerto bastante considerable.

El dispositivo funcional desarrollado en el proyecto fue sometido a pruebas de funcionamiento en diferentes ubicaciones en la ciudad de Bogotá, lo que permitió confirmar que los datos que tomaba el nodo LoRa-GPS y que se subían al servidor para ser visualizados, no sobrepasaban los 20 metros de error al compararla con la ubicación real del dispositivo, mostrando en la mayoría de los casos un error entre 5 a 10 metros, por lo cual el rango de búsqueda de una bicicleta para el que fue diseñado el dispositivo es bueno.

El proyecto que se desarrolló es sólo una pequeña muestra de lo que se puede llegar a hacer con una red y tecnología LoRa, no solo en aplicaciones de rastreo sino en cualquier dispositivo que se quiera conectar u obtener información de sensores para satisfacer las necesidades de distintas industrias, ya que es posible desarrollar una aplicación mucho más robusta para muchos más ámbitos de rastreo y recolección de datos.

El dispositivo diseñado e implementado cuenta con un alcance y robustez limitado puesto que el ambiente de pruebas compuesto por un gateway solo contaba con un canal y la potencia de este no permitía llegar a diferentes zonas de la ciudad,

por lo cual para una próxima versión del dispositivo se podría ampliar en gran medida el rango implementando una red LoRa en toda la ciudad con uno o varios gateway multicanal (8 canales) de alta potencia.

Por último, otro punto para mejorar en una nueva versión del dispositivo es la batería y la forma de cargar la misma, ya que el dispositivo funcional implementado cuenta con una batería de ion de litio recargable 18650 con un voltaje nominal de 3.7V y una capacidad nominal de 8800mAh que, en proyectos o trabajos futuros, podría mejorarse aumentando la capacidad de corriente y por ende el tiempo de funcionamiento del nodo LoRa-GPS. De igual manera al cargar la batería, se podría diseñar un sistema de carga sin necesidad de extraerla como, por ejemplo: carga inalámbrica, carga con energía cinética producida por la bicicleta o carga desde un panel solar.



## 10. GLOSARIO

Se presenta un glosario, con las palabras de mayor relevancia dentro del proyecto.

- **Activation by Personalization (ABP):** Forma de predeterminar las llaves de sesión de red y de aplicación en la producción de un dispositivo.
- **Gateway:** Puerta de enlace de red que proporciona intercambio de información entre redes.
- **Global Positioning System (GPS):** Sistema de Posicionamiento Global que proporciona a los usuarios servicios de posicionamiento, navegación y cronometraje.
- **Global System for Mobile communications (GSM):** Estándar en telecomunicaciones móviles más extendido que describe los protocolos para redes celulares digitales de segunda generación.
- **Industrial, Scientific and Medical (ISM):** Son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica.
- **Low Power Wide Area (LPWA):** Término genérico para una red de baja potencia y área amplia.
- **Printed Circuit Board (PCB):** Es una placa de circuito impreso que sostiene y conecta componentes electrónicos por medio de soldadura en láminas de cobre.
- **Serial Peripheral Interface (SPI):** Interfaz de comunicación serial de corta distancia utilizado mayormente en sistemas embebidos.
- **The Internet of things (IoT):** Hace referencia a una red de objetos físicos “Cosas” que están conectados con sensores, tienen algún tipo de software y otras tecnologías con el objetivo de conectar y transmitir datos con otros dispositivos por medio de internet [42].
- **Global Navigation Satellite System (GNSS):** Hace referencia a un conjunto de satélites que transmiten datos de posicionamiento global, pero que a su vez realizan una sincronización a los receptores GNSS y estos receptores utilizan estos datos para determinar la ubicación exacta en el globo terráqueo.
- **Time Difference of Arrival (TDoA):** Este esquema se basa en la medición precisa de la diferencia de tiempo entre la llegada de las señales a los receptores.

- **Over-the-Air (OTA):** Es un método para distribuir actualizaciones de software a dispositivos que utilizan comunicación inalámbrica.

## 11. REFERENCIAS BIBLIOGRÁFICAS

- [1] SDM, “Encuesta de Movilidad 2019”, Bogotá D.C, 2019.
- [2] L. Barreto, “Peñalosa declaró a Bogotá ‘Capital Mundial de la Bicicleta’”, 2018. [En línea]. Disponible en: <https://bogota.gov.co/mi-ciudad/gestion-publica/bogota-es-la-capital-mundial-de-la-bicicleta-penalosa>.
- [3] El Espectador, “¿Cómo consolidar la capital mundial de la bicicleta?”, Bogotá D.C, 2019. [En línea]. Disponible en: <https://www.elespectador.com/opinion/editorial/como-consolidar-la-capital-mundial-de-la-bicicleta-articulo-848914/>.
- [4] M. Gómez y F. G. Altamar, “¿Capital mundial o mortal de la bici?”, Bogotá D.C, 2019. [En línea]. Disponible en: <https://www.elespectador.com/noticias/bogota/capital-mundial-o-mortal-de-la-bici/>.
- [5] El Espectador, “Preocupa incremento de hurtos violentos en Bogotá”, Bogotá D.C, 2019. [En línea]. Disponible en: <https://www.elespectador.com/noticias/bogota/preocupa-incremento-de-hurtos-violentos-en-bogota/>.
- [6] Radu Mihailescu, “LoRa-positioning in Malmö compared with GPS possibilities, power consumption & accuracy”, University Malmö, 2018.
- [7] T. Boshita, H. Suzuki, y Y. Matsumoto, “Compression Method of Position Information for IoT-based Bus Location System Using LoRaWAN”, 2018.
- [8] T. Hadwen, V. Smallbon, y Q. Zhang, “Energy efficient LoRa GPS tracker for dementia patients”, 2017.
- [9] C. A. Odiard, “Sistema de información de apoyo a las decisiones en la agricultura de precisión basado en redes de sensores remotos y herramientas de sistemas de información geográfica (SIG).”, Universidad Siglo 21, 2020.
- [10] W. R. Da Silva, N. K. L. Oliveira, R. A. L. Rabêlo, C. N. M. Marins, y J. J. P. C. Rodrigues, “An Internet of Things Tracking System Approach Based on LoRa Protocol”, 2018.
- [11] L. Dragino Technology Co., “Dragino LoRaWAN - Products”, 2021. <https://www.dragino.com/products/lora-lorawan-gateway.html>.
- [12] Catsensors, “Sensores LoRaWAN - Catsensors”, 2020. <https://www.catsensors.com/es/lorawan/sensores-lorawan>.

- [13] Duolink, “Tecnologías de medición y conectividad”, 2019. <http://www.duolink.com.ar/>.
- [14] Electronic Cats, “CatWan Relay board”, 2019.
- [15] C. BURBANO, “IMPLEMENTACIÓN DE UNA RED DE SENSORES INALAMBRICOS LPWAN MEDIANTE MODULOS LoRa PARA EL MONITOREO DE LA CALIDAD DEL AGUA EN 2 RIOS”, Universidad distrital francisco josé de caldas, Bogotá, 2017.
- [16] J. Triana y R. Rodriguez, “Prototipo de solución iot con tecnología ‘LoRa’ en monitoreo de cultivos agrícolas.”, Universidad distrital francisco josé de caldas, Bogotá, 2018.
- [17] D. CANGREJO y D. M. HERNÁNDEZ, “SISTEMA DE APOYO A LA PREVENCIÓN DEL ABIGEATO DE GANADO BOVINO UTILIZANDO TECNOLOGÍAS IOT Y CLOUD”, Universidad Santo Tomás, Bogotá, 2019.
- [18] L. Rivera, “Solución IoT para la optimización del proceso de piscicultura en el Centro de Desarrollo Agroalimentario El Limonal”, Universidad Santo Tomás, Bogotá, 2020.
- [19] scj, “GRAN ENTREGATÓN DE BICICLETAS RECUPERADAS POR LAS AUTORIDADES”, 2018. [En línea]. Disponible en: <https://scj.gov.co/es/noticias/gran-entregatón-bicicletas-recuperadas-las-autoridades>.
- [20] G. Torrijos, “Cada tres horas es hurtada una bici en Bogotá”, *El Espectador*, 2018, [En línea]. Disponible en: <https://www.elespectador.com/bogota/cada-tres-horas-es-hurtada-una-bici-en-bogota-article-738148/>.
- [21] L. Ambrona, “Sistemas de localización en interiores”, p. 11 , Universidad Politécnica de Madrid, España, 2018.
- [22] F. Acevedo, G. Coduri, y G. Perera, “Geolocalización con LoRa mediante multilateración”, p. 48 , Universidad de república Uruguay, Uruguay, 2018.
- [23] LoRaWAN, “About LoRaWAN™ | LoRa Alliance™”, *What is the LoRaWAN™ Specification?*, 2019. <https://lora-alliance.org/about-lorawan>.
- [24] Lora Alliance, “LoRaWAN 1.1 Regional Parameters”, *Lora Alliance*. D.KJENDAL (Senet), p. 12, 2017.
- [25] Agencia Nacional del Espectro (ANE), *RESOLUCIÓN N° 000711 DEL 11 OCT. 2016*. Colombia, 2016, p. 5.
- [26] Institute of Standards and Technology, *ADVANCED ENCRYPTION STANDARD (AES)*. 2001.

- [27] W. Stallings, "Cryptography and Network Security", 3a ed., 2011, pp. 312–328.
- [28] P. Patil, P. Narayankar, D. G. Narayan, y S. M. Meena, "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish", *Procedia Comput. Sci.*, vol. 78, pp. 617–624, 2016.
- [29] Dragino Technology Co LTD., "Lora/GPS Shield", 2018.
- [30] Dragino Technology Co LTD., "LG01 LoRa Gateway User Manual". 2018, [En línea]. Disponible en: [https://www.dragino.com/downloads/downloads/UserManual/LG01\\_LoRa\\_Gateway\\_User\\_Manual.pdf](https://www.dragino.com/downloads/downloads/UserManual/LG01_LoRa_Gateway_User_Manual.pdf).
- [31] Shawn, "ATmega328P, the one microcontroller you should start with", 2019. <https://www.seeedstudio.com/blog/2019/10/22/atmega328p-the-one-microcontroller-you-should-start-with/>.
- [32] Microchip, "ATmega328P", [En línea]. Disponible en: <https://www.microchip.com/wwwproducts/en/ATmega328P>.
- [33] F. Romero, "Herramienta web para la monitorización de flota de vehiculos", Universitat Oberta de Catalunya, 2019.
- [34] "Satélites del GPS utilizando trilateración". <https://es.vectorhq.com/vector/gps-satellites-trilateration-101240>.
- [35] Quectel Wireless Solutions Co, "Quectel L80", 2017.
- [36] HOPERF ELECTRONIC, *RFM95/96/97/98(W) - Low Power Long Range Transceiver Module V1.0*. pp. 26–28.
- [37] O. G.-A. Roldán, "Estudio De Disponibilidad De Señales De Localización Gps/Gsm", Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona, 2008.
- [38] I. Dopico Fernández, "Comparativa Y Representación En Base a La Precisión De Un Receptor En Combinación Con Diferentes Sistemas De Transmisión Gnss", Universitat Oberta de Catalunya, 2018.
- [39] Y. Hassan, *Experiencia de Usuario: Principios y Métodos*. 2015.
- [40] Google LLC, "Google Maps (Mapas de bogotá)".
- [41] IDECA, "Densidad de Predios. Bogotá D.C. Año 2018", 2018. .
- [42] ORACLE, "What is the Internet of Things (IoT)?", 2021. [tps://www.oracle.com/internet-of-things/what-is-iot/](https://www.oracle.com/internet-of-things/what-is-iot/).

## 12. ANEXOS

### 1. Código cliente nodo LoRa-GPS

```
#define BAUDRATE 9600
#define DOWNLINK_DELAY 4000
#define INPUT_BUFFER_LIMIT (128+1)
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <SPI.h>
#include <RH_RF95.h>
#include <avr/sleep.h>
#include <avr/wdt.h>
#include <AESLib.h>

AESLib aesLib;
TinyGPS gps;
SoftwareSerial ss(3, 4); // Arduino TX, RX
RH_RF95 rf95;

// Constant
float frequency = 914.9;
bool isInit = true;
const char keyNSC[] PROGMEM = "7L21J5PBZIJJ58BJ|";
//const char keyNSC[] PROGMEM = "GYFBUSOPDRBD4R0G|";
volatile byte wdt_counter=0;//Counter for Watch Dog
char cleartext[INPUT_BUFFER_LIMIT];
// AES Encryption Key
byte aes_key[] = { 0xA8, 0x40, 0x41, 0x1B, 0xB2, 0xE8, 0x00, 0x00, 0x00,
0x00, 0xA8, 0x40, 0x41, 0x1B, 0xB2, 0xE8 };

void setup()
{
  ss.begin(BAUDRATE); // SoftSerial port to get GPS data.
  Serial.begin(BAUDRATE);
  if (!rf95.init()){
    isInit = false;
    Serial.println(F("init failed"));
  }
  Serial.println(F("IG!"));
  // Setup ISM frequency
  rf95.setFrequency(frequency);
  // Setup Power, dBm
  rf95.setTxPower(20);

  // Setup Spreading Factor (6 ~ 12)
  rf95.setSpreadingFactor(7);
```

```

// Setup BandWidth, option:
7800,10400,15600,20800,31200,41700,62500,125000,250000,500000
rf95.setSignalBandwidth(125000); //41700

// Setup Coding Rate:5(4/5),6(4/6),7(4/7),8(4/8)
rf95.setCodingRate4(5); //8

setup_watchdog(7); //Set up WatchDog interrupt time
// 0=16ms, 1=32ms,2=64ms,3=128ms,4=250ms,5=500ms
// 6=1 sec,7=2 sec, 8=4 sec, 9=8sec
delay(100);
// Entering Sleeping Mode
ACSR |=_BV(ACD); //OFF ACD
ADCSRA=0; //OFF ADC
rf95.sleep();
Sleep_avr(); //Sleep_Mode, in sleep mode, the current is 22uA@3.3v
}

void loop()
{
  if (wdt_counter >= 15) // 7 interrupt counters then TX, total 28 Seconds
  {
    if(isInit){
      smartdelay(2000);
      long lat, lon;
      gps.get_position(&lat, &lon);
      if(lat != 999999999 && lon != 999999999)
      {
        char dataLora[43];
        sprintf(cleartext, "%ld%ld", lat, lon);
        //sprintf(cleartext, "4638285-74064988");
        Serial.println(cleartext);
        strncpy_P(dataLora, keyNSC, sizeof(keyNSC));
        strcat(dataLora, encrypt_data().c_str());
        uint8_t data[sizeof(dataLora)];

        strncpy(data, dataLora, sizeof(dataLora));
        Serial.println(sizeof(dataLora));
        Serial.println((char *)data);
        rf95.send(data, sizeof(data));
        rf95.waitPacketSent(); // parameter time ms

        // Now wait for a reply
        uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
        uint8_t len = sizeof(buf);

        if (rf95.waitAvailableTimeout(DOWNLINK_DELAY))
        {
          // Should be a reply message for us now
          if (rf95.recv(buf, &len))
          {

```

```

        Serial.print(F("Got reply from LoRa server= "));
        Serial.println((char*)buf);
        Serial.print(F("RSSI: ")); //signal !!!
        Serial.println(rf95.lastRssi(), DEC);
    }
    else
    {
        Serial.println(F("recv failed"));
    }
}
else if (rf95.waitForAvailableTimeout(DOWNLINK_DELAY))
{
    // Should be a reply message for us now
    if (rf95.recv(buf, &len))
    {
        Serial.print(F("Got reply from LoRa server= "));
        Serial.println((char*)buf);
        Serial.print(F("RSSI: ")); //signal !!!
        Serial.println(rf95.lastRssi(), DEC);
    }
    else
    {
        Serial.println(F("recv failed"));
    }
}
else
{
    Serial.println(F("No reply, is LoRa Server running?"));
}

}
else{
    Serial.println(F("No data was sent"));
}
}
wdt_counter = 0; // Reset Counter
delay(DOWNLINK_DELAY);
}
// Entering Sleeping Mode
ACSR |= _BV(ACD); //OFF ACD
ADCSRA=0; //OFF ADC
rf95.sleep();
Sleep_avr(); //Sleep_Mode
}

//----- Recive data GPS. -----//
static void smartdelay(unsigned long ms)
{
    unsigned long start = millis();
    do
    {

```



```

        while (ss.available())
        {
            gps.encode(ss.read());
        }
    } while (millis() - start < ms);
}
//----- Encrypte Data. -----//

static String encrypte_data()
{
    byte enc_iv[N_BLOCK] = { 0x5A, 0x78, 0x50, 0x36, 0x7A, 0x4E, 0x35, 0x72,
0x6C, 0x53, 0x6D, 0x39, 0x37, 0x63, 0x48, 0x6C };
    int msgLen = strlen(cleartext);
    Serial.print("Longitud: ");Serial.print(msgLen);
    char encrypted[2 * msgLen];
    aesLib.encrypt((byte*)cleartext, msgLen, (char*)encrypted, aes_key,
sizeof(aes_key), enc_iv);
    return encrypted;
}
//----- Encrypte Data. -----//

//-----Set up sleep mode time. -----//
void setup_watchdog(int time) {

    byte wdt_time;

    if (time > 9 ) time = 9;
    wdt_time=time & 7;
    if (time > 7) wdt_time |= ( 1 << 5 );
    wdt_time |= ( 1 << WDCE );

    MCUSR &= ~( 1 << WDRF );//Clear WDRF in MCUSR
    WDTCSR |= ( 1 << WDCE) | ( 1 << WDE );

    // set up new watchdog timeout
    WDTCSR = wdt_time;
    WDTCSR |= _BV(WDIE);// close watchdog
}

//WDT interrupt
ISR(WDT_vect) {
    ++wdt_counter;
}

void Sleep_avr(){
    set_sleep_mode(SLEEP_MODE_PWR_DOWN ); // set up sleep mode
    sleep_enable();
    sleep_mode(); // entering sleep mode
    rf95.sleep();
}

```

```
//-----Set up sleep mode time. -----//
```

## 2. Código servidor gateway LoRa

```
#define BAUDRATE 115200

#include <Console.h>
#include <SPI.h>
#include <RH_RF95.h>

// Singleton instance of the radio driver
RH_RF95 rf95;

int led = A2;
float frequency = 915.0;

void setup()
{
  pinMode(led, OUTPUT);
  Bridge.begin(BAUDRATE);
  Console.begin();
  while (!Console) ; // Wait for console port to be available
  Console.println("Start Sketch");
  if (!rf95.init())
    Console.println("init failed");
  // Setup ISM frequency
  rf95.setFrequency(frequency);
  // Setup Power, dBm
  rf95.setTxPower(23);

  // Setup Spreading Factor (6 ~ 12)
  rf95.setSpreadingFactor(7);

  // Setup BandWidth, option:
  7800,10400,15600,20800,31200,41700,62500,125000,250000,500000
  rf95.setSignalBandwidth(41700); // Mas rango entre mas pequeño

  // Setup Coding Rate:5(4/5),6(4/6),7(4/7),8(4/8)
  rf95.setCodingRate4(8); // a mayos coding ratwe aumenta la duracion de
  transmisión, y mejora la interferencia

  Console.print("Listening on frequency: ");
  Console.println(frequency);
}

void loop()
{
  if (rf95.available())
  {
```

```

// Should be a message for us now
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);
if (rf95.recv(buf, &len))
{
    digitalWrite(led, HIGH);
    RH_RF95::printBuffer("request: ", buf, len);
    Console.print("got request: ");
    Console.println((char*)buf);
    Console.print("RSSI: ");
    Console.println(rf95.lastRssi(), DEC);

    // Send a reply
    uint8_t data[] = "And hello back to you";
    rf95.send(data, sizeof(data));
    rf95.waitPacketSent();
    Console.println("Sent a reply");
    digitalWrite(led, LOW);
}
else
{
    Console.println("recv failed");
}
}
}

```

### 3. Función de descifrado en la aplicación web

```

decrypt (dataEncrypted :string): string {
    let iv = CryptoJS.enc.Hex.parse("5A7850367A4E35726C536D393763486C");
    let key = CryptoJS.enc.Hex.parse("A840411BB2E800000000A840411BB2E8");
    var bytes = CryptoJS.AES.decrypt( dataEncrypted, key, {
        iv: iv,
        mode: CryptoJS.mode.CBC,
        padding: CryptoJS.pad.ZeroPadding
    });
    let plaintext = bytes.toString(CryptoJS.enc.Base64);
    let strPlainText = atob(plaintext).replace(/-/g, "-");
    let data = strPlainText.split(",");
    let lat = parseInt(data[0])/1000000;
    let log = parseInt(data[1])/1000000;
    return lat+", "+log;
}

```

#### 4. Dockerfile para la creación de imagen contenerizada

```
#First stage
FROM node:10-alpine as build-step

RUN mkdir -p /app

WORKDIR /app

COPY package.json /app

RUN npm install

COPY . /app

RUN npm run build --prod

#Second stage
FROM nginx:1.17.1-alpine

COPY --from=build-step /app/dist/LoraClient /usr/share/nginx/html
```