

**MONITOREO IOT AL SISTEMA DE BICICLETAS CAMPUS VERDE DE LA  
UNIVERSIDAD DE NARIÑO**



**SEBASTIAN ERASO MISNAZA  
JUAN GUILLERMO ERASO TREJO**

**UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERIA  
PROGRAMA DE INGENIERIA ELECTRONICA  
SAN JUAN DE PASTO**

**2019**

**MONITOREO IOT AL SISTEMA DE BICICLETAS CAMPUS VERDE DE LA  
UNIVERSIDAD DE NARIÑO**



**SEBASTIAN ERASO MISNAZA  
JUAN GUILLERMO ERASO TREJO**

**TRABAJO DE GRADO PARA OPTAR POR EL TITULO DE  
INGENIERO ELECTRÓNICO**

**ASESOR  
M.Sc. DARIO FERNANDO FAJARDO**

**UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERIA  
PROGRAMA DE INGENIERIA ELECTRONICA  
SAN JUAN DE PASTO**

**2019**

## **NOTA DE RESPONSABILIDAD**

"La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor"

Acuerdo 1. Artículo 324. Octubre 11 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

**NOTA DE ACEPTACIÓN:**

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

## **AGRADECIMIENTO**

*Agradecimientos a:*

A los profesores y demás integrantes del Departamento de Ingeniería Electrónica por su constante compromiso en nuestra formación integral como ingenieros.

Msc. Dario Fernando Fajardo, asesor del trabajo de grado por su confianza y apoyo para el desarrollo satisfactorio de este trabajo.

A nuestro compañero Juan David Velasquez, por brindarnos su tiempo y conocimiento que contribuyeron al desarrollo de este proyecto.

## **DEDICATORIA**

*A mi madre por su apoyo incondicional, que me ha brindado la oportunidad de seguir esta profesión, sin ella nada de esto hubiera sido posible.*

*Juan Guillermo*

*A mi familia por su apoyo y por ser la inspiración para seguir adelante y mejorar cada día en mi profesión y como persona.*

*Sebastián*

## **RESUMEN**

En este proyecto se desarrolló un sistema de monitoreo para las bicicletas eléctricas de Campus Verde en la Universidad de Nariño, basado en una red con tecnología LoRa, con el objetivo de registrar diferentes variables como posición, voltaje, corriente, potencia, distancia, energía, emisiones de CO<sub>2</sub>, temperatura y humedad, se implementó un dispositivo que recolecta la información de los sensores y la transmite a través de la tecnología LoRa.

Se creó un servidor para almacenar en una base de datos la información recibida por la red, que funciona a partir de Web-Sockets. En este se procesa la información para poder descargarla y visualizarla de una manera más comprensible y ordenada, a través de diferentes aplicativos web, entre los que encontramos, mapas con la ubicación actual y registro de trayectorias, gráficas de la variación de las diferentes variables sensadas en el tiempo.

Se efectuaron diferentes pruebas para garantizar el funcionamiento de los dispositivos de transmisión y recepción variando las condiciones de ubicación, con el fin de recopilar información que permitirá buscar la mejor ubicación de los dispositivos receptores en la implementación final de la red.

## **ABSTRACT**

In this project, we developed a monitoring system for the electric bicycles of Campus Verde at the University of Nariño, based on a network with LoRa technology implemented by another group. In order to record different variables such as position, voltage, current, power, distance, energy, CO2 emissions, temperature and humidity. We implemented a device that collects the information of the sensors and the transmission through LoRa technology.

We created a server to save in a database the information received by the network. It works from Web-Sockets, in this the information is processed to be able to download and visualize a more understandable and orderly way, through different applications web, among which we find, maps with the current location and record of trajectories, graphs of the variation of the different variables measured over time.

We made different tests to guarantee the operation of the transmission and reception devices varying the location conditions, in order to gather information that will seek the best location of the receiving devices in the final implementation of the network.

## CONTENIDO

|                                                                |    |
|----------------------------------------------------------------|----|
| GLOSARIO.....                                                  | 14 |
| INTRODUCCIÓN.....                                              | 17 |
| Justificación.....                                             | 18 |
| Objetivo general.....                                          | 19 |
| Objetivos específicos.....                                     | 19 |
| Antecedentes .....                                             | 20 |
| Marco Teórico .....                                            | 23 |
| Arquitectura de una red IoT LoRaWAN.....                       | 23 |
| Nodos.....                                                     | 24 |
| Gateway.....                                                   | 24 |
| Servidor Web-Socket.....                                       | 25 |
| 1. DESARROLLO.....                                             | 26 |
| 1.1 Investigación preliminar.....                              | 26 |
| 1.2 Pruebas preliminares.....                                  | 28 |
| 1.3 Unidades de captura.....                                   | 31 |
| 1.3.1 Alimentación.....                                        | 31 |
| 1.3.2 Circuito impreso.....                                    | 33 |
| 1.3.3 Calibración.....                                         | 35 |
| 1.4 Recepción de información.....                              | 38 |
| 1.4.1 Configuración Gateway.....                               | 38 |
| 1.4.2 Aplicación TTN.....                                      | 40 |
| 1.4.3 Codificación de información.....                         | 41 |
| 1.5 Servidor.....                                              | 42 |
| 2. RESULTADOS.....                                             | 49 |
| 2.1 Hardware.....                                              | 49 |
| 2.2 Aplicativos web .....                                      | 50 |
| 2.2.1 Seguimiento en tiempo real.....                          | 53 |
| 2.2.2 Registro de ubicación por día y distancia recorrida..... | 55 |
| 2.2.3 Registro de emisiones de CO2.....                        | 56 |
| 2.2.4 Registro de sensores.....                                | 56 |
| 3. REPORTE DE COBERTURA.....                                   | 60 |
| 4. CONCLUSIONES.....                                           | 70 |
| 5. RECOMENDACIONES.....                                        | 71 |
| BIBLIOGRAFÍA.....                                              | 73 |
| ANEXOS.....                                                    | 75 |

## LISTA DE TABLAS

|                                                             |    |
|-------------------------------------------------------------|----|
| <b>Tabla 1.</b> Comparación redes de comunicación IoT.....  | 19 |
| <b>Tabla 2.</b> Comparación exactitud GPS.....              | 37 |
| <b>Tabla 3.</b> Consumo de unidad de captura.....           | 49 |
| <b>Tabla 4.</b> Prueba de distancia Gateway 1 .....         | 62 |
| <b>Tabla 5.</b> Prueba de distancia Gateway 2 .....         | 62 |
| <b>Tabla 6.</b> Segunda prueba de distancia Gateway 1... .. | 64 |
| <b>Tabla 7.</b> Promedio de paquetes por minuto.....        | 65 |

## LISTA DE FIGURAS

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| <b>Figura 1.</b> Arquitectura red LoRaWAN.....                                   | 23 |
| <b>Figura 2.</b> Diagrama de bloques del sistema desarrollado.....               | 26 |
| <b>Figura 3.</b> Gateways conectados en la red TTN en Colombia.....              | 27 |
| <b>Figura 4.</b> Diagrama de bloques prueba preliminar.....                      | 28 |
| <b>Figura 5.</b> Coordenadas geográficas en servidor ThingSpeak.....             | 28 |
| <b>Figura 6.</b> Indicador de fuerza de la señal recibida (RSSI) de Dragino..... | 29 |
| <b>Figura 7.</b> Recorrido en servidor ThingSpeak.....                           | 30 |
| <b>Figura 8.</b> Recorrido en campus universitario.....                          | 30 |
| <b>Figura 9.</b> Diagrama de bloques unidad de captura.....                      | 31 |
| <b>Figura 10.</b> Conexiones de alimentación.....                                | 32 |
| <b>Figura 11.</b> Esquemático del circuito.....                                  | 33 |
| <b>Figura 12.</b> Componentes del circuito.....                                  | 34 |
| <b>Figura 13.</b> Calibración de sensor CO2.....                                 | 35 |
| <b>Figura 14.</b> Circuito divisor de voltaje.....                               | 36 |
| <b>Figura 15.</b> Comparación GPS.....                                           | 37 |
| <b>Figura 16.</b> Diagrama de bloques recepción y visualización de datos.....    | 38 |
| <b>Figura 17.</b> Conexión del Gateway en TTN.....                               | 39 |
| <b>Figura 18.</b> Método de autenticación ABP.....                               | 40 |
| <b>Figura 19.</b> Keys de autenticación para los nodos ABP .....                 | 41 |
| <b>Figura 20.</b> Trama del paquete de datos (payload).....                      | 41 |
| <b>Figura 21.</b> Reconstrucción del paquete de datos recibido.....              | 41 |
| <b>Figura 22.</b> Formato JSON de la base de datos.....                          | 43 |
| <b>Figura 23.</b> Visualización de base datos en formato JSON.....               | 44 |
| <b>Figura 24.</b> Interfaz aplicativo tracking.....                              | 45 |
| <b>Figura 25.</b> Aplicativo registro de ubicación.....                          | 46 |

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| <b>Figura 26.</b> Aplicativo registro de emisiones CO2.....           | 47 |
| <b>Figura 27.</b> Aplicativo registro de sensores.....                | 48 |
| <b>Figura 28.</b> Caja protectora del dispositivo de captura.....     | 50 |
| <b>Figura 29.</b> Estado de conexión nodos.....                       | 51 |
| <b>Figura 30.</b> Tráfico de datos servidor TTN.....                  | 52 |
| <b>Figura 31.</b> Recepción en servidor desarrollado.....             | 52 |
| <b>Figura 32.</b> Ubicación Gateways prueba 1.....                    | 53 |
| <b>Figura 33.</b> Posición inicial de prueba de tracking.....         | 54 |
| <b>Figura 34.</b> Tracking en tiempo real.....                        | 54 |
| <b>Figura 35.</b> Aplicativo de distancia y trayectoria.....          | 55 |
| <b>Figura 36.</b> Recorrido con emisiones de CO2.....                 | 56 |
| <b>Figura 37.</b> Gráficas de voltaje versus tiempo.....              | 57 |
| <b>Figura 38.</b> Gráfica de corriente versus tiempo.....             | 57 |
| <b>Figura 39.</b> Gráficas de potencia y energía calculadas.....      | 58 |
| <b>Figura 40.</b> Gráfica de temperatura versus tiempo.....           | 58 |
| <b>Figura 41.</b> Gráfica de humedad versus tiempo.....               | 59 |
| <b>Figura 42.</b> Formato de descarga de los datos visualizados.....  | 59 |
| <b>Figura 43.</b> Metadata Gateway.....                               | 60 |
| <b>Figura 44.</b> Cobertura de gateways prueba 1.....                 | 61 |
| <b>Figura 45.</b> RSSI vs Distancia.....                              | 61 |
| <b>Figura 46.</b> Ubicación gateways prueba 2.....                    | 63 |
| <b>Figura 47.</b> Resultados de cobertura prueba 2.....               | 64 |
| <b>Figura 48.</b> Radio de cobertura.....                             | 66 |
| <b>Figura 49.</b> Ubicación de Gateways en campus universitarios..... | 67 |
| <b>Figura 50.</b> Recorrido por campus universitarios.....            | 68 |
| <b>Figura 51.</b> Regresión lineal de potencia recibida.....          | 69 |

## LISTA DE ANEXOS

|                                                                                        |    |
|----------------------------------------------------------------------------------------|----|
| <b>Anexo1.</b> Especificaciones técnicas de los equipos utilizados en el proyecto..... | 75 |
| 1.1 Dragino Ig01-p.....                                                                | 75 |
| 1.2 WiFi LoRa 32.....                                                                  | 76 |
| 1.3 RAK2245 Pi HAT Edition.....                                                        | 77 |
| 1.4 Antena omnidireccional LoRa de fibra de vidrio 806-960MHz.....                     | 79 |
| 1.5 GPS NEO 6M.....                                                                    | 80 |
| 1.6 TTGO t-beam ESP32.....                                                             | 81 |
| 1.7 CONVERTOR REDUCTOR BUCK DC-DC.....                                                 | 83 |
| 1.8 Xiaomi Banco de energía 20000mAh.....                                              | 84 |
| 1.9 ACS711 Current Sensor Carrier.....                                                 | 85 |
| 1.10 Sensor de CO <sub>2</sub> – SCD30.....                                            | 86 |
| 1.11 Temtop M2000.....                                                                 | 87 |
| <b>Anexo 2.</b> Código de adquisición de datos.....                                    | 88 |
| <b>Anexo 3.</b> Menú principal aplicativo web.....                                     | 89 |

## GLOSARIO

IoT: Se trata de una red que interconecta objetos físicos valiéndose del Internet. Los mencionados objetos se valen de sistemas embebidos, o lo que es lo mismo, hardware especializado que le permite no solo la conectividad a Internet, sino que además programa eventos específicos en función de las tareas que le sean dictadas remotamente.<sup>1</sup>

LORA: Es una tecnología inalámbrica al igual que WiFi, Bluetooth, LTE, SigFox o Zigbee. LoRa utiliza un tipo de modulación en radiofrecuencia, como la AM o la FM o el PSK; pero patentado por Semtech una importante empresa fabricante de chips de radio. Esta tecnología de modulación se llama Chirp Spread Spectrum, o CSS, y se usa en comunicaciones militares y espaciales desde hace décadas. La gran ventaja de la misma es que puede lograr comunicaciones a largas distancias (típicamente kilómetros) y tiene gran solidez frente a las interferencias.<sup>2</sup>

LORA-WAN: Define el protocolo de comunicación y la arquitectura del sistema para la red, mientras que la capa física LoRa permite el enlace de comunicación de largo alcance. LoRaWAN también es responsable de administrar las frecuencias de comunicación, la velocidad de datos y la potencia de todos los dispositivos.

GATEWAY: La puerta de enlace es el dispositivo que actúa de interfaz de conexión entre aparatos o dispositivos, y también posibilita compartir recursos entre dos o más computadoras. Su propósito es traducir la información del protocolo utilizado en una red inicial, al protocolo usado en la red de destino.<sup>3</sup>

GPS: El Sistema de Posicionamiento Global, es un sistema que permite determinar en toda la Tierra la posición de cualquier objeto, persona o vehículo con una precisión de hasta centímetros, aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y empleado por el Departamento de Defensa de los EE. UU. Para determinar su posición, un usuario utiliza 4 o más satélites y utiliza la trilateración.<sup>4</sup>

---

<sup>1</sup> RIVERA, Nicolas. "Qué es el Internet of Things y cómo cambiará nuestra vida en el futuro". {En línea}. {14 marzo de 2019} disponible en: <https://hipertextual.com/2015/06/internet-of-things>.

<sup>2</sup> TECNOLOGIA LORA Y LORAWAN: Tecnología LoRa y LoRaWAN {en línea}. Catsensors. {15 marzo de 2019} disponible en: <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan>.

<sup>3</sup> CONCEPTOS BÁSICOS DE REDES: Términos de la unidad {en línea}. Quizlet. {15 marzo de 2019} disponible en: <https://quizlet.com/es/249490101/conceptos-basicos-de-redes-flash-cards/>

<sup>4</sup> GIMENEZ, Tamara y ROS, María. "SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)". {En línea}. {15 marzo de 2019} disponible en: [https://webs.um.es/bussos/GPSresumen\\_TamaraElena.pdf](https://webs.um.es/bussos/GPSresumen_TamaraElena.pdf)

**SISTEMA EN TIEMPO REAL:** Un sistema en tiempo real (STR) es aquel sistema digital que interactúa activamente con un entorno con dinámica conocida en relación con sus entradas, salidas y restricciones temporales, para darle un correcto funcionamiento de acuerdo con los conceptos de predictibilidad, estabilidad, y controlabilidad. Los STR se pueden encontrar en lugares muy importantes debido a los servicios que prestan: ellos monitorizan, controlan y protegen, por ejemplo, los sistemas de transmisión y distribución que hacen llegar la energía eléctrica a las industrias y también a nuestros hogares. Los STR están presentes en las áreas de monitoreo de tráfico de trenes; su importancia es relevante debido a que diariamente se transportan millones de pasajeros.<sup>5</sup>

**BASE DATOS:** Una base de datos es un conjunto de información, organizada sistemáticamente de tal forma que sea posible consultarla para análisis o transmisión. Entre las bases de datos convencionales encontramos bibliotecas, guías telefónicas, empresas de salud. Pero actualmente las bases de datos se manejan digitalmente, lo que se conoce como sistema de gestión de bases de datos el cual agiliza el acceso a la información.

**SERVIDOR WEB:** La principal función de un servidor Web es almacenar los archivos de un sitio y emitirlos por Internet para poder ser visitado por los usuarios. Básicamente, un servidor Web es una gran computadora que guarda y transmite datos vía el sistema de redes llamado Internet. Cuando un usuario entra en una página de Internet, su navegador se comunica con el servidor enviando y recibiendo datos que determinan qué es lo que ve en la pantalla. Por eso, decimos que los servidores Web están para almacenar y transmitir datos de un sitio según lo solicita el navegador de un visitante.<sup>6</sup>

**TTN (The Things Network):** Es una infraestructura abierta de Internet de las Cosas respaldada por sus miembros. Los miembros contribuyen colocando puertas de enlace o ejecutando servidores de red. Juntos creamos una red colaborativa segura y redundante. The Things Network está creciendo hacia una red global robusta y estable, que proporciona conectividad donde se necesita.<sup>7</sup>

**THINGSPEAK:** Es un servicio de plataforma de análisis de IoT que le permite agregar, visualizar y analizar flujos de datos en vivo en la nube. Puede enviar datos a ThingSpeak desde sus dispositivos, crear visualizaciones instantáneas de datos en vivo y enviar alertas utilizando servicios web como Twitter y Twilio.

---

<sup>5</sup> ELVIR, Dilcia. "SISTEMAS DE TIEMPO REAL". {En línea}. {15 marzo de 2019} disponible en: <https://docs.google.com/document/edit?id=1QWdZfVdJ9WndrUfces9AqVAvZTtZOHpVUegPeql3SRQ&hl=en>

<sup>6</sup> DELGADO, César. "Servidor Web". {En línea}. {15 marzo de 2019} disponible en: [https://upanama.edu.ec/archivos/repositorio/6000/6126/html/59\\_servi.htm](https://upanama.edu.ec/archivos/repositorio/6000/6126/html/59_servi.htm)

<sup>7</sup> SUPPORT: THE THINGS NETWORK SUPPORT. {En línea}. Thethingsnetwork. {15 marzo de 2019} disponible en: <https://www.thethingsnetwork.org/support>

Con el análisis MATLAB dentro de ThingSpeak, puede escribir y ejecutar código MATLAB para realizar pre procesamiento, visualizaciones y análisis. ThingSpeak permite a los ingenieros y científicos crear prototipos.<sup>8</sup>

---

<sup>8</sup> MATHWORKS: ThingSpeak. {En línea}. Mathworks. {15 marzo de 2019} disponible en <https://la.mathworks.com/help/thingspeak/>

## INTRODUCCIÓN

En el desarrollo tecnológico, la continua búsqueda por optimizar procedimientos, ha creado la necesidad de almacenar diferentes variables, desde objetos cotidianos hasta procesos complejos del sector industrial. Para esto se necesita la implementación de un sistema de monitoreo capaz de realizar seguimiento de manera remota a diferentes procesos, con el fin de analizar esta información y tomar decisiones para mejorar la eficiencia de los mismos.

Una de las principales razones por las cuales los sistemas de monitoreo no avanzan en el campo y en la ciudad se debe a que implican el pago recurrente de un costo por el servicio de transferencia de datos. De experiencias pasadas en la Universidad de Nariño encontramos sistemas de monitoreo en las granjas de Nariño Vive Digital, estaciones de variables ambientales para diagnóstico energético de ALTERNAR, se tiene como lección aprendida la búsqueda de una alternativa de comunicación que evite este costo recurrente que hace que los desarrollos no perduren a lo largo del tiempo.

Diferentes investigadores y la industria de servicios de comunicación digital hacen esfuerzos por lograr redes de comunicación robustas que ayuden a cerrar esta brecha tecnológica. Entre diferentes opciones se encuentra el protocolo de comunicación en radiofrecuencia LoRa (Long Range), que es patentado por Semtech y cuenta con un amplio conjunto de ventajas como la alta tolerancia a las interferencias, alta sensibilidad es sus transceivers (-137dBm), bajo consumo de energía, un gran alcance de hasta 20 km con línea de vista y baja transferencia de datos (hasta 255 bytes/s).

En la última década se presentó un gran auge en el desarrollo de nuevas tecnologías para redes privadas de comunicación inalámbricas, tales como zigbee, LoRa, Sigfox, etc. Todo esto, hace a esta tecnología ideal para conexiones a grandes distancias con baja demanda de datos, así como para redes IoT en ciudades inteligentes y específicamente para redes privadas de sensores o actuadores.

En este proyecto se desarrolló una red privada basada en esta tecnología para prestar servicio a un conjunto de bicicletas que actualmente posee la Universidad de Nariño y que posiblemente sea ampliado a otras universidades. Se obtuvo el registro de la ubicación de una bicicleta, tiempo de uso, distancias y trayectorias recorridas, esfuerzos realizados, ahorro energético, emisiones asociadas a un medio de transporte, entre otros; para estimar proyecciones de demanda futura y

permitir la visualización de estas variables en la red mencionada. El trabajo se realizó en el siguiente orden: se comenzó con una revisión bibliográfica y pruebas preliminares definiendo conceptos y la arquitectura de la red, posteriormente la configuración del Gateway y el desarrollo de las unidades de captura para enviar información. Se crea el servidor que recibe y procesa la información para visualizarla en los aplicativos web, finalmente con los datos almacenados se realiza un reporte de resultados de cobertura y otras características de la red.

**Modalidad:** Investigación.

**Línea:** Sistemas de comunicaciones.

Este trabajo es de carácter investigativo debido a que se evalúa el funcionamiento de una nueva tecnología de redes de comunicación de bajo consumo, basados en métodos gráficos y estadísticos aplicados a los datos que fueron almacenados a través de los dispositivos implementados en este proyecto.

### **Justificación**

El registro de información es una tarea en constante desarrollo para ingenieros e investigadores, localmente el grupo de investigación cuenta con diferentes experiencias en el tratamiento de información proveniente de las granjas digitales, las estaciones de monitoreo de variables ambientales para diagnóstico energético de ALTERNAR y el seguimiento de bicicletas eléctricas de Campus Verde. Como conclusión y tarea aprendida, se identifica la necesidad de una alternativa de comunicación que no implique el pago recurrente por el servicio de transferencia de datos.

Existen diferentes opciones en cuanto a redes de comunicación para la ejecución de proyectos IoT en ciudades, se diferencian por sus características de consumo de energía, tasa de datos, ancho de banda, usabilidad entre otras. Estas pueden describirse de acuerdo a la información de la Tabla 1:

**Tabla 1.** Comparación redes de comunicación IoT

| Tecnología | Consumo en los terminales | Alcance  | Madurez  | Disponibilidad | Uso   | Tasa de Datos |
|------------|---------------------------|----------|----------|----------------|-------|---------------|
| GSM/GPRS   | Muy alto                  | Alto     | Muy alto | Muy alto       | Alta  | Alta          |
| SigFox     | Bajo                      | Medio    | Alto     | Medio          | Alta  | Muy baja      |
| LoRa       | Bajo                      | Medio    | Bajo     | Bajo           | Baja  | Muy baja      |
| WiFi       | Alto                      | Bajo     | Muy alto | Alto           | Alta  | Muy alta      |
| BLE        | Muy bajo                  | Muy bajo | Alto     | Bajo           | Media | Baja          |
| ZigBee     | Medio                     | Bajo     | Medio    | Muy bajo       | Baja  | Baja          |

**Fuente:** Adaptación de

<https://www.esmartcity.es/comunicaciones/comunicacion-gestion-inteligente-residuos-urbanos-tecnologia-narrow-band-iot-nb-iot>

De acuerdo a la aplicación objetivo que es un monitoreo constante, se hizo necesaria un tipo de comunicación con bajo consumo de energía y un alcance medio sobre el perímetro de la ciudad. La tasa de datos no fue altamente exigente al tratarse del monitoreo de algunas señales y la ubicación geográfica con una tasa de muestreo para el envío de paquetes de información de 10 a 20 segundos.

Después del análisis entre las opciones anteriormente descritas se seleccionaron de acuerdo a la pertinencia de la aplicación las tecnologías de SigFox, LoRa y GSM/GPRS. Entre las dos opciones novedosas LoRa y SigFox se eligió LoRa, teniendo en cuenta que SigFox es de características de protocolo cerrado a diferencia de LoRa que permite desarrollos de protocolo abierto.

### Objetivo general

Desarrollar una plataforma IoT con bajo consumo de energía y mediano alcance, que permita monitorear y registrar variables físicas medidas sobre las bicicletas de Campus Verde, en sectores de la ciudad de San Juan de Pasto.

### Objetivos específicos

- Desarrollar una comunicación con *Web-Sockets* que recepcione las variables sensadas y posibles variables futuras para emitirlas a internet.

- Construir la base de datos a partir de mediciones reales sobre dispositivos integrados a un conjunto de bicicletas en la Universidad de Nariño.
- Diseñar una aplicación que permita visualizar las señales de interés en la red, generando estadísticas que permitan evaluar el comportamiento con respecto a distancia y consumo.

## Antecedentes

El Departamento de Electrónica y en especial el Grupo de Investigación en Ingeniería Eléctrica y Electrónica (GIIEE) de la Universidad de Nariño ha venido trabajando en el área de automatización de granjas, fertirriego, riego por goteo y control de variables ambientales. Entre estos trabajos se encuentran:

En el año 2012 en la Universidad de Nariño se realiza como trabajo de investigación “Diseño e implementación de una red de sensores inalámbricos para el control de riegos por goteo”<sup>9</sup> por los estudiantes Nesly Diana Castro Castro y Luis Eduardo Chamorro Fernández bajo la dirección del docente Carlos Viteri, adscritos al Departamento de Electrónica de la Universidad de Nariño. Esta experiencia consistió en el diseño e implementación de una red de sensores inalámbricos que monitorean la variable fundamental para controlar el riego por goteo usado en cultivos de fresa. De acuerdo a la investigación realizada, pudieron ser detectados los diferentes requerimientos que debía tener la red, como lo son su forma de transmisión y las variables evaluadas para el control, siendo la variable más importante, la humedad volumétrica del suelo.

En el 2013 en la Universidad de Nariño se realiza la investigación “Diseño e implementación de un dispositivo portátil para el control de humedad en cultivo de maíz fertirrigado bajo el invernadero de la facultad de ciencias agrícolas de la Universidad de Nariño”<sup>10</sup>, por los estudiantes Romer Orlando Pantoja Cuasquen y Luis Alberto Burbano Pantoja. En este desarrollo se controló la variable humedad del suelo en un cultivo de maíz, aplicando fertirriego controlado por un sistema automático, previo estudio de los niveles de humedad sugeridos para este cultivo. En esta investigación conjunta entre el Departamento de Electrónica y la Facultad

---

<sup>9</sup> CASTRO, Nesly; CHAMORRO, Luis y VITERI, Carlos. Diseño e implementación de una red de sensores inalámbricos para el control de riegos por goteo. Pasto, 2012. Trabajo de grado (ingeniería electrónica). Universidad de Nariño. Facultad de Ingeniería. Departamento de Electrónica.

<sup>10</sup> BURBANO, Luis; PANTOJA, Romer y FAJARDO, Dario. Diseño e implementación de un dispositivo portátil para el control de humedad en cultivo de maíz fertirrigado bajo el invernadero de la facultad de ciencias agrícolas de la Universidad de Nariño. Pasto, 2013. Trabajo de grado (ingeniería electrónica). Universidad de Nariño. Facultad de Ingeniería. Departamento de Electrónica.

de Agronomía de la Universidad de Nariño, se realizó la implementación de un dispositivo portátil capaz de medir y controlar los cambios de humedad que se encuentran en el suelo, a través de las diferentes etapas de desarrollo de un cultivo de maíz bajo invernadero, con la capacidad de regar la cantidad de agua necesaria según el cultivo lo requería evitando su desperdicio. Se pudo evidenciar el ahorro del agua como aporte al medio ambiente y el aumento del forraje en los cultivos de maíz en el invernadero de la Universidad de Nariño.

En 2018 en la Universidad de Nariño se realiza la investigación “Implementación de una granja digital mediante el internet de las cosas IoT<sup>11</sup>, por los estudiantes David Felipe Guerrero Díaz y Fabio Rigoberto Rubio Benavides. En esta propuesta de trabajo se implementó un sistema de riego por goteo en una granja regional ubicada en la Institución Educativa Municipal Obonuco (Pasto, Nariño), capaz de optimizar el gasto de recurso hídrico en el momento de la distribución en el cultivo, en respuesta a un algoritmo de control que se alimenta de la información entregada por la medición de variables ambientales como humedad de suelo y temperatura, entre otros. Todo esto monitoreado y controlado desde una aplicación móvil o una página web, diseñada exclusivamente para contribuir en la formación académica de los estudiantes. Como parte del proceso de apropiación social del conocimiento se instruye y desarrolla una aplicación para móvil con estudiantes de bachillerato de la institución educativa, esta aplicación permite monitorear y controlar el riego de forma manual y automática en actividades de clase y el cultivo.

En 2018 en Cork Institute of Technology se realizó la investigación “Analyzing LoRa: a Use Case Perspective”<sup>12</sup>. En este trabajo desarrollado por Muhammad Omer Farooq y Dirk Pesch, se analiza el impacto de la configuración de comunicación de LoRa en cuatro casos de uso de IoT: medición inteligente, estacionamiento inteligente, alumbrado público inteligente y seguimiento de la flota de vehículos. Llegando a conclusiones importantes como que con la configuración correspondiente a la velocidad de datos más rápida alcanza una relación de entrega de paquetes hasta 380% más alta y usa 0.004 veces la energía en comparación con otras configuraciones, pero cubre un área de comunicación más pequeña.

En 2017 en National Chung-Cheng University se realizó la investigación “Demo Abstract: A LoRa Wireless Mesh Networking Module for Campus-Scale

---

<sup>11</sup> GUERRERO, Felipe; RUBIO, Fabio y FAJARDO, Dario. Implementación de una granja digital mediante el internet de las cosas IoT. Pasto, 2018. Trabajo de grado (ingeniería electrónica). Universidad de Nariño. Facultad de Ingeniería. Departamento de Electrónica.

<sup>12</sup> MUHAMMAD, Omer y DIRK, Pesch. Analyzing LoRa: A Use Case Perspective. En: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). Vol.; 4. (Feb.2018); p. 355-360.

Monitoring”<sup>13</sup>, por Kai-Hsiang Ke, Qi-Wen Liang, Guan-Jie Zeng, Jun-Han Lin y Huang-Chen Lee. En esta investigación se desarrolló 19 dispositivos LoRa distribuidos en un campus de 800m por 600m, para recolectar datos en intervalos de un minuto.

En 2018 en Monash University se realizó la investigación”WE-Safe: A Wearable IoT Sensor Node for Safety Applications via LoRa”<sup>14</sup>, por Fan Wu., Christoph Rudiger., Jean-Michel Redoute., Mehmet Rasit Yuce. En esta investigación se desarrolló un sistema de alerta de condiciones ambientales nocivas, monitoreando principalmente monóxido de carbono, dióxido de carbono y radiación ultravioleta a través de la tecnología LoRa.

En el 2018 en Electronics and Telecommunications Research Institute, se realizó la investigación “LoRa based Renewable Energy Monitoring System with Open IoT Platform”<sup>15</sup>, por Chang-Sic Choi, Jin-Doo Jeong, Il-Woo Lee y Wan-Ki Park. En este documento, se describe la implementación del sistema de monitoreo para las instalaciones de generación de energía renovable con la arquitectura del sistema, el método de implementación y el programa de análisis. Utilizando varias plataformas abiertas de IoT, como Arduino, Raspberry Pi y una red LoRa de bajo costo.

En el año 2018 en la Universidad Distrital Francisco Jose de Caldas se publica la tesis “Prototipo de solución IoT con tecnología “lora” en monitoreo de cultivos agrícolas”<sup>16</sup>, presentada por Jordan Camilo Triana Useche y Ronald Estiven Rodriguez Leguizamo. La cual consiste en el desarrollo teórico-práctico de un prototipo de solución IoT para el monitoreo de cultivos agrícolas, implementando una de las nuevas tecnologías en el sector IoT “LoRa”, que cuenta con un bajo consumo de energía y larga distancia (mayor a 8 km en línea de vista) entre el dispositivo transmisor y el dispositivo receptor, con el fin de conocer las variables físicas que afectan directamente al proceso de un cultivo a través de un aplicativo web.

---

<sup>13</sup> HSIANG, Kai; WEN, Qi; JIE, Guan, HAN, Jun y CHEN, Huang. Demo Abstract: A LoRa Wireless Mesh Networking Module for Campus-Scale Monitoring. En: 2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). Vol.; 16. (Abr.2017); p. 259-260.

<sup>14</sup> WU, Fan; RUDIGER, Christoph; REDOUTE, Jean-Michel y YUCE, Mehmet. WE-Safe: A Wearable IoT Sensor Node for Safety Applications via LoRa. En: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). Vol.; 4. (Feb.2018); p. 144-148.

<sup>15</sup> CHOI, Chang-Sic; JEONG Jin-Doo; LEE, Il-Woo y PARK, Wan-Ki. LoRa based Renewable Energy Monitoring System with Open IoT Platform. En: 2018 International Conference on Electronics, Information, and Communication (ICEIC). Vol.; 1. (Ene.2018); p. 184-185.

<sup>16</sup> TRIANA, Jordan y RODRIGUEZ, Ronald. Prototipo de solución IoT con tecnología “lora” en monitoreo de cultivos agrícolas. Bogotá, 2018. Trabajo de grado (ingeniería en telecomunicaciones). Universidad Distrital Francisco José de Caldas. Facultad de Tecnología. Departamento de Telecomunicaciones.

## Marco teórico

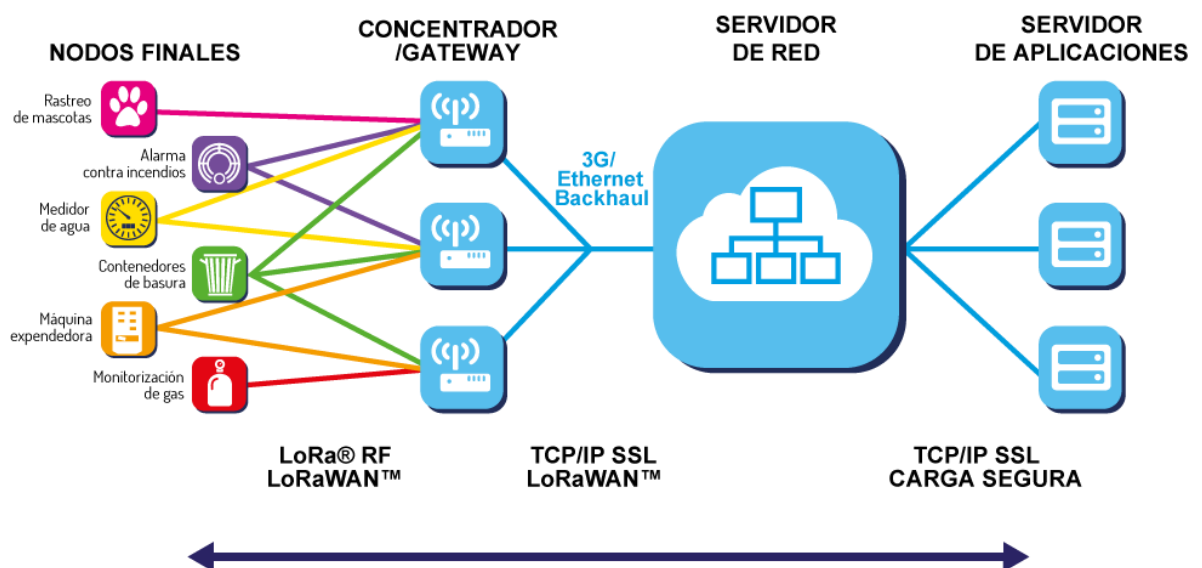
En esta sección se explica la forma general de una red LoRaWAN, describiendo sus componentes para entender su funcionamiento.

### Arquitectura de una red IoT LoRaWAN

Por medio del IoT se puede controlar y detectar sistemas de manera inalámbrica, a través de una red de comunicación. Facilita la integración de variables físicas con sistemas de computación, mejorando la eficiencia, precisión y reducción de costos. Entre las aplicaciones más comunes se encuentran:

- Gestión de tráfico y un sistema de alerta para diferentes situaciones que se presentan en las vías.
- Monitoreo de redes eléctricas para mejorar su eficiencia
- Automatización en actividades domésticas, conocido como domótica.
- Estaciones meteorológicas.

**Figura 1.** Arquitectura red LoRaWAN.



**Fuente:** [https://deepdata.es/wp-content/uploads/2018/12/infograf%C3%ADa\\_red.png](https://deepdata.es/wp-content/uploads/2018/12/infograf%C3%ADa_red.png)

Alguna de las características de la tecnología LoRa son<sup>17</sup>, tecnología inalámbrica de bajo consumo, sensibilidad hasta -146dBm, potencia de emisión hasta 20dBm, enlaces de 166dBm, comunicación bidireccional, tamaño máximo de 256 bytes de información, trabaja con protocolos IPV6, alta inmunidad a interferencias, bajo consumo 10 mA con modos de recepción en “Low dutty cycle” – Rango de frecuencia [137 – 960]MHz.

## **Nodos**

Los dispositivos en la red son asíncronos y se transmiten cuando tienen datos disponibles para enviar. Los datos transmitidos por un dispositivo de nodo final son recibidos por múltiples puertas de enlace, que envían los paquetes de datos a un servidor de red centralizado. El servidor de red filtra paquetes duplicados, realiza comprobaciones de seguridad y administra la red. Los datos se envían a los servidores de aplicaciones. La tecnología muestra una alta confiabilidad para la carga moderada, sin embargo, tiene algunos problemas de rendimiento relacionados con el envío de acuses de recibo.

## **Gateway**

En las redes LoRaWan un Gateway es un dispositivo dentro de la arquitectura de red que recibe los datos transmitidos por un dispositivo de nodo final y que reenvían los paquetes de datos a un servidor de red centralizado. Los gateways son enrutadores equipados con un concentrador LoRa, lo que les permite recibir paquetes LoRa. Por lo general, puede encontrar dos tipos de puertas de enlace:

- Las pasarelas se ejecutan con un firmware mínimo, por lo que son de bajo costo y fáciles de usar (por ejemplo, The Things Gateway) y solo ejecutan el software de reenvío de paquetes.
- Gateways que ejecutan un sistema operativo, que posee un software de reenvío de paquetes que se ejecuta como un programa de fondo (por ejemplo, Kerlink IoT Station, Multitech Conduit). Esto le da más libertad al administrador del Gateway para administrar su puerta de enlace e instalar su propio software.

---

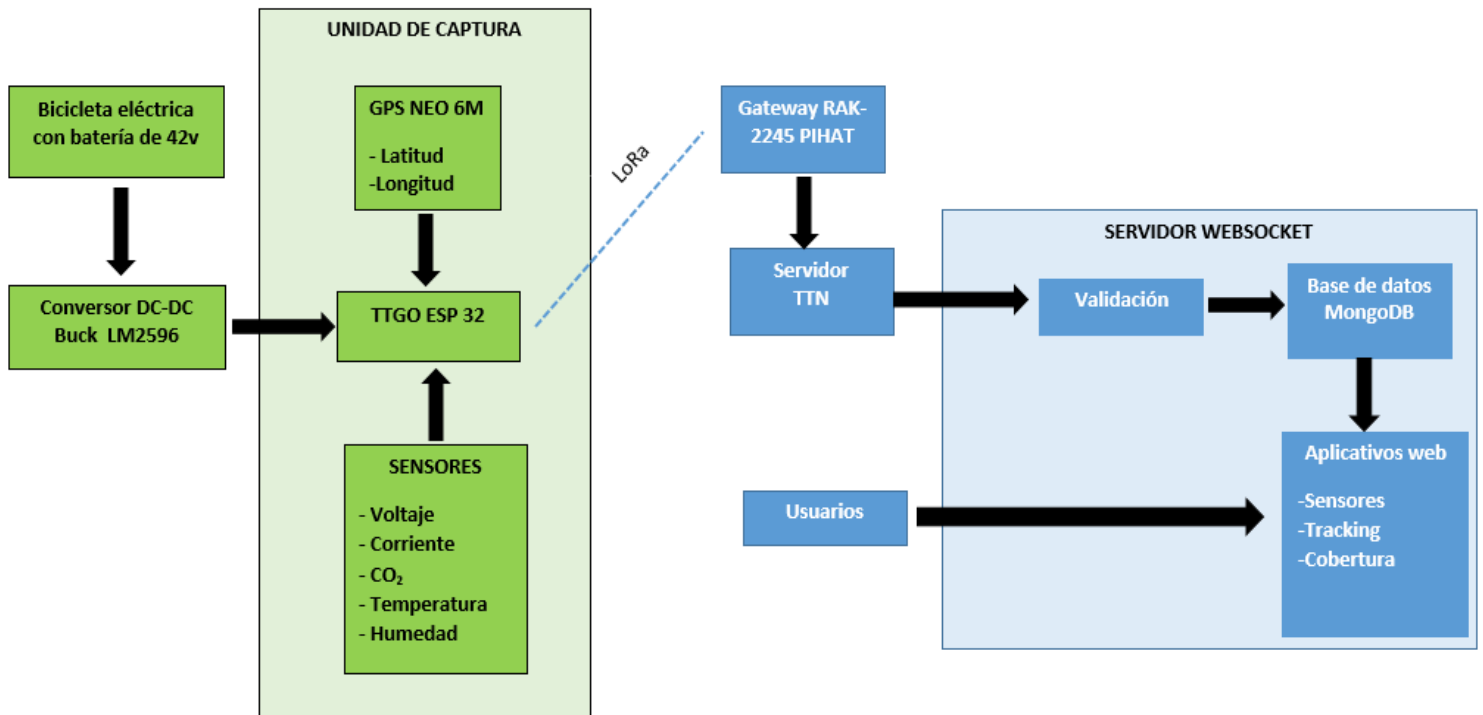
<sup>17</sup> LORAWAN: LoRaWAN Architecture. {En línea}. Thethingsnetwork. {15 marzo de 2019} disponible en: <https://www.thethingsnetwork.org/docs/lorawan/architecture.html>

## **Servidor Web-Socket**

Consiste en una tecnología que permite una comunicación interactiva entre el servidor y un usuario. A través de los cuales se envía mensajes a un servidor y se recibe respuestas por medio de eventos sin necesidad de consultar al servidor. Por tanto, un servidor *Web-Socket* es una aplicación TCP que a través de un protocolo específico se escucha en cualquier puerto de un servidor, el cual puede ser desarrollado en diversos lenguajes de programación como C++, Python, PHP y JavaScript.

## 1. DESARROLLO

En la Figura 2 se muestra el diagrama de bloques de todo el sistema de monitoreo implementado, que contiene la parte de adquisición, transmisión, recepción y visualización de datos.



**Figura 2.** Diagrama de bloques del sistema desarrollado.

**Fuente:** Elaboración propia.

### 1.1 Investigación Preliminar

En el contexto del internet de las cosas IoT se presentan continuamente proyectos a diferente escala, particularmente en Colombia se plantea por parte del ministerio TIC una subasta del espectro en las bandas de, 700MHz, 1900 MHz y 2500MHz<sup>18</sup>

<sup>18</sup> ARANGO, María. "Lista la resolución para la subasta del espectro en Colombia". [En línea]. {18 octubre de 2019} disponible en: <https://www.eltiempo.com/tecnosfera/novedades-tecnologia/mintic-publica-resolucion-para-la-subasta-del-espectro-radioelectrico-421954>

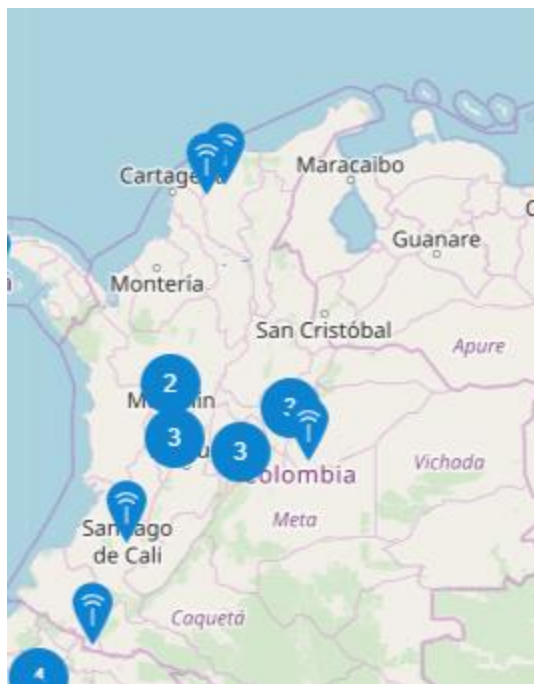
que serían utilizadas en redes de transmisión de datos de bajo consumo, que fomenta el crecimiento de estas redes de datos.

Se parte de un estudio de los conceptos más importantes, de fuentes como artículos, videos, trabajos de grado e internet, principalmente en el desarrollo de servidores web y aplicación de dispositivos LoRa, que son el punto de partida para desarrollar este proyecto.

Para el desarrollo del servidor web se tuvo en cuenta el libro Aprendiendo JavaScript<sup>19</sup>, que contiene un tutorial de cómo crear un servidor con *Web-Sockets* y conectarse a él para una aplicación de chat. Es importante ya que menciona las librerías e instrucciones básicas que definen la estructura del servidor.

La plataforma de tecnología de comunicación LoRa más grande es The Things Network [8] que cuenta con 90653 miembros, 9623 gateways distribuidos en 147 países. Particularmente en Colombia se cuenta con puntos en las ciudades de Barranquilla, Bogotá, Boyacá, Cartagena, Manizales, Medellín.

**Figura 3.** Gateways conectados en la red TTN en Colombia.



**Fuente:** <https://www.thethingsnetwork.org>

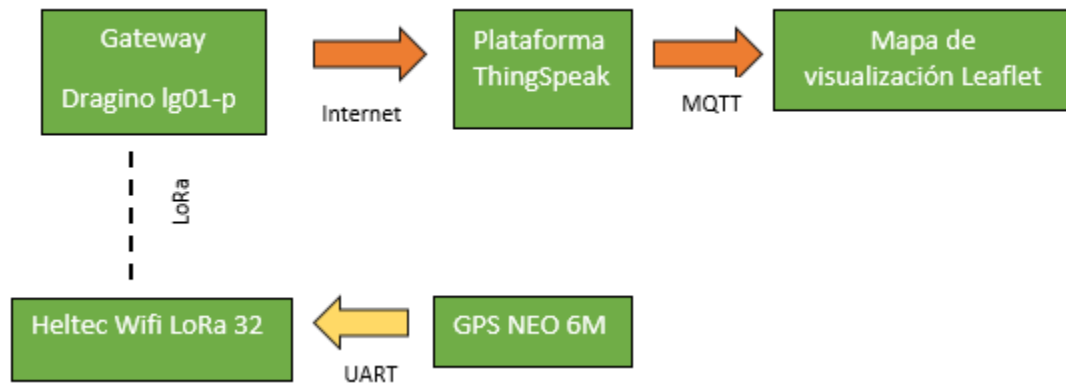
---

<sup>19</sup> AZAUSTRE, Carlos. Aprendiendo JavaScript. Madrid: abril de 2016. {18 octubre de 2018} disponible en: [https://play.google.com/store/books/details?id=cnjhCwAAQBAJ&rdid=book-cnjhCwAAQBAJ&rdot=1&source=gbs\\_vpt\\_read&pcampaignid=books\\_booksearch\\_viewport](https://play.google.com/store/books/details?id=cnjhCwAAQBAJ&rdid=book-cnjhCwAAQBAJ&rdot=1&source=gbs_vpt_read&pcampaignid=books_booksearch_viewport)

## 1.2 Pruebas preliminares

En la Figura 4 se presenta el esquema de los dispositivos y aplicaciones utilizados para realizar las pruebas preliminares.

**Figura 4.** Diagrama de bloques prueba preliminar.



**Fuente:** Elaboración propia.

Con el fin de conocer y aplicar la tecnología LoRa, se parte con equipos básicos y de fácil acceso, estableciendo una primera conexión entre el Gateway Dragino y el nodo Heltec Wifi LoRa 32 enviando coordenadas de posición, las cuales son visualizadas en una aplicación IoT de la plataforma de ThingSpeak, donde en primera instancia se observó las variables en forma numérica, como se observa en la Figura 5 y Figura 6. Las especificaciones técnicas de los equipos utilizados se encuentran en los anexos 1.1 y 1.2.

**Figura 5.** Coordenadas geográficas en servidor ThingSpeak.



**Fuente:** <https://thingspeak.com/channels/774981>

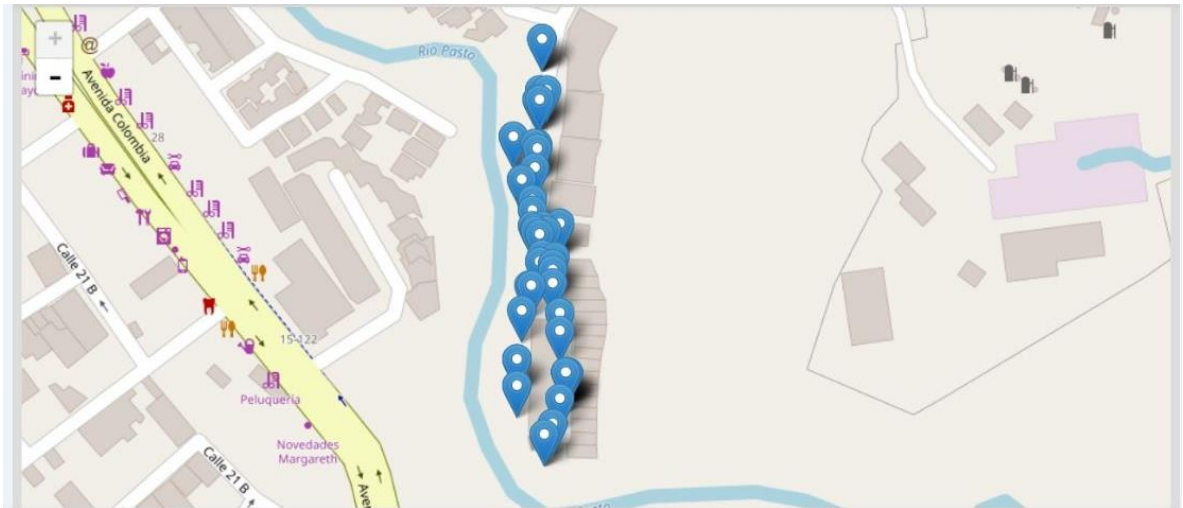
**Figura 6.** Indicador de fuerza de la señal recibida (RSSI) de Dragino.



**Fuente:** <https://thingspeak.com/channels/774981>

Se conoce que el aplicativo de mapeo web más utilizado es Google Maps, pero para una aplicación de seguimiento presenta limitaciones en cantidad de actualizaciones en un tiempo determinado, por este motivo, su representación en el mapa se desarrolla con ayuda de la biblioteca para JavaScript Leaflet de aplicaciones en mapeo web, implementada en el *plugin* de ThingSpeak que permite programar en código abierto. Se realizó las pruebas en los sectores de Avenida Colombia y Universidad de Nariño, obteniendo los siguientes resultados en la Figura 7 y Figura 8 respectivamente, donde se observa que el alcance de cobertura obtenido con estos equipos es muy limitado, esto hizo necesario implementar equipos con mejor potencia de recepción, para mejorar la cobertura en sectores de la ciudad.

**Figura 7.** Recorrido en servidor ThingSpeak.



**Fuente:** <https://thingspeak.com/apps/plugins/298935>

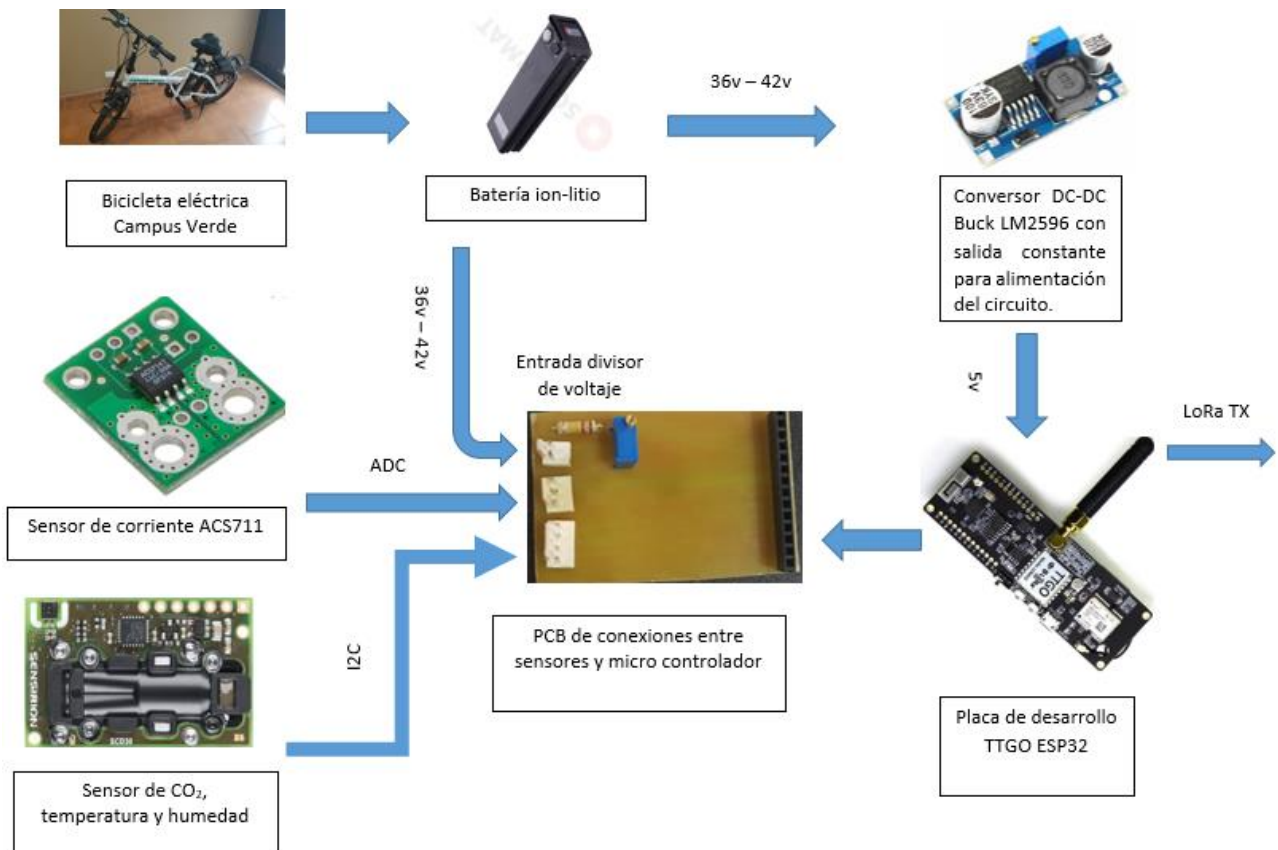
**Figura 8.** Recorrido en campus universitario.



**Fuente:** <https://thingspeak.com/apps/plugins/298935>

### 1.3 Unidades de captura

Se desarrolla un prototipo para cada bicicleta el cual sigue el siguiente diagrama de bloques.



**Figura 9.** Diagrama de bloques unidad de captura.

**Fuente:** Adaptado de <https://es.aliexpress.com/w/wholesale-ttgo-esp32.html>.

#### 1.3.1 Alimentación

Se presentaron 2 opciones para alimentar el sistema: la primera un sistema independiente que utilice como fuente de poder el banco de baterías Xiaomi, que se recargue en el momento que el usuario cargue la bicicleta, y la segunda opción con alimentación directa a partir de la batería de la bicicleta.

Para determinar el consumo del sistema se realizaron pruebas de duración con el banco de baterías, donde se observó un tiempo extendido de funcionamiento, pero se presentó el inconveniente de limitación de corriente en el momento de cargar, puesto que los cargadores actuales de las bicicletas cuentan con una corriente nominal de 2 amperios, que no fue suficiente para alimentar las 2 baterías al mismo tiempo. Esto llevó a tomar la segunda opción teniendo en cuenta que las bicicletas de Campus Verde cuentan con una batería de ion-litio de 36v y 10Ah, además de conocer que el sistema cuenta con tecnología LoRa el cual es de bajo consumo, cercano a 100mAh por lo que no se perjudicaría en el rendimiento de la asistencia eléctrica.

Para desarrollar este sistema de alimentación se realizaron conectores de tipo canon conectados en paralelo al conector de carga, esta conexión nos da un voltaje entre 36v – 42v, el cual es reducido por medio de un convertor dc-dc buck obteniendo una salida constante de 5v, utilizada para alimentar la tarjeta y los sensores del sistema.

**Figura 10.** Conexiones de alimentación.

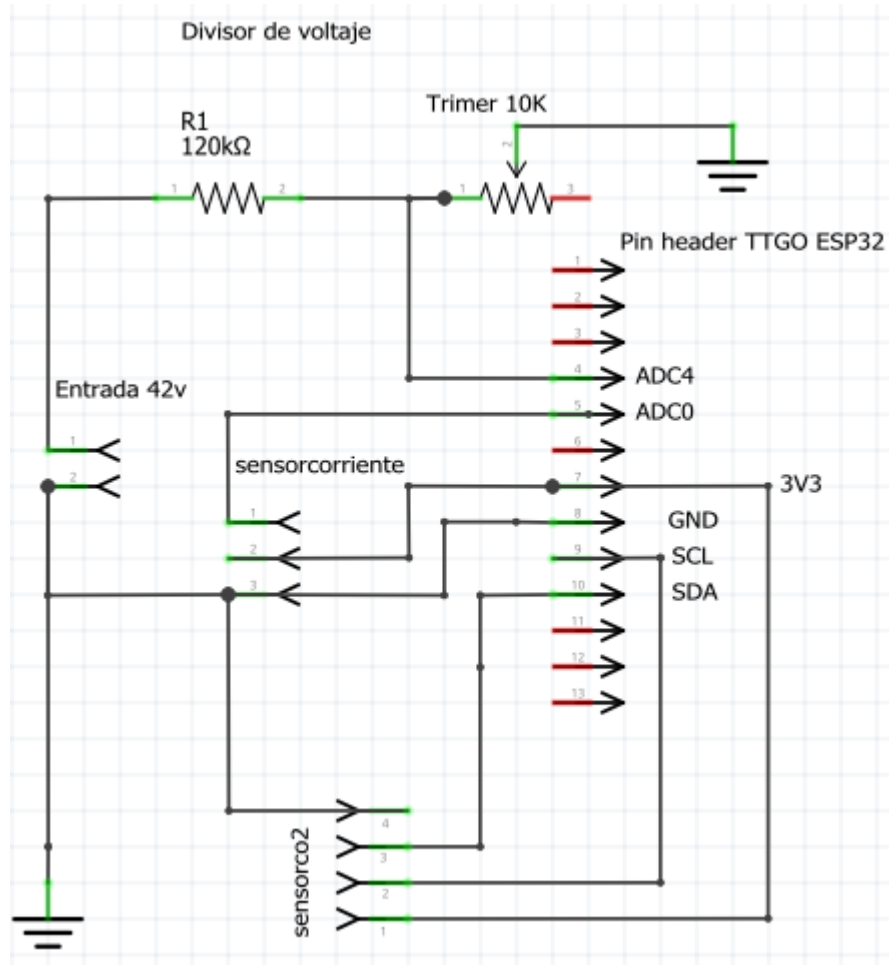


**Fuente:** Registro fotográfico Juan Guillermo Eraso y Sebastian Eraso.

### 1.3.2 Circuito impreso

Se elaboró un circuito impreso para recibir en la placa TTGO ESP 32, la información de los sensores de voltaje, corriente y CO<sub>2</sub>, donde su esquema se observa en la Figura 11.

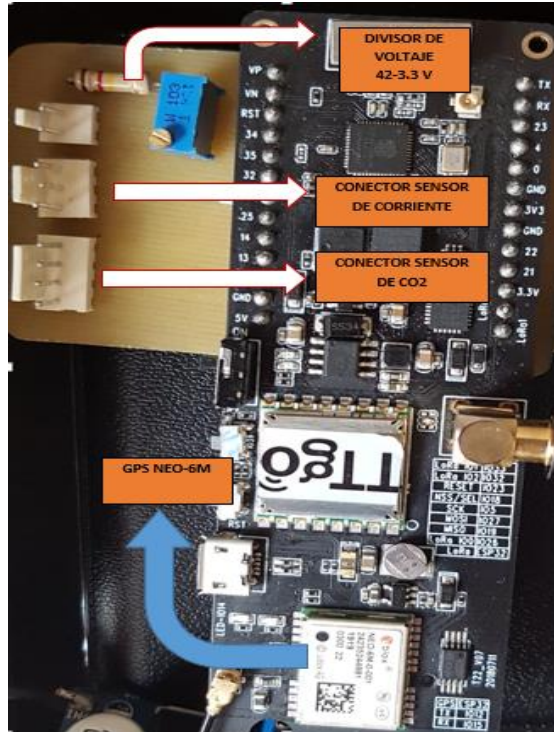
**Figura 11.** Esquemático del circuito.



**Fuente:** Adaptado de Fritzing.

En la Figura 12 se observa el circuito acoplado a la placa TTGO ESP 32, que contiene los conectores para los respectivos sensores y sus conexiones.

**Figura 12.** Componentes del circuito.



**Fuente:** Registro fotográfico Juan Guillermo Eraso y Sebastian Eraso.

Para sensar el voltaje de la batería debido a que en el mercado es difícil encontrar un sensor de voltaje de no más de 24v, se optó por la simple solución de realizar un divisor de voltaje que permita escalar el voltaje a un rango de 0 a 3.3 v, para leerlo en un conversor análogo digital. Por su parte para sensar corriente se utilizó el sensor ACS711.

En la parte relacionada con la medición de variables ambientales, obtenemos del sensor SCD30 las variables de temperatura en [°C], humedad relativa [%] y dióxido de carbono CO<sub>2</sub> en partes por millón [ppm], las cuales se transmiten a la tarjeta TTGO a través de una comunicación I2C.

El procesamiento de la información de las variables obtenidas por los sensores, se realizó por medio de la placa TTGO ESP 32 en la cual se leen periódicamente los valores, además de su localización a través del GPS integrado, organizándolas en un paquete que se envía por medio de la tecnología LoRa. El código de lectura de los sensores se especifica en el anexo 2.

### 1.3.3 Calibración

Al utilizar sensores electrónicos es importante realizar un proceso de calibración, para obtener una mayor veracidad en la información recolectada. Para esta aplicación se utilizaron sensores de voltaje, GPS, corriente y CO<sub>2</sub>, a continuación, se explica el procedimiento de calibración de cada sensor.

El sensor SCD30 utiliza los valores de temperatura, humedad y presión atmosférica para su auto calibración, para esto el fabricante recomienda realizar mediciones por un mínimo de 2 días en exteriores. Realizando pruebas se observó que el tipo de cables son importantes para el correcto funcionamiento, por esto se recurrió a la página de soporte del fabricante que recomienda el tipo de cables Breadboard Jumper.

Con el fin de evaluar el correcto funcionamiento del sensor y la veracidad de sus valores, se comparó sus mediciones con las obtenidas por el sensor Temtop M2000, donde sus valores difieren en aproximadamente 100ppm.

**Figura 13.** Calibración de sensor CO<sub>2</sub>.

```
.co2(ppm):758 temp(C):23.0 humidity(%):52.1  
.co2(ppm):760 temp(C):23.0 humidity(%):52.1  
.co2(ppm):762 temp(C):23.0 humidity(%):52.1
```



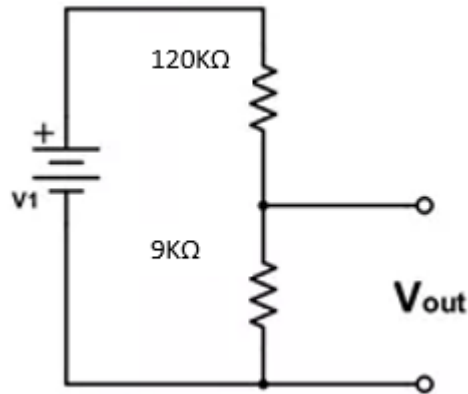
**Fuente:** Registro fotográfico Juan Guillermo Eraso y Sebastian Eraso.

Para el sensor de voltaje como se mencionó anteriormente se utilizó un divisor de voltaje, por tanto, su calibración consistió obtener el factor de conversión K entre el valor del voltaje medido a la batería y el voltaje reducido que se encuentra por siguiente ecuación:

$$V_{out} = V_1 * K , \quad (1)$$

Donde  $V_{out}$  es el voltaje reducido por el divisor de voltaje,  $V_1$  es el voltaje medido de la batería y  $K$  es el factor de conversión que depende de los valores de las resistencias utilizadas en el divisor de voltaje de la Figura 14.

**Figura 14.** Circuito divisor de voltaje.



**Fuente:** Adaptado de:

<https://www.digikey.com/media/Images/Marketing/Resources/Calculators/voltage-divider-diagram.png?ts=fc917814-b472-48e2-aba9-5761999f5ce5&la=es-US>.

De la ecuación (1) se despejó el valor de  $K$ , que se tiene en cuenta al momento de obtener los valores de voltaje en los recorridos de cada bicicleta.

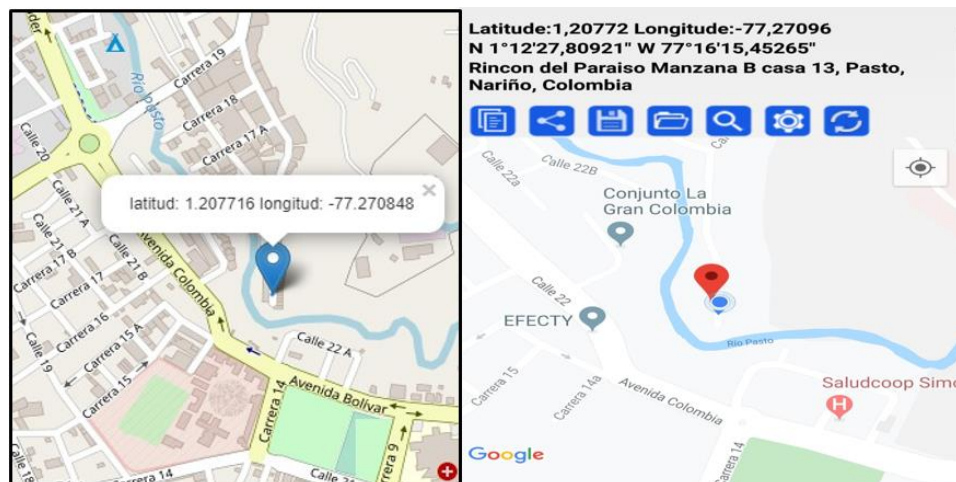
Con respecto a la calibración del sensor de corriente, se debe tener en cuenta que como el sensor mide corrientes negativas y positivas, el valor de 0 amperios no corresponde en la salida del sensor a 0 voltios, por tanto, se conectó el sensor con la bicicleta apagada para encontrar el valor del sensor en voltios que se utilizó como referencia de 0 amperios, para calcular la corriente se utilizó la ecuación:

$$corriente = (V_s - V_{ref}) * \frac{1}{Resolucion} , \quad (2)$$

Donde  $V_s$  es el voltaje de salida del sensor,  $V_{ref}$  es el voltaje de salida con una corriente de 0 amperios y la resolución es de 90mV/A.

Para determinar la exactitud en la ubicación de las bicicletas, se compararon resultados entre 2 tipos de GPS, el de la bicicleta y el del celular Samsung S7 ubicando estos en el mapa del servidor y el de Google Maps respectivamente.

**Figura 15.** Comparación GPS.



**Fuente:** Extractado de Google Maps.

Para tener un índice de comparación, en la Tabla 2 se calculó el porcentaje de error con la siguiente expresión  $error\% = \frac{|GPS\ bicicleta - GPS\ celular|}{GPS\ celular} * 100$ , para 4 ubicaciones.

**Tabla 2.** Comparación exactitud GPS.

| GPS NEO 6M |            | GPS SAMSUNG S7 |           | ERROR    |           |
|------------|------------|----------------|-----------|----------|-----------|
| LATITUD    | LONGITUD   | LATITUD        | LONGITUD  | LATITUD  | LONGITUD  |
| 1.207716   | -77.270848 | 1.20772        | -77.27096 | 0.00033% | 0.00014%  |
| 1.208465   | -77.271008 | 1.20856        | -77.27108 | 0.0078%  | 0.000093% |
| 1.209476   | -77.270888 | 1.2094         | -77.27091 | 0.0063%  | 0.00028%  |
| 1.209917   | -77.270688 | 1.2099         | -77.27074 | 0.0014%  | 0.000067% |

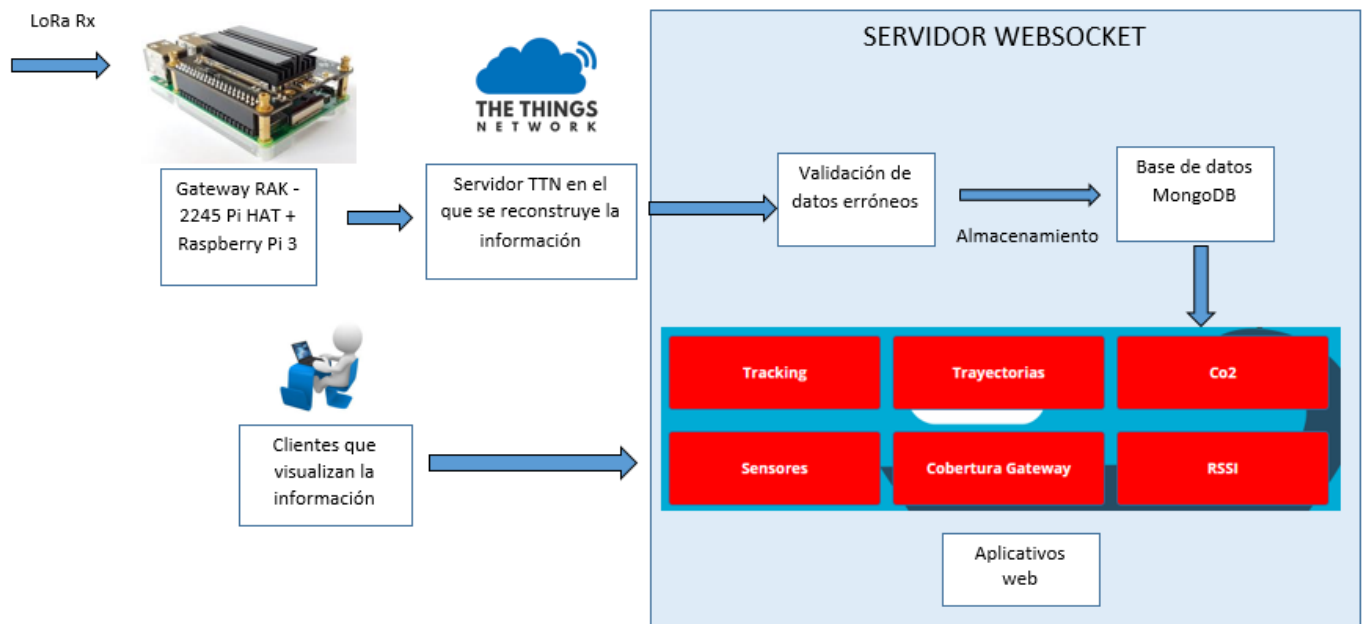
**Fuente:** Elaboración propia.

De los resultados anteriores se aprecia que la ubicación es confiable, puesto que la variación de un GPS a otro es mínima, y al visualizarlas se puede identificar fácilmente el sector en el que se encuentra.

## 1.4 Recepción de información

En la Figura 16 se representa el procedimiento que sigue la información desde la recepción a través de la tecnología LoRa, hasta su almacenamiento y visualización en el servidor web .

**Figura 16.** Diagrama de bloques recepción y visualización de datos.



**Fuente:** Adaptado de <https://www.thethingsnetwork.org/spa/static/img/560e799.png>

### 1.4.1 Configuración Gateway

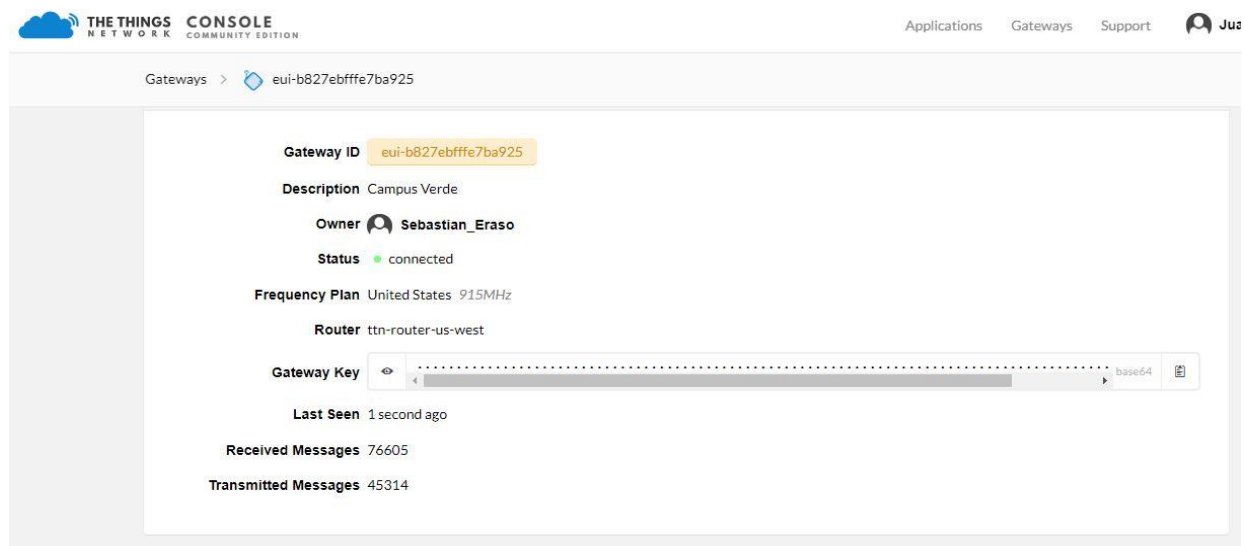
A partir del análisis de las pruebas preliminares se eligió el módulo RAK-2245 PIHAT para RaspberryPI 3, que tiene un rendimiento superior en cuanto a

alcance, sensibilidad y número de canales, además se utilizó una antena omnidireccional de fibra de vidrio especificados en el anexo 1.3 y 1.4.

Para utilizar el Gateway es necesario que la Raspberry cuente con un sistema operativo, en este caso el de propio fabricante denominado Raspbian, el cual contiene una interfaz gráfica que permite realizar configuraciones y programaciones necesarias como acceso a internet y la instalación del firmware basados en la guía que proporciona el fabricante<sup>20</sup> para que el módulo adquiriera las características de un Gateway.

Una vez instalado se accede a las configuraciones del sistema en donde se selecciona el servidor y la frecuencia de comunicación. Los servidores disponibles para este módulo son LoraServer y TTN, donde seleccionamos el último ya que permite acceder a la información desde cualquier sitio con internet y la frecuencia para nuestra región es la de 915 MHz.

Con el Gateway configurado se realiza la conexión con el servidor TTN, para lo cual se crea una cuenta y se registra el Gateway con su respectiva identificación, como resultado de este proceso se obtiene un Gateway con el estatus de conectado listo para recibir información.



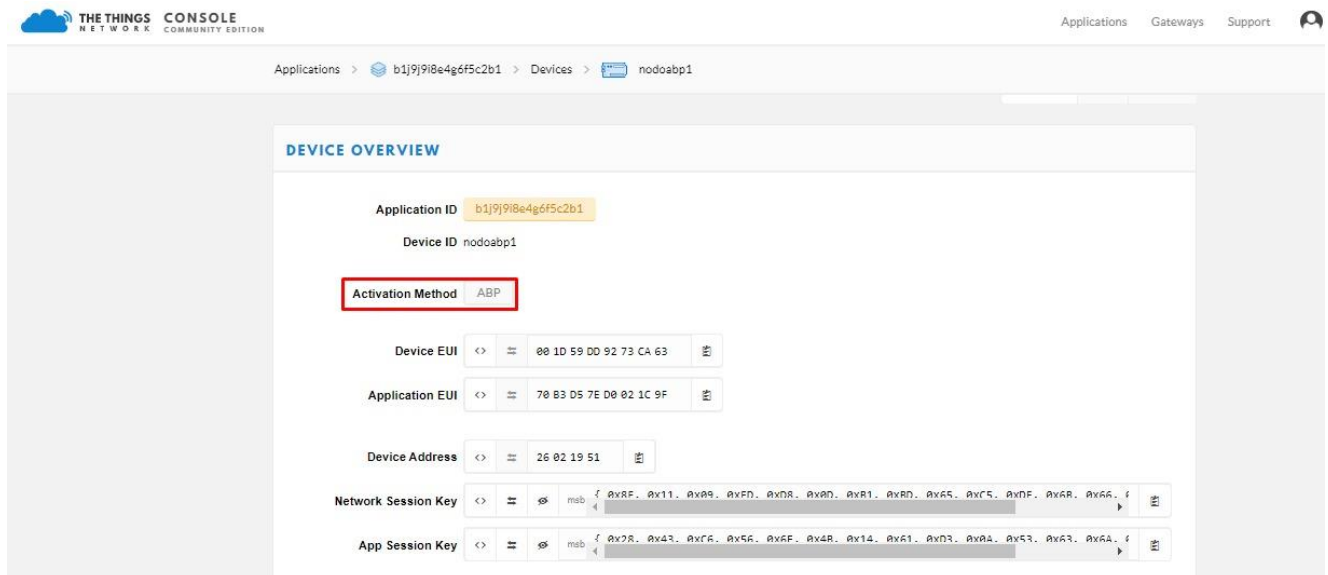
**Figura 17.** Conexión del Gateway en TTN.

**Fuente:** <https://console.thethingsnetwork.org/gateways/>

<sup>20</sup> CONFIGURING THE GATEWAY. {En línea}. Rakwireless. {15 marzo de 2019} disponible en: <https://doc.rakwireless.com/rak2245-pi-hat-edition---lorawan---gateway-concentrator-module/configuring-the-gateway>

### 1.4.2 Aplicación en TTN

Para realizar la comunicación entre el nodo (TTGO ESP32) y el Gateway fue necesario crear una aplicación en el servidor TTN con una identificación única para su uso, dentro de ella se registraron los dispositivos o nodos que enviarán la información requerida. A cada dispositivo se le puede seleccionar dos métodos de autenticación, ABP (Activation-by-personalisation) donde la dirección del dispositivo y la clave de la aplicación son pre configuradas, OTAA (Over-the-air activation) donde la dirección del dispositivo y la clave de aplicación son asignadas en un proceso de activación sobre el aire. Se implementó los 2 métodos para evaluar el mejor en nuestra aplicación, donde se observó que el método ABP presento mejores resultados en cuanto a la estabilidad de la comunicación debido a que una vez enlazado a la aplicación no requiere una reconexión, a diferencia del método OTAA el cual al entrar en una zona donde se perdió la comunicación presento dificultades en cuanto al tiempo de reconexión.



**Figura 18.** Método de autenticación ABP.

**Fuente:**

<https://console.thethingsnetwork.org/applications/b1j9j9i8e4g6f5c2b1/devices/nodoabp1>

Como se observa cada método brinda diferentes claves relacionadas con la seguridad para registrar dispositivos, que deben utilizarse en los dispositivos de envío de la siguiente manera.

**Figura 19.** Keys de autenticación para los nodos ABP.

```
// ABP
static const PROGMEM ul_t NWKSKEY[16] = { 0x23, 0x44, 0x48, 0xA8, 0x84, 0xD1, 0x95, 0x8F, 0x38, 0x39, 0x14, 0x5A, 0x09, 0xA3, 0x46, 0xBF };
static const ul_t PROGMEM APPSKEY[16] = { 0xEE, 0x54, 0x75, 0x4D, 0x65, 0x84, 0x38, 0xB6, 0xFF, 0xF8, 0x15, 0x0D, 0x86, 0xD3, 0x72, 0x6B };
static const u4_t DEVADDR = 0x2602135D;
void os_getArtEui (ul_t* buf) { }
void os_getDevEui (ul_t* buf) { }
void os_getDevKey (ul_t* buf) { }
```

**Fuente:** Arduino IDE, variables de autenticación.

### 1.4.3 Codificación de información

Para el envío de la información a través de la modulación LoRa, se organizó el paquete de envíos en una trama de datos de la siguiente manera, latitud, longitud, id, voltaje, corriente, CO<sub>2</sub>, temperatura, humedad, donde algunas requieren mayor cantidad de bytes por su tipo de variable.

**Figura 20.** Trama del paquete de datos (*payload*) para enviar.

| Latitud |    |    |    | Longitud |    |    |    | Id | Voltaje |    | Corriente |    | CO <sub>2</sub> |    |    | Temperatura |    | Humedad |    |
|---------|----|----|----|----------|----|----|----|----|---------|----|-----------|----|-----------------|----|----|-------------|----|---------|----|
| 00      | 12 | 6D | EA | FB       | 64 | F0 | 58 | 02 | 00      | 20 | 03        | 5D | 00              | 02 | AD | 00          | D4 | 02      | 39 |

1 Byte

**Fuente:** Elaboración propia.

Teniendo en cuenta la misma lógica y orden del envío, se reconstruye la información separando cada variable en la aplicación. Para esto el servidor TTN cuenta con una función de *payload*, que nos permite programarla para recuperar la información de la forma deseada.

**Figura 21.** Reconstrucción del paquete de datos recibido.

```
3 var lat = (bytes[0] << 24) | (bytes[1] << 16) | (bytes[2] << 8) | (bytes[3]);
4 var lon = (bytes[4] << 24) | (bytes[5] << 16) | (bytes[6] << 8) | (bytes[7]);
5 var id = (bytes[8]);
6 var vol = (bytes[9] << 8) | (bytes[10]);
7 var corr = (bytes[11] << 8) | (bytes[12]);
8 var co2 = (bytes[13] << 16) | (bytes[14] << 8) | (bytes[15]);
9 var temp = (bytes[16] << 8) | (bytes[17]);
10 var hum = (bytes[18] << 8) | (bytes[19]);
```

**Fuente:** Adaptado de

<https://console.thethingsnetwork.org/applications/b1j9j9i8e4g6f5c2b1/payload-formats>

## 1.5 Servidor

La necesidad de crear un servidor radica en la importancia de almacenar la información, para acceder a ella libremente en los diferentes aplicativos web. Si bien el servidor TTN cuenta con diferentes integraciones, que permiten visualizar los datos de diferentes maneras, estas presentan algunas limitaciones en cuanto a capacidad de almacenamiento, actualización de datos, seguridad, baja calidad y cantidad en diseños. Por lo tanto, la alternativa elegida es implementar un servidor capaz de descargar la información del servidor TTN, en el momento en que es actualizada por el Gateway.

La creación del servidor consistió en un proceso, que inicio con la instalación de Node.js que es un entorno en tiempo de ejecución multiplataforma basado en JavaScript, que permite realizar la programación del mismo, donde utilizamos diferentes librerías que nos facilitan conformar la estructura.

La librería socket.io se utiliza para una comunicación bidireccional entre el servidor y los clientes en tiempo real. Esta comunicación se efectúa a través de eventos, tanto servidor como cliente pueden escucharlos con el comando *socket.on* y emitirlos con *socket.emit*. Por medio de ellos se crean instrucciones que permiten descargar la información del servidor TTN, a través de la librería que este mismo proporciona y, almacenarla en nuestro servidor en la base de datos MongoDB<sup>21</sup>, que es un sistema de base de datos NoSQL orientado a documentos de código abierto, en la que se guardan estructuras de datos con formato JSON como se muestra en la Figura 22, que brinda un esquema dinámico por lo que su integración a aplicaciones es más fácil y rápida.

**Figura 22.** Formato JSON de la base de datos.

```
const ECGSchema = new Schema ({
  id_Number : Number,
  lat : [Number],
  lon:[Number],
  voltaje:[Number],
  corriente:[Number],
  co2:[Number],
  temperatura:[Number],
  humedad:[Number],
  rssi:[Number],
  gateway:[Number],
  fecha : []
})
```

**Fuente:** Software Atom.

---

<sup>21</sup> MONGODB: La base de datos líder para aplicaciones modernas. {En línea}. MongoDB. {15 marzo de 2019} disponible en: <https://www.mongodb.com/es>

En la Figura 22 se representa un objeto JSON conformado de diferentes arreglos para cada una de las variables, en este caso id\_Number corresponde a una identificación asignada para diferenciar cada bicicleta, posteriormente tenemos las coordenadas geográficas latitud (lat) y longitud (lon), seguido de las variables de los sensores y su respectiva fecha. El funcionamiento consiste en transmitir a través de un evento la información obtenida del servidor TTN, en el que cada variable se ubica en la última posición de un vector de la base de datos, el cual aumenta su tamaño cada vez que recibe nueva información. La base de datos es de tipo dinámica por lo cual podemos tanto actualizar como suprimir información, según sea requerido.

En la Figura 23 se observa que la información de las variables en la base de datos del servidor, presenta dificultades al momento de tener muchos datos, ya que no solo se cuenta con la información de una sola bicicleta y, él envió es continuo aproximadamente de cada 10 segundos. Por esto se planteó el uso de aplicativos web donde su interfaz gráfica se desarrolló a través de HTML y su funcionamiento por medio de *Web-Sockets*, incluidos en el servidor, que hacen más practica la visualización de la información de coordenadas y valores de los sensores.

**Figura 23.** Visualización base de datos en formato JSON

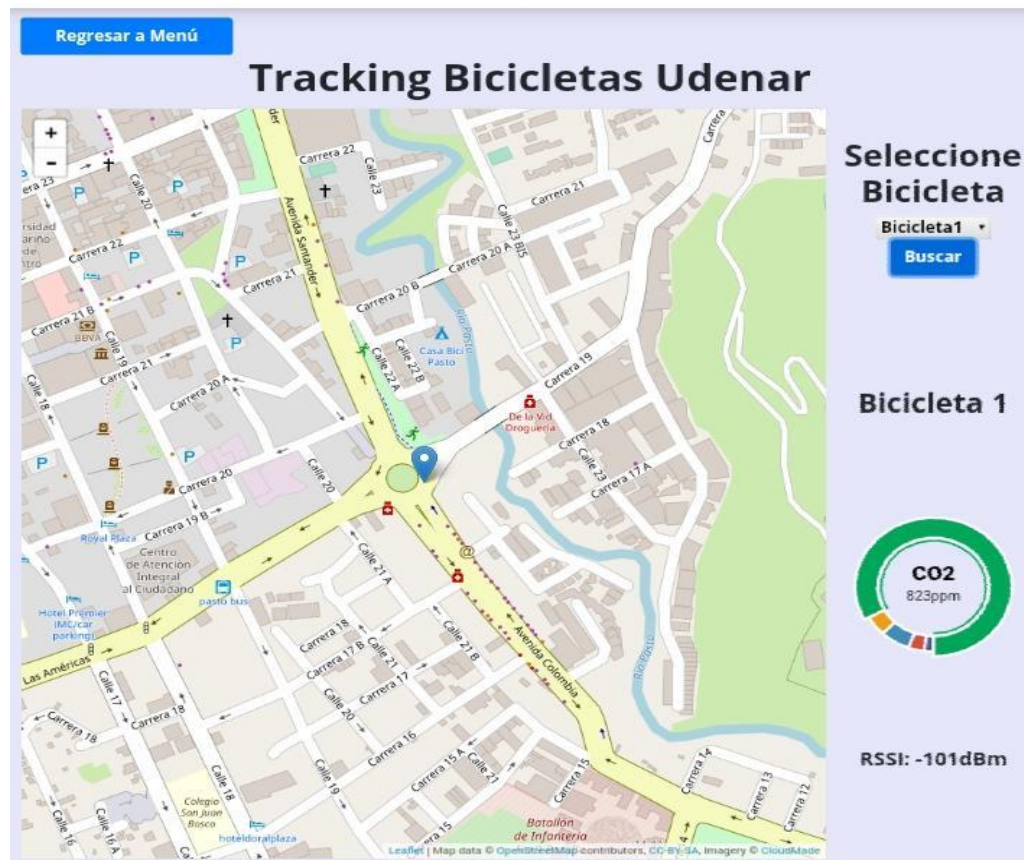
```
{
  + lat: [...],
  - lon: [
    -77.270856,
    -77.270848,
    -77.270848
  ],
  + voltaje: [...],
  + corriente: [...],
  - co2: [
    462,
    461,
    461
  ],
  - temperatura: [
    20.2,
    20.2,
    20.2
  ],
  - humedad: [
    63,
    63,
    63
  ],
  - fecha: [
    "2019-10-24 19:59:30",
    "2019-10-24 19:59:40",
    "2019-10-24 19:59:51"
  ],
  _id: "5d9cf58cb349e01554740ecb",
  id_Number: 3,
  _v: 3
}
```

**Fuente:** <http://192.68.185.29:8081/api/ecg>

El primer aplicativo desarrollado consiste en el seguimiento en tiempo real de la posición de la bicicleta, el cual recibe a través de un evento las coordenadas

actuales en el instante que llegan al servidor, de una determinada bicicleta seleccionada por el usuario, por medio de un botón y las ubica a través de un marcador en el mapa implementado en la plataforma Leaflet.

**Figura 24.** Interfaz aplicativo tracking.



**Fuente:** Software Leaflet.

No siempre se desea ver la ubicación en tiempo real, en ocasiones se desea acceder a las trayectorias de fechas anteriores, por lo cual se desarrolló un aplicativo que permite buscar para cada bicicleta posiciones a lo largo de un día.

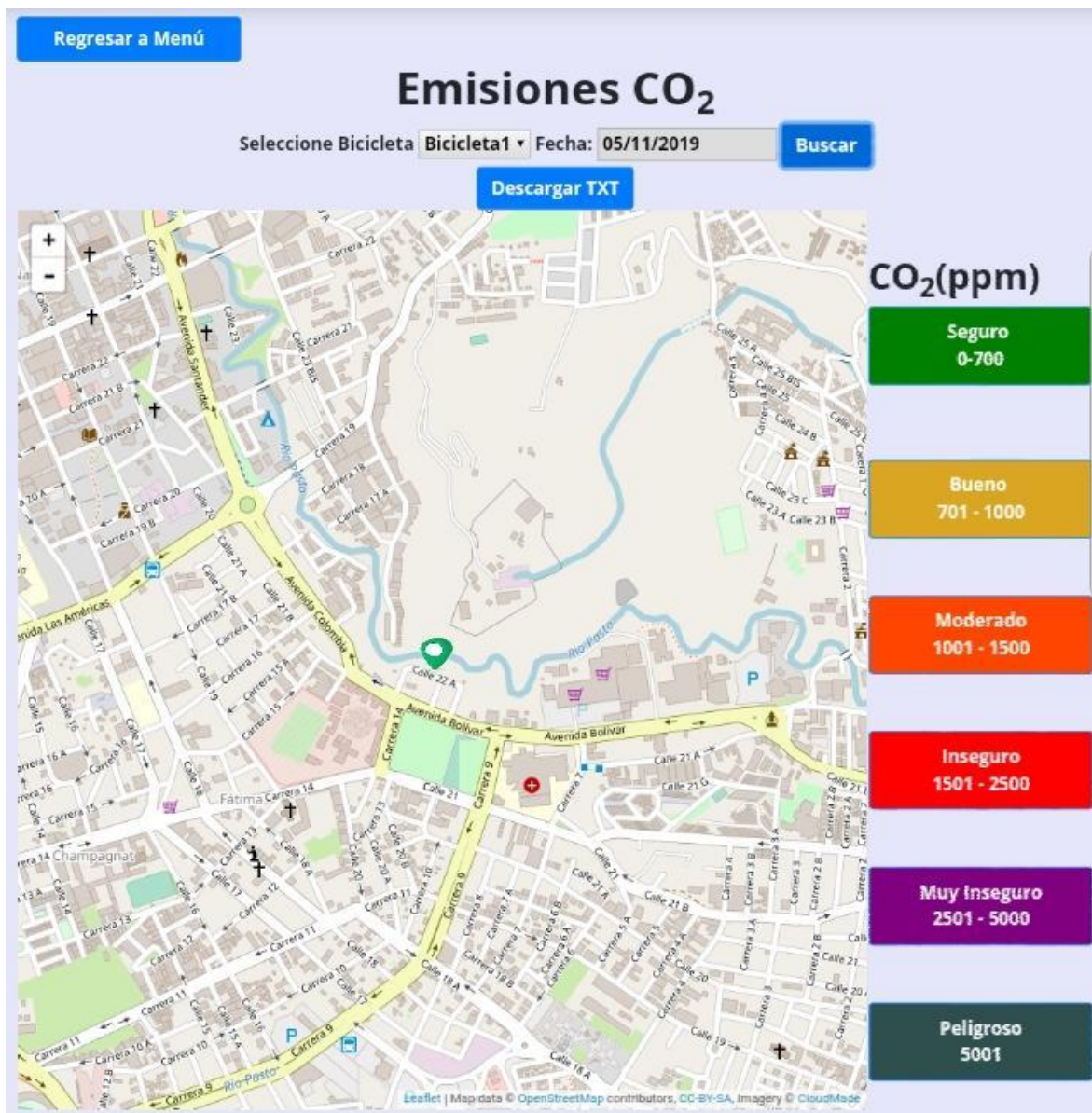
**Figura 25.** Aplicativo registro de ubicación.



**Fuente:** Software Leaflet.

Para el análisis de emisiones de CO<sub>2</sub> es importante conocer el lugar en que se realiza la medición, por esto se desarrolló un aplicativo que permite observar el rango de las emisiones además de su ubicación.

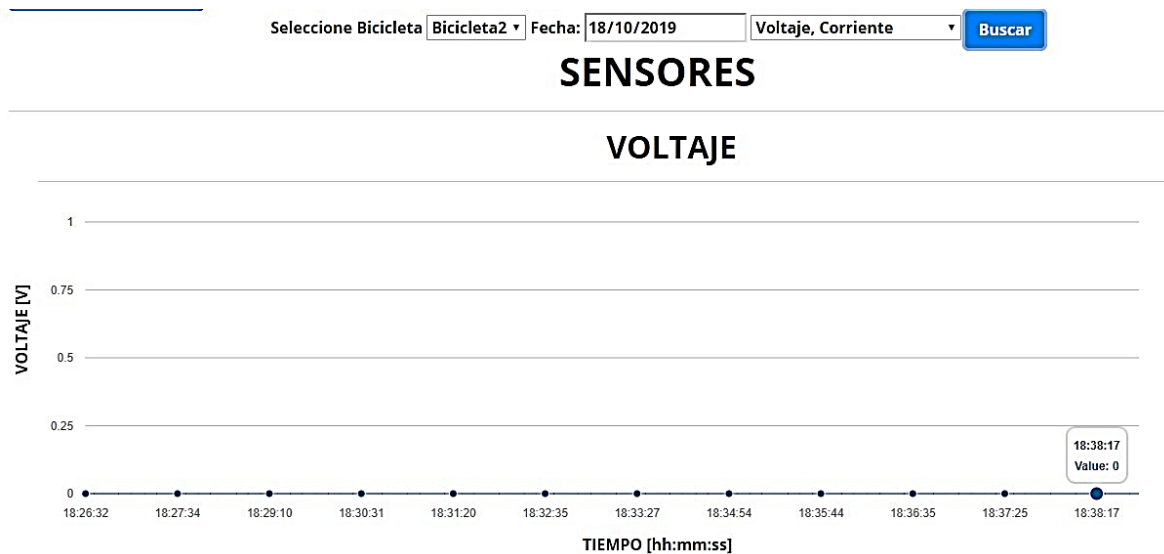
Figura 26. Aplicativo registro de emisiones CO<sub>2</sub>.



Fuente: Software Leaflet.

Para una visualización más cómoda de las variables restantes, pertenecientes a los sensores integrados en el dispositivo, se empleó el graficador de JavaScript denominado Morris.js, este realiza gráficas del valor del sensor vs tiempo, del mismo modo que los aplicativos anteriores permite buscar los datos a lo largo de un día y, se puede visualizar los sensores de voltaje, corriente, temperatura, humedad, además de una potencia calculada con los valores de los dos primeros.

**Figura 27.** Aplicativo registro de sensores.



**Fuente:** Software Morris.js

## 2. RESULTADOS

### 2.1 Hardware

Para la implementación del sistema encargado de medir las variables en una bicicleta de ubicación, emisiones de  $CO_2$ , temperatura, humedad, voltaje y corriente, y enviarlas a través de la tecnología LoRa, se tuvo en cuenta que no afecte la comodidad del usuario ni la imagen de la bicicleta, además que puede estar expuesto a diferentes condiciones climáticas. Para esto se desarrolló un circuito que compacto, el cual permite realizar las conexiones entre los sensores y la placa TTGO ESP 32, alimentada con la batería de la bicicleta por medio del conversor dc-dc buck, su promedio de consumo se especifica en la Tabla 3.

**Tabla 3.** Consumo de unidad de captura.

| DISPOSITIVO | CONSUMO mA | ESTADO TRANSMISIÓN<br>mA |
|-------------|------------|--------------------------|
| TTGO ESP32  | 10         | 30                       |
| SCD30       | 19         | 19                       |
| ACS714      | 2          | 2                        |

**Fuente:** Elaboración propia.

Para la protección y ubicación de los componentes del sistema se utilizó una caja de medidas 14.5cm de largo, 5.4 cm alto y 9.5 cm ancho, adecuada para las conexiones del sistema.

**Figura 28.** Caja protectora del dispositivo de captura.



**Fuente:** Registro fotográfico Juan Guillermo Eraso y Sebastian Eraso.

El sensor de CO<sub>2</sub> SCD30 es el único componente externo, porque al ubicarlo al interior de la caja este no pudo detectar cambios del ambiente exterior, por este motivo fue ubicado en la parte inferior de la caja, donde tiene mayor protección frente a ciertas condiciones ambientales.

## 2.2 Aplicativos web

Para verificar la comunicación entre el nodo y Gateway, se utilizó como puente de información el servidor TTN, y sus diferentes interfaces que permitieron verificar el estado de conexión del Gateway y de los nodos, además del tiempo de actualización de cada paquete de datos.

**Figura 29.** Estado de conexión nodos.



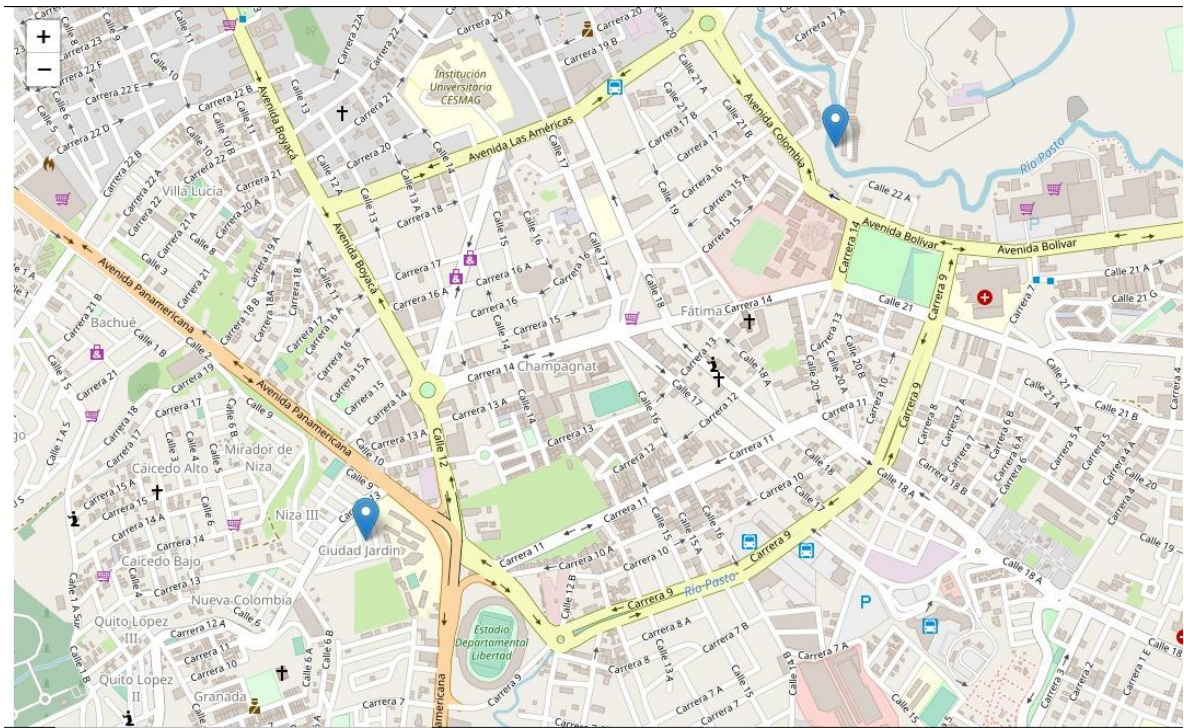
**Fuente:**

<https://console.thethingsnetwork.org/applications/b1j9j9i8e4g6f5c2b1/devices/nodoabp1>

Esto permitió determinar que dispositivos se encuentran activos, en caso de desconexión buscar las causas, fuera por pérdida de cobertura analizando la última posición recibida o alimentación del sistema si el tiempo de desconexión fue considerable. Los contadores de trama tanto *frames up* como *frames down*, son métodos de seguridad adicionales que posee el TTN, para el proyecto fue la mejor opción fijarlos a un determinado valor, porque en momentos de pérdida de cobertura no fue posible recuperar la conexión, debido a que ocurría una desincronización entre los contadores de trama del servidor y del nodo.

Para comprobar que el paquete de información se recibió de acuerdo a los parámetros programados en el envío, se accedió a la interfaz de este servidor para observar el tráfico de datos que se muestra en la Figura 30, que permitió visualizar los datos recibidos y organizados por el *payload*, para comprobar que los datos fueran coherentes, verificar si los sensores se encuentran activos para cada dispositivo.



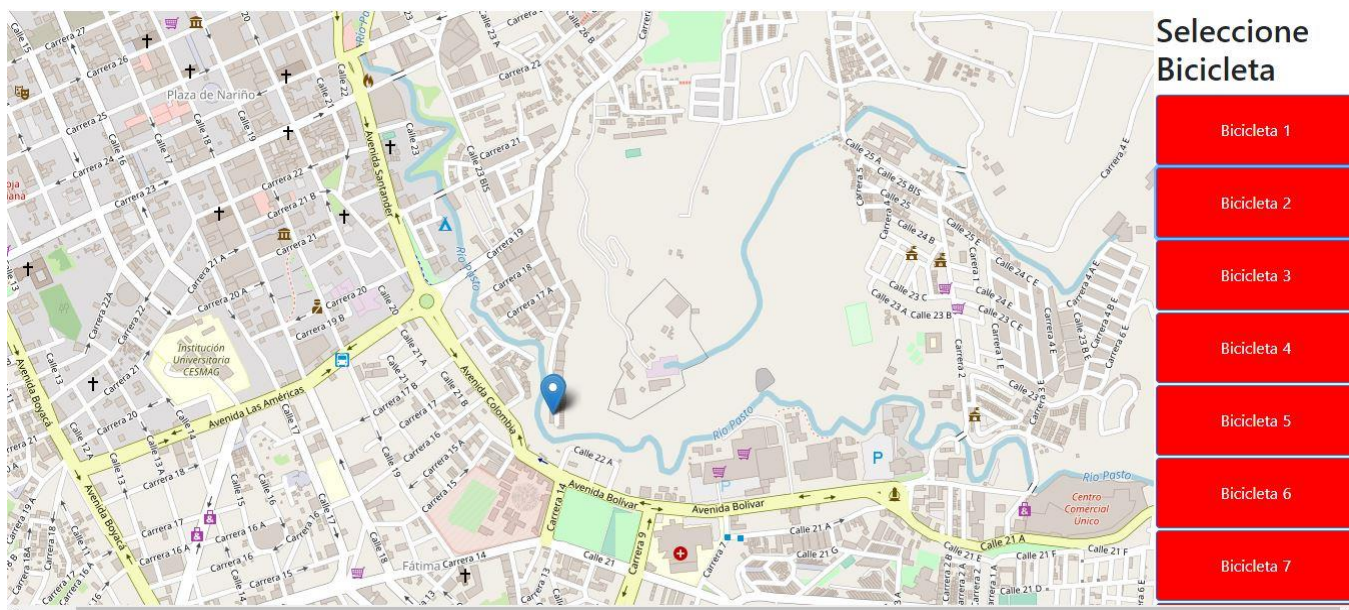


**Figura 32.** Ubicación Gateways prueba 1.

**Fuente:** Software Leaflet.

### 2.2.1 Seguimiento en tiempo real

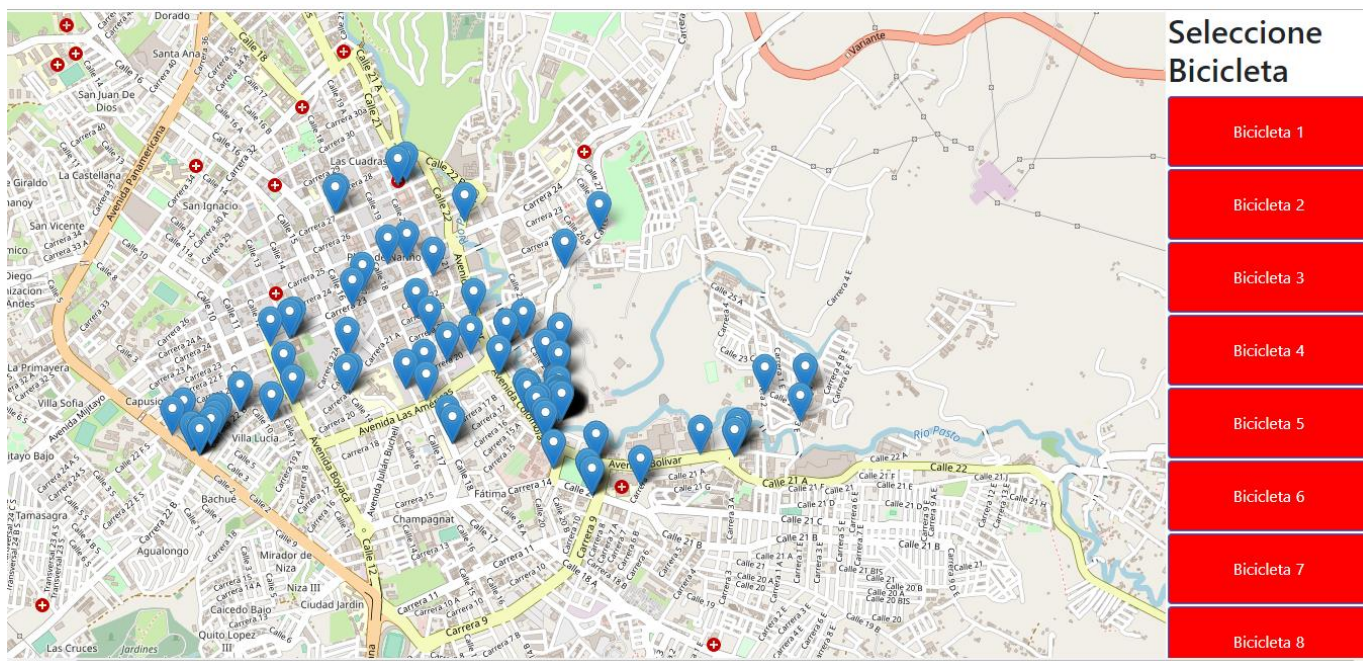
En este aplicativo se ubica en un mapa la última posición de latitud y longitud encontrada en la base de datos, en el momento en que se actualiza la información a la base, se emite un evento que automáticamente posiciona las nuevas coordenadas de la bicicleta, esto ocurre en un intervalo cercano a 10 segundos para cada bicicleta. En la Figura 33 y Figura 34 se muestran los resultados de una prueba realizada para una bicicleta, se observa que comenzó de una posición inicial cercana a la Avenida Colombia. El recorrido registrado se realiza por los sectores de Avenida Colombia, Parque Bolívar, Batallón Boyacá, Iglesia Fátima, Avenida Champagnat, Avenida Panamericana y Avenida Américas.



**Figura 33.** Posición inicial de prueba de tracking.

**Fuente:** Software Leaflet.

**Figura 34.** Tracking en tiempo real.



**Fuente:** Software Leaflet.

### 2.2.2 Registro de ubicación por día y distancia recorrida.

Este aplicativo permitió acceder a la información almacenada en la base de datos, posicionar las trayectorias recorridas por la bicicleta en una determinada fecha, y calcular la distancia total recorrida a partir de la fórmula del *haversine* o fórmula del semiverseno<sup>22</sup>, que permitió calcular la distancia entre dos puntos conociendo su latitud y longitud.

En el aplicativo se trazaron las trayectorias uniendo por medio de una recta cada posición con la anterior, imprime la distancia aproximada recorrida por la bicicleta. Además brindó la opción de descargar estos resultados en un bloc de notas.

**Figura 35.** Aplicativo de distancia y trayectoria.



**Fuente:** Software Leaflet.

<sup>22</sup> GUERRERO, José. "Distancia entre dos puntos de la superficie terrestre mediante la fórmula de Haversine con Python". {En línea}. {18 marzo de 2019} disponible en: <https://joseguerreroa.wordpress.com/2012/11/13/distancia-entre-dos-puntos-de-la-superficie-terrestre-mediante-la-formula-de-haversine-con-python/>

### 2.2.3 Registro de emisiones CO<sub>2</sub>

La importancia de este aplicativo fue el registro continuo de emisiones de CO<sub>2</sub> medidas en partes por millón, con su respectiva ubicación, que permitió clasificar sectores de la ciudad de acuerdo a los niveles de emisión de CO<sub>2</sub>. Estos datos fueron útiles para organizaciones encargadas del cuidado ambiental, para evaluar alternativas de solución que contribuyeron a esta problemática ambiental.



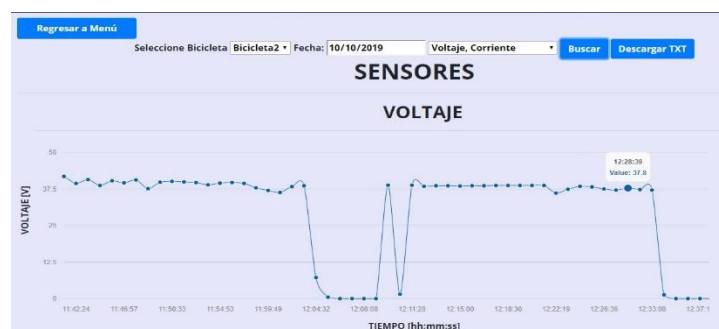
Figura 36. Recorrido con emisiones de CO<sub>2</sub>.

Fuente: Software Leaflet.

### 2.2.4 Registro de sensores

Este aplicativo tiene una interfaz para buscar y visualizar la variación de voltaje, corriente, potencia y energía calculada a partir de las anteriores variables, temperatura y humedad con respecto a una determinada hora.

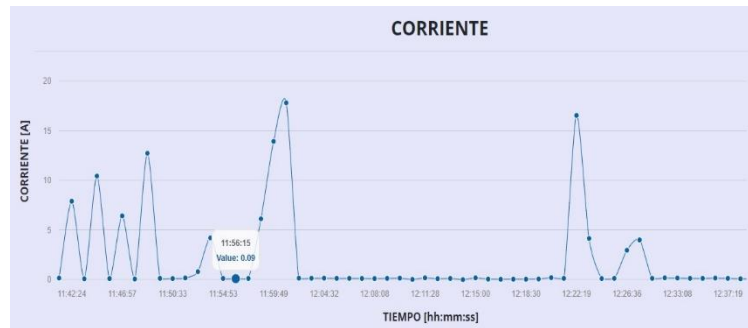
Figura 37.  
voltaje versus



Gráfica de  
tiempo.

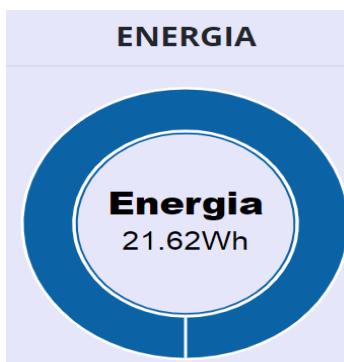
Fuente: Software Morris.js

Figura 38. Gráfica de corriente vs tiempo.



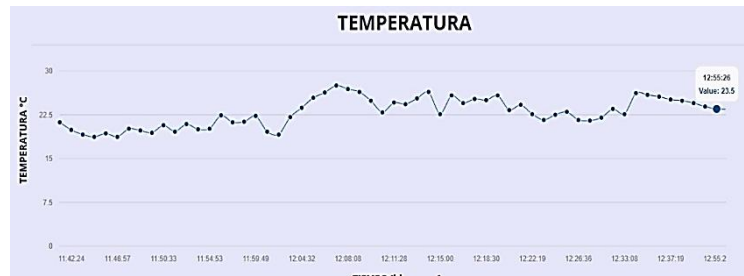
Fuente: Software Morris.js.

Figura 39. Gráficas de potencia y energía calculadas.



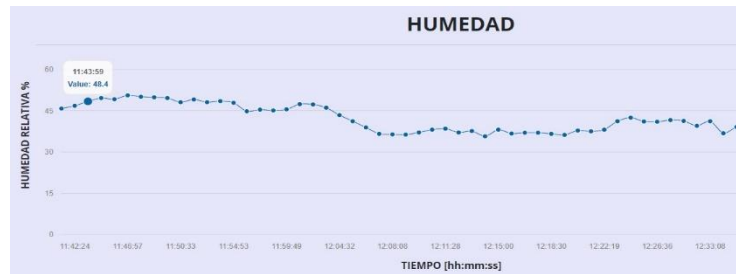
Fuente: Software Morris.js.

**Figura 40.** Gráfica de temperatura versus tiempo.



**Fuente:** Software Morris.js.

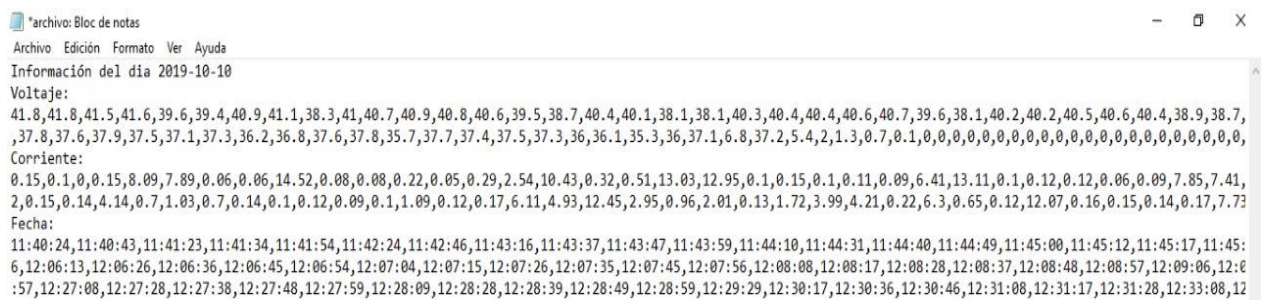
**Figura 41.** Gráfica de humedad vs tiempo



**Fuente:** Software Morris.js

Además, este aplicativo permitió descargar los datos visualizados de las diferentes variables con su respectiva hora en un archivo de texto, en caso de que fueran requeridos en otras aplicaciones fuera del servidor.

**Figura 42.** Formato de descarga de los datos visualizados.



**Fuente:** Software Node.js

### 3. REPORTE DE COBERTURA

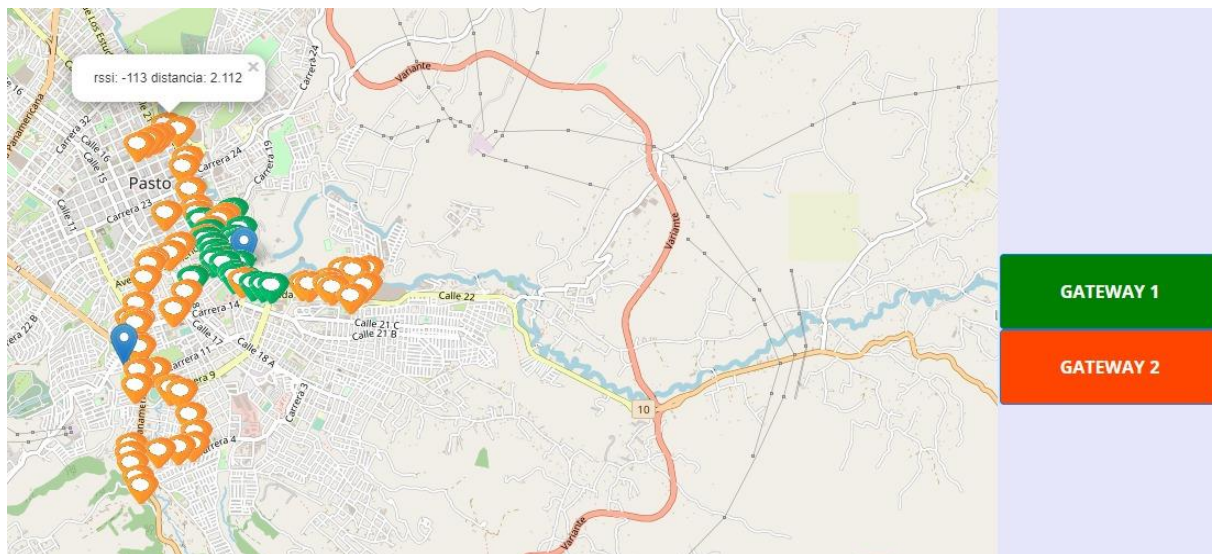
Para evaluar características de funcionamiento de la red como tiempos de envío, alcance e indicador de fuerza de la señal recibida (RSSI). Se efectuó un recorrido con una de las bicicletas utilizando la ubicación de los Gateway de la Figura 32, almacenando en la base de datos las variables sensadas, identificando que Gateway recibió la información y cuál fue su valor de RSSI, a partir de la *metadata* actualizada por el Gateway a cada paquete de datos como se observa en la Figura 43.

**Figura 43.** Metadata Gateway.

```
{
  time: '2019-11-04T20:17:22.824095818Z',
  frequency: 905.3,
  modulation: 'LORA',
  data_rate: 'SF7BW125',
  airtime: 77056000,
  coding_rate: '4/5',
  gateways: [
    {
      gtw_id: 'eui-b827ebfffe7ba925',
      timestamp: 121058772,
      time: '2019-11-04T20:18:21.379984Z',
      channel: 7,
      rssi: -25,
      snr: 10,
      rf_chain: 1
    }
  ]
}
```

**Fuente:** Software Windows PowerShell.

En la Figura 44 se diferencia por color a que Gateway llegó la información, la distancia con respecto a este y su RSSI. Existen zonas en que los dos Gateway presentaron cobertura, para estos casos se tiene en cuenta el mayor valor de RSSI. Esta prueba proporciona una noción de la cobertura que tuvo cada Gateway en sectores de la ciudad.

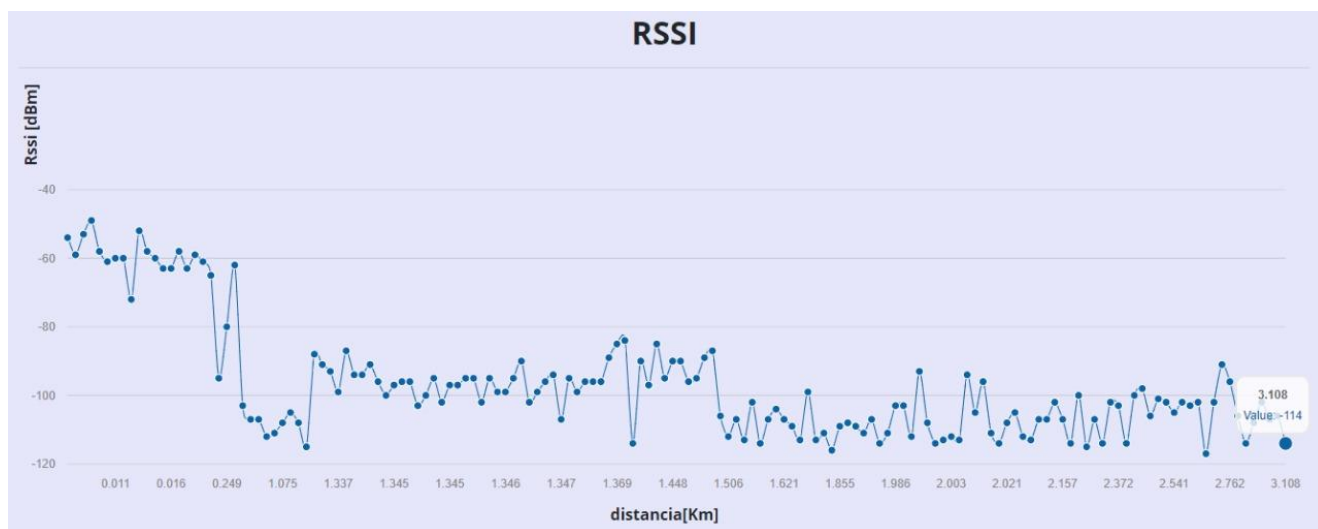


**Figura 44.** Cobertura de gateways prueba 1.

**Fuente:** Software Leaflet.

Otra forma de visualizar los datos de la prueba es graficar el indicador de fuerza de la señal recibida (RSSI) en función de la distancia.

**Figura 45.** RSSI vs Distancia.



**Fuente:** Software Morris.js.

Para tener una idea más clara de los resultados se realizó un promedio de RSSI para diferentes rangos de distancia cada 500 metros, como se observa en la Tabla 4 y Tabla 5.

| GATEWAY 1     |            |                    |
|---------------|------------|--------------------|
| DISTANCIA (m) | RSSI [dBm] | NUMERO DE PAQUETES |
| 0 – 500       | -69.8      | 218                |
| 500 – 1000    | -115       | 6                  |
| 1000-1500     | -119       | 1                  |
| 1500-2000     | -----      | 0                  |

**Tabla 4.** Prueba de distancia Gateway 1.

**Fuente:** Elaboración propia.

**Tabla 5.** Prueba de distancia Gateway 2

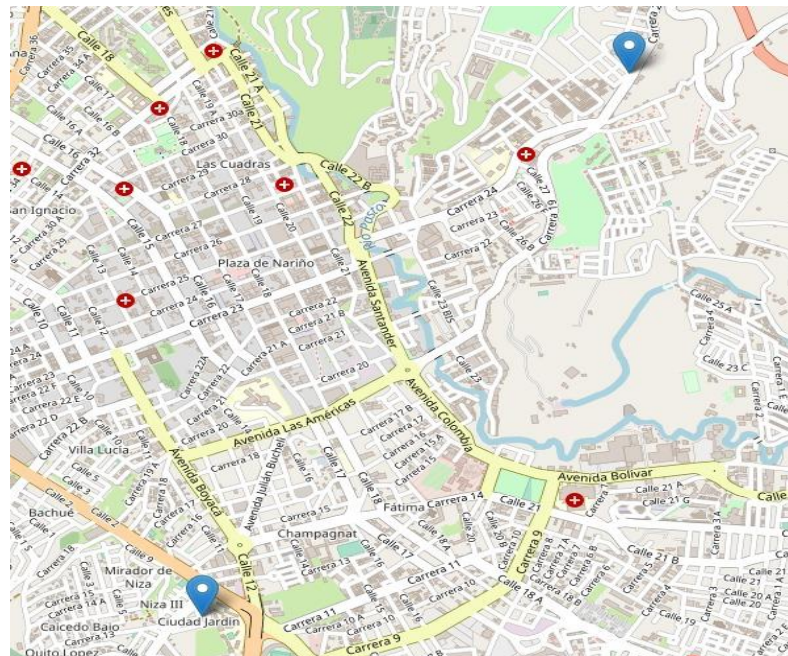
| GATEWAY 2    |            |                    |
|--------------|------------|--------------------|
| DISTANCIA(m) | RSSI [dBm] | NUMERO DE PAQUETES |
| 0 – 500      | -96.8      | 30                 |
| 500 – 1000   | -104.8     | 47                 |
| 1000-1500    | -112       | 23                 |
| 1500-2000    | -109.6     | 32                 |
| 2000-2500    | -112.6     | 11                 |

**Fuente:** Elaboración propia.

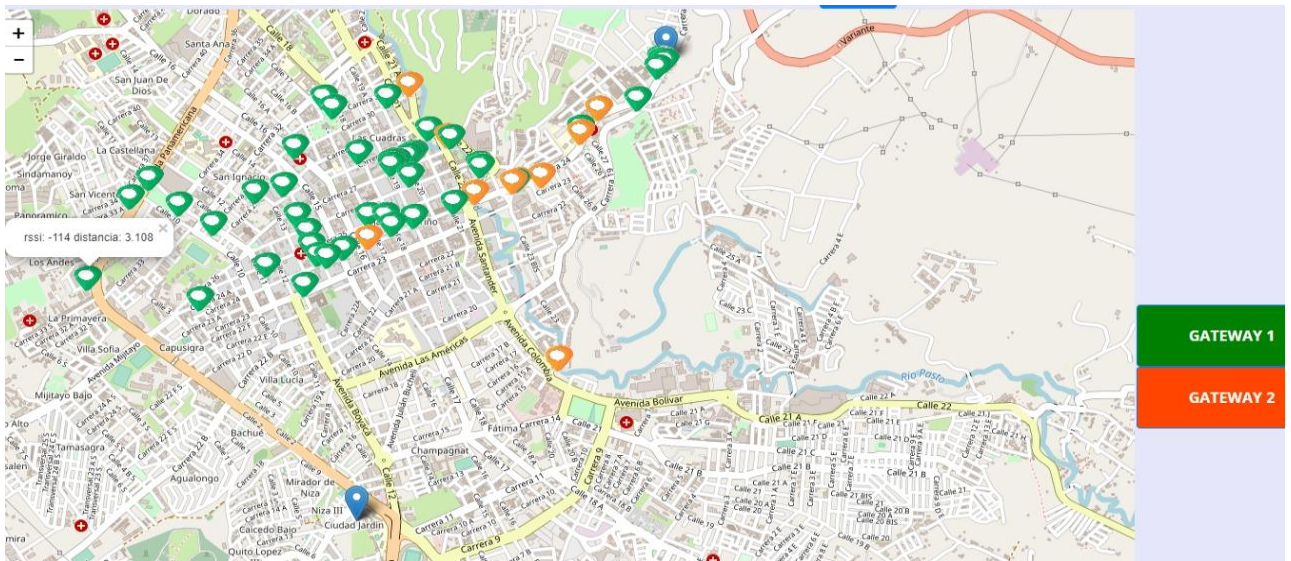
Los datos muestran que el alcance de distancia es aceptable. No necesariamente mayor distancia implica menor RSSI, esta fue afectada principalmente por la línea de vista, es decir la cantidad de obstáculos que se encontraban entre el nodo y el Gateway.

Como se observa el Gateway 1 presentó una cobertura menor a la esperada, en comparación con el Gateway 2 que cubrió una distancia de más de 2 kilómetros. Analizando el contexto de cada Gateway hay una diferencia en altura y por tanto en línea de vista, porque el Gateway 2 se ubicó a 30 metros más que el Gateway 1. Debido a esto se decidió reubicar el Gateway 1 a mayor altura como se muestra en la Figura 46 y se realizó la prueba indicada en la Figura 47 evaluando la zona de cobertura del Gateway reubicado sin tener en cuenta la zona de cobertura del Gateway 2 ya que este se encontraba en la misma posición.

**Figura 46.** Ubicación gateways prueba 2.



**Fuente:** Software Leaflet.



**Figura 47.** Resultados de cobertura prueba 2.

**Fuente:** Software Leaflet.

Los datos de la prueba anterior se resumen en la Tabla 6.

**Tabla 6.** Segunda prueba de distancia Gateway 1.

| GATEWAY 1     |            |                    |
|---------------|------------|--------------------|
| DISTANCIA (m) | RSSI [dBm] | NUMERO DE PAQUETES |
| 0 – 500       | -63.8      | 23                 |
| 500 – 1000    | -108.6     | 3                  |
| 1000-1500     | -95.1      | 27                 |
| 1500-2000     | -108       | 18                 |
| 2000-2500     | -106.8     | 16                 |
| >2500         | -103.8     | 10                 |

**Fuente:** Elaboración propia.

El rendimiento del Gateway 1 mejoró respecto a la anterior ubicación, puesto que la zona de cobertura aumentó más de 2 kilómetros.

Para evaluar la red con respecto al tiempo de recepción de datos, se realizó un conteo de paquetes teniendo en cuenta la hora de inicio y fin de las pruebas, obteniendo de estas un promedio de paquetes por minuto, como se indica en la Tabla 7.

**Tabla 7.** Promedio de paquetes por minuto

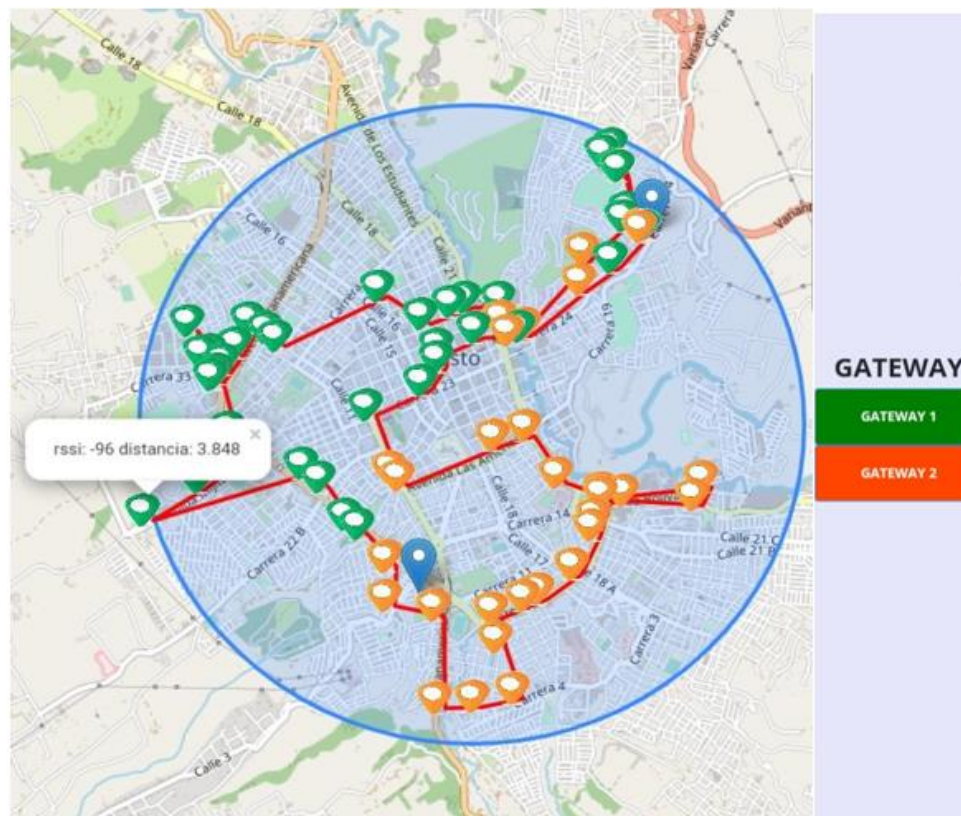
|          | HORA<br>INICIO | HORA<br>FINAL | DURACION<br>MINUTOS | NUMERO<br>DE<br>PAQUETES | PROMEDIO<br>POR<br>MINUTO |
|----------|----------------|---------------|---------------------|--------------------------|---------------------------|
| PRUEBA 1 | 14:46          | 17:27         | 161                 | 368                      | 2.28                      |
| PRUEBA 2 | 14:27          | 15:28         | 61                  | 126                      | 2.06                      |

**Fuente:** Elaboración propia.

Se obtuvo un promedio aproximado de 2 paquetes por minuto, el nodo fue programado para enviar paquetes cada 10 segundos, cuando se encontraban en distancia mínima se reciben a 15 segundos aproximadamente. Por tanto, en teoría se perdieron la mitad de los paquetes de datos, debido a que se depende de muchos factores externos no se puede establecer un tiempo estándar de recepción, puesto que en unas zonas los datos llegaron más rápido que en otras.

Teniendo un estimado de la cobertura de cada gateway a partir de las pruebas anteriores, se efectuó una prueba adicional recorriendo los sectores de cobertura de cada gateway, para estimar un radio de cobertura en la ciudad con la ubicación de los gateways de la Figura 46.

**Figura 48.** Radio de cobertura.



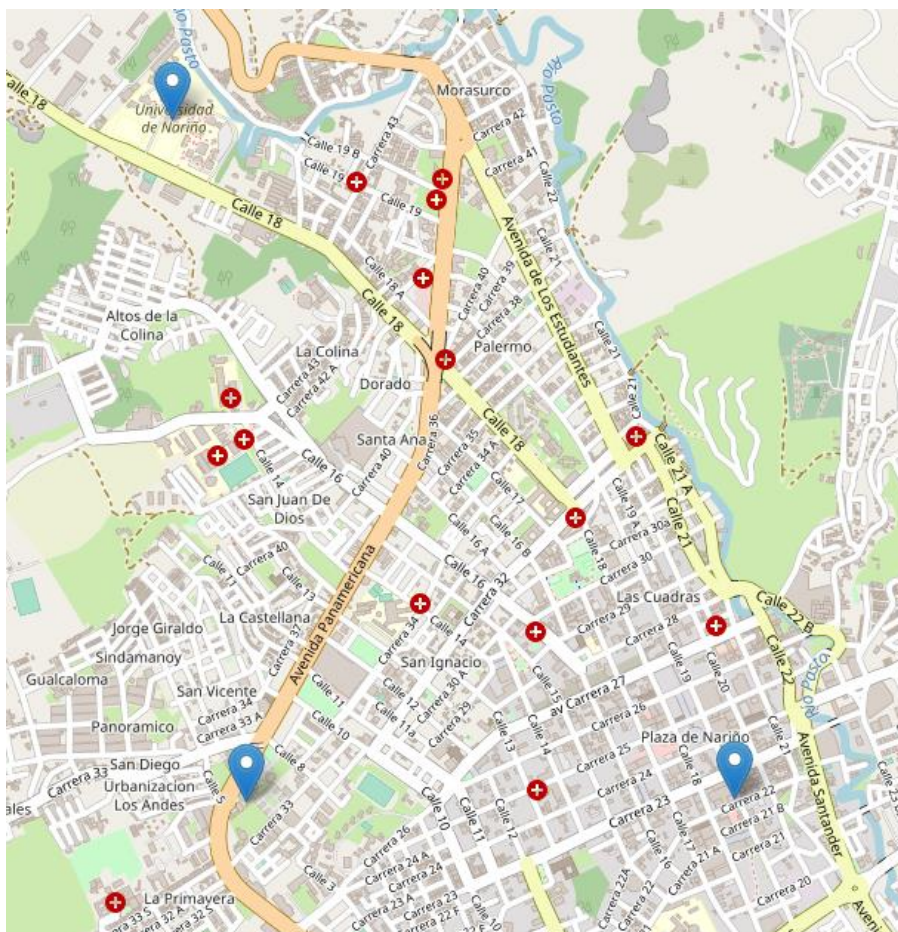
**Fuente:** Software Leaflet.

En la Figura 48 se observa los puntos recorridos y en que Gateway fueron recibidos, evaluando las distancias de los puntos al Gateway, se trazó una circunferencia de 2 Km de radio aproximadamente que encierra la zona de cobertura, teniendo en cuenta que en ciertas zonas de la ciudad los obstáculos no permitieron la comunicación.

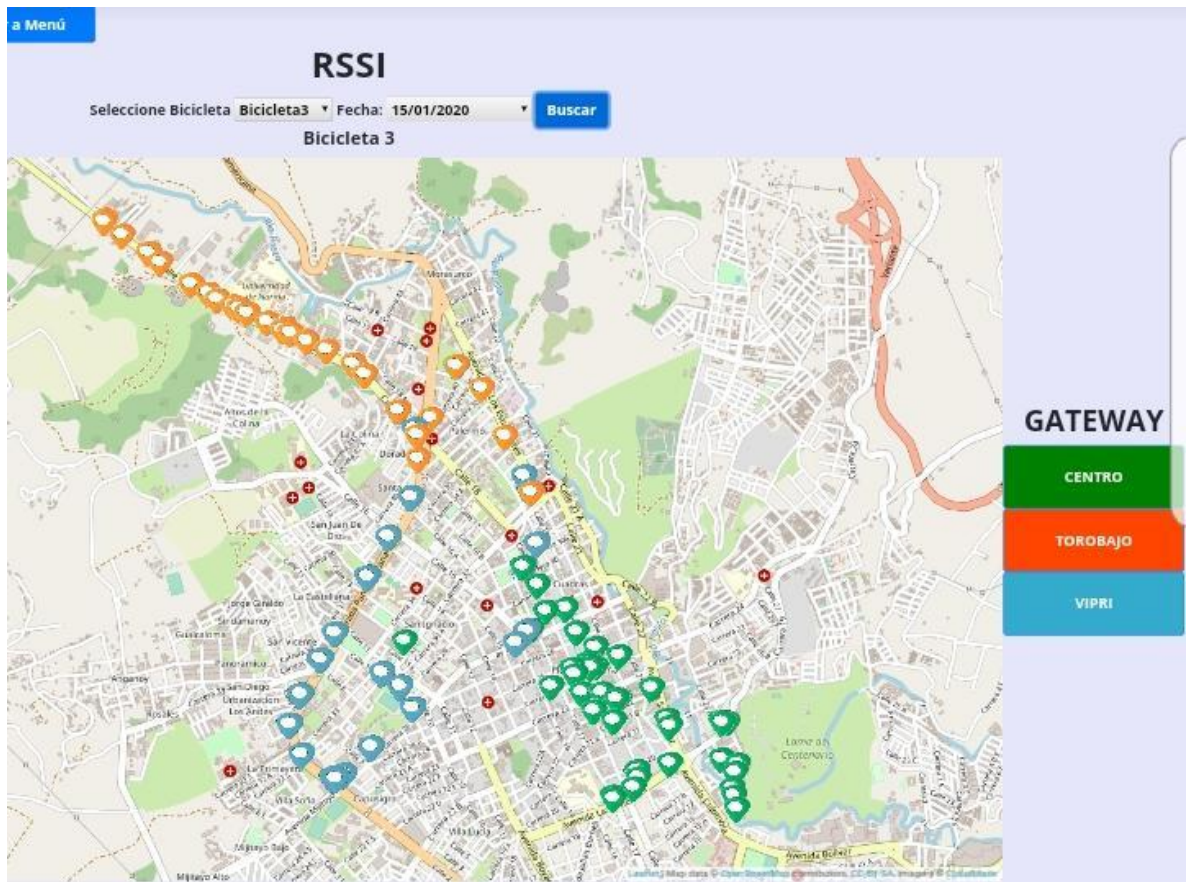
Finalmente se ubicaron los Gateways en sus posiciones definitivas, que corresponden a los 3 campus de la Universidad de Nariño, Torobajo, Centro y Vipri, como se muestra en la Figura 49.

Se realizó un recorrido por los sectores cercanos a cada campus de la Universidad de Nariño, para evaluar el funcionamiento de la red en su nueva ubicación como se muestra en la Figura 50.

**Figura 49.** Ubicación de Gateways en campus universitarios.



Fuente: Software Leaflet.



**Figura 50.** Recorrido por campus universitarios.

**Fuente:** Software Leaflet.

Con los datos de la prueba anterior se realizó el cálculo de la distancia máxima del enlace de comunicación, tomando los datos de RSSI [dBm] y de distancia [log (Km)] entre los puntos y el Gateway que recibió la información, se encontró la regresión lineal entre estos, obteniendo la ecuación:

$$P_{Rx} = -27.87 \log(d) - 108.9, \quad (3)$$

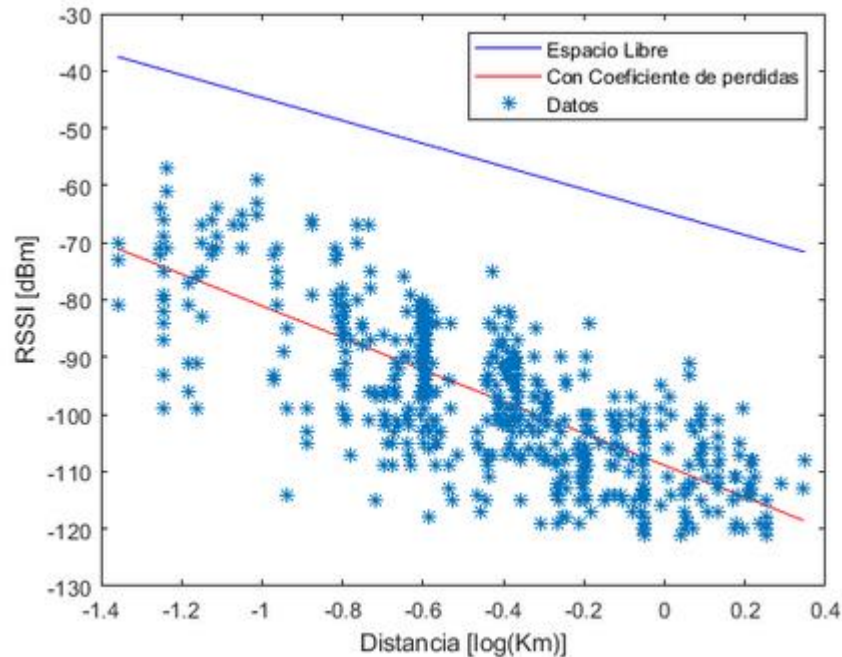
donde  $P_{Rx}$  es la potencia recibida en función del logaritmo de la distancia (d). En la ecuación (3) se reemplazó el valor de  $P_{Rx}$  por -121dBm que fue el valor mínimo de RSSI recibido por el Gateway, para encontrar la distancia máxima asociada a esta potencia, obteniendo:

$$d_{max} = 10^{\frac{-121+108.9}{-27.86}} \quad (4)$$

$$d_{max} = 2.71 \text{ Km.} \quad (5)$$

El valor de la ecuación (5) representa un aproximado de la distancia promedio en que se tiene cobertura respecto a cada Gateway. En la Figura 51 se comparó la recta de la ecuación (3) con la recta del balance de enlace en espacio libre, donde se notó la diferencia entre datos reales e ideales, debido a que la pendiente de los datos medidos es mayor, es decir la potencia de recepción decae más rápido al aumentar la distancia.

**Figura 51.** Regresión lineal de potencia recibida.



**Fuente:** Software Matlab.

#### 4. CONCLUSIONES

Haciendo parte de una comunidad IoT de cooperación mundial *The Things Network*, se desarrolló una red y un servidor que permitió integrar otras aplicaciones de diferentes campos de investigaciones, donde fue necesario un continuo monitoreo y almacenamiento de información, en lugares donde fue difícil el acceso a redes de datos convencionales.

A pesar de ser una tecnología nueva y los resultados se basaron en datos empíricos, LoRa presentó resultados satisfactorios en aspectos como consumo, alcance recibiendo datos en distancias hasta de 3.8 Km con línea de vista y RSSI de -96 y confiabilidad con un promedio de 2 paquetes correctos por minuto, por ello se estableció una red basada en esta tecnología.

La elección del método de autenticación de la tecnología LoRa varía según el tipo de aplicación, para este proyecto la opción más conveniente fue el método ABP, el cual presentó mejores resultados ante el cambio de posición.

El protocolo de comunicación Web-Socket, proporcionó una comunicación estable entre la información de la base de datos y los aplicativos web, permitiendo una visualización rápida, accesible y agradable para el usuario.

Al pertenecer a la comunidad IoT de *The Things Network*, se compartió datos del proyecto con personas de todo el mundo pertenecientes a la plataforma, para esto fue necesario la existencia de una comunidad oficial con 8 miembros inscritos y más de un Gateway registrado, aumentando el prestigio del proyecto.

## **5. RECOMENDACIONES**

El método de autenticación OTAA presenta mejores condiciones de seguridad, por tanto, se recomienda su uso en aplicaciones estáticas, en las cuales el nodo y el Gateway se encuentren en línea de vista todo el tiempo.

El sistema de unidad de captura funciona de manera constante alimentándose de la bicicleta, aunque su consumo es bajo, en casos de que la bicicleta no se encuentre en circulación por periodos considerables, se sugiere desconectar el sistema para no realizar una descarga innecesaria de la batería.

La bicicleta se encuentra expuesta a diferentes condiciones externas, por esto es necesario realizar una calibración previa a los sensores y dispositivos de medición que posee la bicicleta, con el fin de obtener datos correctos.

Los resultados obtenidos en el sistema de monitoreo, se deben tener en cuenta a la hora de implementar la red planteada por la contraparte de este proyecto, porque brindan información importante para determinar la ubicación de los equipos en la que se obtenga una mayor cobertura de sectores de la ciudad de San Juan de Pasto.

Debido a que se cubrió con buena resolución la muestra de gases de CO<sub>2</sub>, los resultados almacenados en la base de datos pueden ser empleados en futuras investigaciones de tipo ambiental. Además de brindar estadísticas de consumo del funcionamiento de estos vehículos eléctricos.

Según el análisis de los resultados obtenidos, se plantea que, para lograr una cobertura de la ciudad de San Juan de Pasto, se sugiere la ubicación estratégica de 4 gateways, en sectores con una altura y línea de vista apropiada.

Para que el sistema funcione continuamente es necesario que personal capacitado, se encargue de una revisión periódica de las unidades de captura y de la base de datos, debido a que se pueden presentar fallos por falta de mantenimiento.

## BIBLIOGRAFÍA

RIVERA, Nicolas. "Qué es el Internet of Things y cómo cambiará nuestra vida en el futuro". {En línea}. {14 marzo de 2019} disponible en: <https://hipertextual.com/2015/06/internet-of-things>.

TECNOLOGIA LORA Y LORAWAN: Tecnología LoRa y LoRaWAN {en línea}. Catsensors. {15 marzo de 2019} disponible en: <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan>.

CONCEPTOS BÁSICOS DE REDES: Términos de la unidad {en línea}. Quizlet. {15 marzo de 2019} disponible en: <https://quizlet.com/es/249490101/conceptos-basicos-de-redes-flash-cards/>  
GIMENEZ, Tamara y ROS, María. "SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)". {En línea}. {15 marzo de 2019} disponible en: [https://webs.um.es/bussons/GPSresumen\\_TamaraElena.pdf](https://webs.um.es/bussons/GPSresumen_TamaraElena.pdf)

ELVIR, Dilcia. "SISTEMAS DE TIEMPO REAL". {En línea}. {15 marzo de 2019} disponible en: <https://docs.google.com/document/edit?id=1QWdZfVdJ9WndrUfces9AqVAvZTtZOHpVUegPeql3SRQ&hl=en>

DELGADO, César. "Servidor Web". {En línea}. {15 marzo de 2019} disponible en: [https://upanama.e-ducativa.com/archivos/repositorio/6000/6126/html/59\\_servi.htm](https://upanama.e-ducativa.com/archivos/repositorio/6000/6126/html/59_servi.htm)

SUPPORT: THE THINGS NETWORK SUPPORT. {En línea}. Thethingsnetwork. {15 marzo de 2019} disponible en: <https://www.thethingsnetwork.org/support>

MATHWORKS: ThingSpeak. {En línea}. Mathworks. {15 marzo de 2019} disponible en <https://la.mathworks.com/help/thingspeak/>

CASTRO, Nesly; CHAMORRO, Luis y VITERI, Carlos. Diseño e implementación de una red de sensores inalámbricos para el control de riegos por goteo. Pasto, 2012. Trabajo de grado (ingeniería electrónica). Universidad de Nariño. Facultad de Ingeniería. Departamento de Electrónica.

BURBANO, Luis; PANTOJA, Romer y FAJARDO, Dario. Diseño e implementación de un dispositivo portátil para el control de humedad en cultivo de maíz fertirrigado bajo el invernadero de la facultad de ciencias agrícolas de la Universidad de Nariño. Pasto, 2013. Trabajo de grado (ingeniería electrónica). Universidad de Nariño. Facultad de Ingeniería. Departamento de Electrónica.

GUERRERO, Felipe; RUBIO, Fabio y FAJARDO, Dario. Implementación de una granja digital mediante el internet de las cosas IoT. Pasto, 2018. Trabajo de grado (ingeniería electrónica). Universidad de Nariño. Facultad de Ingeniería. Departamento de Electrónica.

MUHAMMAD, Omer y DIRK, Pesch. Analyzing LoRa: A Use Case Perspective. En: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). Vol.; 4. (Feb.2018); p. 355-360.

HSIANG, Kai; WEN, Qi; JIE, Guan, HAN, Jun y CHEN, Huang. Demo Abstract: A LoRa Wireless Mesh Networking Module for Campus-Scale Monitoring. En: 2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). Vol.; 16. (Abr.2017); p. 259-260.

WU, Fan; RUDIGER, Christoph; REDOUTE, Jean-Michel y YUCE, Mehmet. WE-Safe: A Wearable IoT Sensor Node for Safety Applications via LoRa. En: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). Vol.; 4. (Feb.2018); p. 144-148.

CHOI, Chang-Sic; JEONG Jin-Doo; LEE, Il-Woo y PARK, Wan-Ki. LoRa based Renewable Energy Monitoring System with Open IoT Platform. En: 2018 International Conference on Electronics, Information, and Communication (ICEIC). Vol.; 1. (Ene.2018); p. 184-185.

TRIANA, Jordan y RODRIGUEZ, Ronald. Prototipo de solución IoT con tecnología “lora” en monitoreo de cultivos agrícolas. Bogotá, 2018. Trabajo de grado (ingeniería en telecomunicaciones). Universidad Distrital Francisco José de Caldas. Facultad de Tecnología. Departamento de Telecomunicaciones.

LORAWAN: LoRaWAN Architecture. {En línea}. Thethingsnetwork. {15 marzo de 2019} disponible en: <https://www.thethingsnetwork.org/docs/lorawan/architecture.html>

ARANGO, María. “Lista la resolución para la subasta del espectro en Colombia”. {En línea}. {18 octubre de 2019} disponible en: <https://www.eltiempo.com/tecnosfera/novedades-tecnologia/mintic-publica-resolucion-para-la-subasta-del-espectro-radioelectrico-421954>

AZAUSTRE, Carlos. Aprendiendo JavaScript. Madrid: abril de 2016. {18 octubre de 2018} disponible en: [https://play.google.com/store/books/details?id=cnjhCwAAQBAJ&rdid=book-cnjhCwAAQBAJ&rdot=1&source=gbs\\_vpt\\_read&pcampaignid=books\\_booksearch\\_viewport](https://play.google.com/store/books/details?id=cnjhCwAAQBAJ&rdid=book-cnjhCwAAQBAJ&rdot=1&source=gbs_vpt_read&pcampaignid=books_booksearch_viewport)

CONFIGURING THE GATEWAY. {En línea}. Rakwireless. {15 marzo de 2019} disponible en: <https://doc.rakwireless.com/rak2245-pi-hat-edition---lorawan---gateway-concentrator-module/configuring-the-gateway>

MONGODB: La base de datos líder para aplicaciones modernas. {En línea}. MongoDB. {15 marzo de 2019} disponible en: <https://www.mongodb.com/es>

GUERRERO, José. “Distancia entre dos puntos de la superficie terrestre mediante la fórmula de Haversine con Python”. {En línea}. {18 marzo de 2019} disponible en: <https://joseguerreroa.wordpress.com/2012/11/13/distancia-entre-dos-puntos-de-la-superficie-terrestre-mediante-la-formula-de-haversine-con-python/>

## ANEXOS

### Anexo 1. Especificaciones técnicas de los equipos utilizados en el proyecto.

#### 1.1 Dragino lg01-p:

El LG01-P Puerta de Enlace LoRa 915 MHz (América) es un Gateway de canal abierto de código abierto que le permite conectar una red inalámbrica a una red IP a través de WiFi, Ethernet, 3G o 4G celular. La conexión inalámbrica LoRa permite a los usuarios enviar datos y alcanzar rangos extremadamente largos a bajas velocidades de datos. Proporciona comunicación de amplio alcance de espectro ultra largo y alta inmunidad a las interferencias.

LG01-P tiene una interfaz WiFi, un puerto Ethernet y un puerto host USB. Estas interfaces proporcionan métodos flexibles para que los usuarios conecten sus redes de sensores a Internet. LG01-P ejecuta el sistema Open Source OpenWrt, el usuario puede modificar libremente el archivo fuente o compilar el sistema para admitir sus aplicaciones personalizadas.



**Figura 1.** Dragino lg01-p.

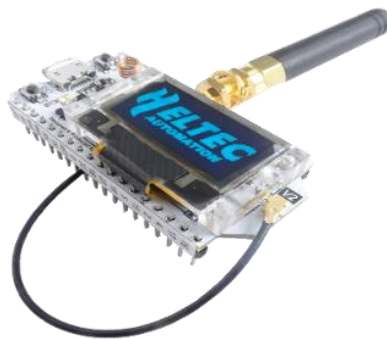
**Tabla 1.** Especificaciones Dragino lg01-p.

| Especificaciones técnicas                          |                         |                         |
|----------------------------------------------------|-------------------------|-------------------------|
| Puerta de enlace de canal único de código abierto. |                         |                         |
| Interfaz WiFi, puerto Ethernet y puerto host USB   |                         |                         |
| Conexión a Internet a través de LAN, WiFi, 3G o 4G |                         |                         |
| Arduino IDE compatible                             |                         |                         |
| Alcance máximo en LoRa: 5 ~ 10km                   |                         |                         |
| Procesador Linux 400 MHz, 24K MIPS                 |                         | Flash: 16MB; RAM: 64 MB |
| MCU / LoRa Side:<br>ATMega328P                     | Flash: 32 KB, RAM: 2 KB | Chip LoRa: SX2176 / 78  |

**Fuente:** <https://www.dragino.com/products/lora/item/119-lg01-s.html>

## 1.2 Wi-Fi LoRa 32:

Wi-Fi LoRa 32 es una placa de desarrollo IoT clásica diseñada y producida por Heltec Automation (TM), es un producto altamente integrado basado en ESP32 + SX127x, tiene funciones de Wi-Fi, BLE, LoRa, también cuenta con sistema de gestión de batería Li-Po y una pantalla de 0.96 pulgadas OLED . Es la mejor opción para ciudades inteligentes, granjas inteligentes, hogares inteligentes y fabricantes de IoT.



**Figura 2.** Wi-Fi LoRa 32

**Tabla 2.** Especificaciones Wi-Fi LoRa 32

| Especificaciones técnicas.                                                                            |
|-------------------------------------------------------------------------------------------------------|
| MCU ESP32( 240MHz Tensilica LX6 dual-core + 1 ULP, 600 DMIPS, 520KB SRAM, Wi-Fi, dual mode Bluetooth) |
| LoRa chip SX1276 (868 and 915 version), SX1278 (433 and 470 version)                                  |
| 12 – bits ADC                                                                                         |
| 8 – bits DAC                                                                                          |
| 29 general GPIO                                                                                       |
| WIFI scan 115Ma                                                                                       |
| WIFI AP 135Ma                                                                                         |
| LoRa 10dB emission 50Ma                                                                               |
| LoRa 12dB emission 60Ma                                                                               |
| LoRa 15dB emission 110Ma                                                                              |
| LoRa 20dB emission 130mA LoRa                                                                         |

**Fuente:** <https://heltec.org/project/wifi-lora-32/>

### 1.3 RAK2245 Pi HAT Edition:

RAK2245 Pi HAT es el más grande de la familia RAK2245, sin embargo, ofrece compatibilidad completa con la Raspberry Pi. Por lo tanto, es una solución perfecta para una puerta de enlace de DIY que utiliza la Raspberry Pi en su núcleo. Esto permite implementar rápidamente una solución LoRaWAN con un bajo costo y un rendimiento estable. Al igual que con los otros módulos de la línea RAK2245, se basa en el Semtech 1301 y los chips frontales SX1257 / 58 duales. Es un concentrador LoRa de 8 canales completo con una serie de mejoras sobre la última generación (RAK831) de este producto.



**Figura 3.** RAK2245 – Pi Hat.

**Tabla 3.** Especificaciones RAK2245-Pi Hat

| Especificaciones técnicas.                                                             |
|----------------------------------------------------------------------------------------|
| Tx Power hasta 27dBm                                                                   |
| Sensibilidad RX hasta -139dBm                                                          |
| Resistencia al ruido mejorada debido al filtrado de cadena RF mejorado (SAW adicional) |
| Soporte completo de pila LoRaWAN 1.0.2                                                 |
| Bandas compatibles: (EU433, CN470, IN865, EU868, AU915, US915, KR920, AS920 y AS923)   |
| Módulo integrado de GPs Ublox MAX-7Q                                                   |
| Encabezado hembra de 40 pines (totalmente compatible con Raspberry Pi)                 |
| Diseño de calor rediseñado para una mejor disipación térmica y ruido térmico reducido  |

**Fuente:** <https://store.rakwireless.com/products/rak2245-pi-hat>

### Raspberry Pi 3 Model B+:

Procesador de cuatro núcleos a 1,4 GHz de 64 bits, LAN inalámbrica de doble banda, Bluetooth 4.2 (BLE), Ethernet más rápido y soporte de alimentación por Ethernet (con PoE HAT separado)



**Figura 4.** Raspberry Pi 3 Model B+.

**Tabla 4.** Especificaciones Raspberry Pi 3 Model B+.

| Especificaciones técnicas.                                                        |
|-----------------------------------------------------------------------------------|
| Broadcom BCM2837B0, Cortex-A53 (ARMv8) SoC de 64 bits a 1.4GHz                    |
| SDRAM LPDDR2 de 1 GB                                                              |
| LAN inalámbrica IEEE 802.11.b / g / n / ac de 2.4GHz y 5GHz, Bluetooth 4.2, BLE   |
| Gigabit Ethernet sobre USB 2.0 (rendimiento máximo 300 Mbps)                      |
| Cabecera GPIO extendida de 40 pines                                               |
| HDMI de tamaño completo                                                           |
| 4 puertos USB 2.0                                                                 |
| Puerto Micro SD para cargar su sistema operativo y almacenar datos                |
| Entrada de alimentación de 5V / 2.5A DC                                           |
| Compatibilidad con alimentación por Ethernet (PoE) (requiere un PoE HAT separado) |

**Fuente:** <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>

#### 1.4 Antena omnidireccional LoRa de fibra de vidrio 806-960MHz:



**Figura 5.** Antena omnidireccional de 806-960MHz.

**Tabla 5.** Especificaciones técnicas antena omnidireccional

| Especificaciones técnicas                 |
|-------------------------------------------|
| Ancho de banda 806 – 960 MHz              |
| Antena en fibra de vidrio omnidireccional |
| Alta ganancia 12dBi                       |
| Base estación antena GSM para exterior    |
| Impedancia de entrada 50 $\Omega$         |

**Fuente:** <https://es.aliexpress.com/i/32675964450.html>

### 1.5 GPS NEO 6M:

El GPS NEO 6M, puede funcionar con un voltaje de alimentación en el rango de 3V a 5V, mientras que las señales que entran y salen del módulo son de 3.3V, por lo que se requiere un convertidor de niveles lógicos si un micro controlador de 5V va a comunicarse con módulo GPS. Si solamente se desean recibir los datos NMEA basta con conectar el pin TX con el RX del micro controlador y recibir los datos que envía el módulo.



**Figura 6.** GPS NEO 6M.

**Tabla 6.** Especificaciones GPS NEO 6M.

| Especificaciones técnicas.                      |
|-------------------------------------------------|
| Voltaje de Alimentación DC: 3.5V a 5V           |
| Comunicación serial                             |
| Antena cerámica activa incluida                 |
| LED indicador de señal                          |
| Baudios por defecto: 9600                       |
| EEPROM para guardar configuración de parámetros |

|                                        |
|----------------------------------------|
| Sistema de coordenadas: WGS-84         |
| Sensibilidad de captura -148dBm        |
| Sensibilidad de rastreo: -161 dBm      |
| Máxima altura medible: 18000           |
| Máxima velocidad 515 m/s               |
| Exactitud: 1 micro segundo             |
| Frecuencia receptora: L1 (1575.42 Mhz) |

**Fuente:** <https://ferretronica.com/products/modulo-gps-neo-6m-v2>

### 1.6 TTGO t-beam ESP32:

La placa está construida alrededor de un chip ESP32 de doble núcleo, con 4 MB de flash SPI integrado, que proporciona Wi-Fi y Bluetooth LE a través de una "antena 3D" en la PCB. El soporte LoRa viene en tres variantes diferentes, operando a 433MHz, 868MHz y 915MHz dependiendo de la región, con una antena SMA incluida. Mientras que el rastreo de ubicación es proporcionado por el módulo GPS NEO-6M con antena de cerámica.



**Figura 7.** TTGO t-beam ESP32.

| Especificaciones técnicas.                                                        |
|-----------------------------------------------------------------------------------|
| Voltaje de funcionamiento: 1,8-3,7 v                                              |
| Corriente aceptable: 10-14 ma                                                     |
| Corriente de transmisión: 120mA @ + 20dBm                                         |
| 90mA @ + 17dBm                                                                    |
| 29mA @ + 13dBm                                                                    |
| Frecuencia de funcionamiento: 433/470 MHz (versión 433HMz) /868 M/915 M (868 MHz) |
| Potencia de transmisión: + 20dBm                                                  |
| Sensibilidad de recepción: -139dBm @ LoRa & 62,5 Khz & SF = 12 y 146bps           |
| -136dBm @ LoRa y 125 Khz y SF = 12 y 293bps                                       |
| -118dBm @ LoRa y 125 Khz y SF = 6 y 9380bps                                       |
| -123dBm @ FSK y 5 Khz y 1,2 Kbps                                                  |

**Tabla 7.** Especificaciones TTGO t-beam ESP32.

**Fuente:** <https://es.aliexpress.com/w/wholesale-ttgo-esp32.html>

### 1.7 CONVERTOR REDUCTOR BUCK DC-DC (CN LM2596HV):

Este módulo está basado en el Regulador DC-DC Step Down LM2596 que es un circuito integrado monolítico adecuado para el diseño fácil y conveniente de una fuente de conmutación tipo buck. Es capaz de conducir una corriente de hasta 3A. Maneja una carga con excelente regulación de línea y bajo voltaje de rizado. Este dispositivo está disponible con voltaje de salida ajustable. El módulo reduce al mínimo el uso de componentes externos para simplificar el diseño de fuentes de alimentación.



**Figura 8.** Conversor reductor buck dc-dc.

| Especificaciones técnicas.                              |
|---------------------------------------------------------|
| Basada en el regulador LM2596, salida entre 1,5 y 60Vdc |
| Voltaje de entrada: 4.5-60V                             |
| Voltaje de salida: (Adjustable)                         |
| Corriente de salida: Maxima 3 <sup>a</sup>              |
| Dimensiones: 43*20*14mm                                 |
| Frecuencia de switching: 150 KHz                        |

**Tabla 8.** Conversor reductor buck dc-dc.

**Fuente:** <https://ssdielect.com/en/conversores-dc-dc/1107-cn-lm2596hv.html>

### **1.8 Xiaomi Banco de energía 20000mAh:**

Banco de baterías que puede alimentar dispositivos electrónicos a 5 voltios, con capacidad de carga y descarga simultánea.



**Figura 9.** Xiaomi banco de energía.

**Tabla 9.** Especificaciones técnicas banco de poder Xiaomi.

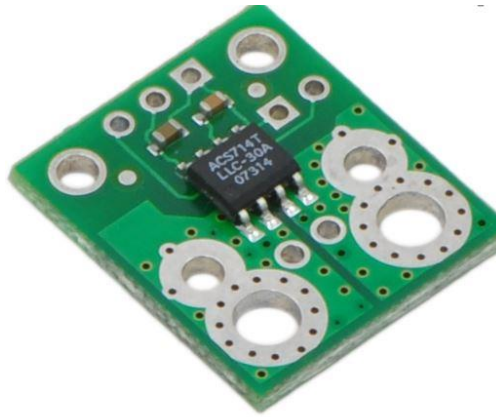
| Especificaciones técnicas.                                                                                                                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Diseñado para una corriente de entrada bidireccional de -30 A a 30 A (aunque el robusto sensor IC puede sobrevivir hasta cinco veces la condición de sobre corriente).                                                                                                                                                  |
| La resistencia interna del camino conductivo es típicamente de 1.2 mΩ, y la PCB está hecha con cobre de 2 oz, por lo que se pierde muy poca energía en la placa.                                                                                                                                                        |
| El uso de un sensor de efecto Hall significa que el IC puede aislar eléctricamente la ruta de corriente de la electrónica del sensor (hasta 2.1 kV RMS), lo que permite que el sensor se inserte en cualquier lugar a lo largo de la ruta de corriente y se use en aplicaciones que requieren electricidad aislamiento. |
| Alta precisión y fiabilidad: error típico de salida total de $\pm 1.5\%$ a temperatura ambiente con calibración de fábrica, un voltaje de compensación de salida extremadamente estable y una histéresis magnética casi nula.                                                                                           |

**Fuente:** <https://www.amazon.es/Xiaomi-Power-Bank-Mi-20-000mAh/dp/B078K1CHB5>

### 1.9 ACS711 Current Sensor Carrier:

Esta placa es una simple portadora del sensor de corriente lineal basado en efecto Hall  $\pm 30A$  ACS711 de Allegro, que ofrece una ruta de corriente de baja resistencia ( $\sim 1.2 \text{ m}\Omega$ ) y aislamiento eléctrico de hasta 2.1 kV RMS. Esta versión acepta una entrada de corriente bidireccional con una magnitud de hasta 30 A y emite un voltaje analógico proporcional (66 mV / A) centrado a 2.5 V con un error típico de  $\pm 1.5\%$ . Funciona de 4.5 V a 5.5 V y está diseñado para su uso en sistemas de 5 V

| Especificaciones técnicas.                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Capacidad de 20.000 mAh                                                                                                                                                         |
| Salida de doble USB interfaces                                                                                                                                                  |
| Baterías de polímero de litio de alta densidad                                                                                                                                  |
| Carga bidireccional rápida                                                                                                                                                      |
| Totalmente compatible con Xiaomi / Apple / Samsung/ Huawei/ HTC/ Meizu/ LG y otros teléfonos móviles principales, así como las tabletas, también puede cargar el último MacBook |



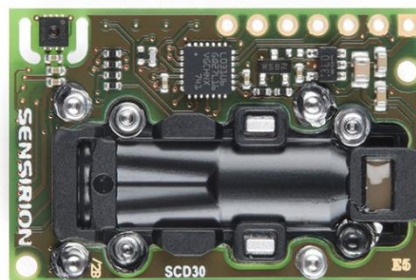
**Figura 10.** Sensor de corriente ACS711.

**Tabla 10.** Especificaciones técnicas sensor de corriente ACS711.

**Fuente:** <https://www.pololu.com/product/2452>

#### **1.10 Sensor de CO<sub>2</sub>, Humedad y Temperatura - SCD30:**

El SCD30 de Sensirion es un sensor de CO<sub>2</sub> basado en infrarrojos no dispersivos (NDIR) de alta calidad capaz de detectar de 400 a 10000 ppm con una precisión de  $\pm (30 \text{ ppm} + 3\%)$ . Para mejorar la precisión, el SCD30 tiene sensores de temperatura y humedad integrados, así como comandos para establecer la altitud actual. Para mayor precisión, el SCD30 también acepta lecturas de presión ambiental.



**Figura 11.** Sensor SCD30 de CO<sub>2</sub>.

**Tabla 11.** Especificaciones técnicas sensor SCD30 de CO<sub>2</sub>.

| Especificaciones técnicas.                              |
|---------------------------------------------------------|
| Tensión de alimentación: 3.3V - 5.5V                    |
| Tecnología de sensor de CO2 NDIR                        |
| Sensor integrado de temperatura y humedad.              |
| Detección de doble canal para una estabilidad superior. |
| Rango de medición: 400 ppm - 10,000 ppm                 |
| Precisión: $\pm (30 \text{ ppm} + 3\%)$                 |
| Consumo de corriente: 19 mA @ 1 medida cada 2 s.        |
| Totalmente calibrado                                    |
| Interfaz digital UART o I2C                             |

**Fuente:** <https://www.sensirion.com/environmental-sensors/carbon-dioxide-sensors-co2/>

### 1.11 Temtop M2000:

Monitor de calidad del aire para PM2.5 PM10, partículas CO2 y HCHO, temperatura, humedad, cuenta con una alarma de audio ajustable, grabación de datos y fácil Calibración.



**Figura 12.** Sensor Temtop M2000.

**Tabla 12.** Especificaciones técnicas sensor Temtop M2000.

| Especificaciones técnicas.                           |
|------------------------------------------------------|
| Pantalla: pantalla LCD a color TFT                   |
| Voltaje de la batería: 3.7 VDC                       |
| Rango de temperatura: 0 ~ 50 °C                      |
| Rango de humedad: 0 ~ 90% HR                         |
| Rango de medición PM2.5: 0 ~ 999 ug / m <sup>3</sup> |
| Resolución: 0.1 ug / m <sup>3</sup>                  |
| Rango de medición PM10: 0 ~ 999 ug / m <sup>3</sup>  |
| Resolución: 0.1ug / m <sup>3</sup>                   |
| Rango de medición Co2: 0 ~ 5000 PPM                  |
| Resolución: 1 PPM                                    |

**Fuente:** <https://www.amazon.com/Temtop-Particles-Temperature-Recording-Calibration/dp/B07DHYG2VK>

## Anexo 2. Código de adquisición de datos.

```
float Voltaje(){
    float voltaje=0;
    for (int j=0; j<5; ++j) {
        sensorValue = analogRead(analogInPin);
        delay(20);
        voltaje = voltaje + sensorValue;
    }
    voltaje=voltaje/5;
    Serial.println(sensorValue);
    return voltaje;
}

float Corriente(){
    float corriente=0;
    for (int j=0; j<5; ++j) {
        sensorValue2 = analogRead(analogInPin2);
        delay(20);
        corriente = corriente + sensorValue2;
    }
    corriente = corriente/5;
    Serial.println(sensorValue2);
    if(corriente < 0){
        return 0;
    }
    else
    {
        return corriente;
    }
}
```

**Figura 1.** Lectura de voltaje y corriente.

```
void Co2(){
    if (airSensor.dataAvailable())
    {
        Serial.print("co2(ppm):");
        co=airSensor.getCO2();
        Serial.print(co);

        Serial.print(" temp(C):");
        temp =airSensor.getTemperature();
        Serial.print(temp, 2);

        Serial.print(" humidity(%):");
        hum =airSensor.getHumidity();
        Serial.print(hum, 2);

        Serial.println();
    }
}
```

**Figura 2.** Lectura de CO<sub>2</sub>.

Fuente: [https://github.com/sparkfun/SparkFun\\_SCD30\\_Arduino\\_Library](https://github.com/sparkfun/SparkFun_SCD30_Arduino_Library)

### Anexo 3. Menú principal aplicativo web.

 **BICICLETAS  
campus  
VERDE**  
Universidad de Nariño

**Ingresa Tus Datos**

Usuario

Clave

**Figura1.** Ingreso al servidor

 **BICICLETAS  
campus  
VERDE**  
Universidad de Nariño

**Sistema de Monitoreo Bicicletas Campus Verde**

|          |                   |      |
|----------|-------------------|------|
| Tracking | Trayectorias      | Co2  |
| Sensores | Cobertura Gateway | RSSI |

**Figura 2.** Menú de selección de aplicativos.