



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

CARRERA DE INGENIERÍA ELECTRÓNICA

**DISEÑO DE UNA PLATAFORMA DE MONITOREO DE NIVELES DE CO₂
BASADA EN IOT**

Trabajo de titulación previo a la obtención del
título de Ingeniero Electrónico

AUTOR: JUAN DANIEL PÁEZ CARABAJO

TUTOR: ING. JUAN PAÚL INGA ORTEGA, MgT.

Cuenca – Ecuador

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Juan Daniel Páez Carabajo con documento de identificación N° 0106022247 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 29 de agosto del 2023

Atentamente,



Juan Daniel Páez Carabajo

0106022247

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO
DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Juan Daniel Páez Carabajo con documento de identificación No. 0106022247, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico con enfoque investigativo: "Diseño de una plataforma de monitoreo de niveles de CO2 basada en IoT", el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana..

Cuenca, 29 de agosto del 2023

Atentamente,



Juan Daniel Páez Carabajo

0106022247

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Juan Paúl Inga Ortega con documento de identificación N° 0104166491, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO DE UNA PLATAFORMA DE MONITOREO DE NIVELES DE CO2 BASADA EN IOT, realizado por Juan Daniel Páez Carabajo con documento de identificación N° 0106022247, obteniendo como resultado final el trabajo de titulación bajo la opción proyecto técnico con enfoque investigativo que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 29 de agosto del 2023

Atentamente,



Ing. Juan Paúl Inga Ortega, MgT.

01041664941

AGRADECIMIENTOS

Quiero expresar mis sincero agradecimiento a todas las personas que contribuyeron de manera significativa en la realización de esta investigación y en la culminación de este trabajo.

En primer lugar quiero agradecer a mi asesor académico, Ing. Juan Inga, por su orientación, paciencia a lo largo de este trabajo.

Agradezco a mis padres, Jovanny y Silvia, por su incansable apoyo emocional y financiero. Su confianza en mí y su aliento constante me motivaron a superar los desafíos y perseguir mis metas académicas.

Finalmente, quiero agradecer a María Fernanda por su contante comprensión, paciencia y aliento durante este camino.

A cada uno de ustedes, mi más sincero agradecimiento. Sin su apoyo y contribuciones, este logro no habría sido posible

DEDICATORIA

A mis padres y hermanos, quienes siempre creyeron en mí y me brindaron su amor y apoyo incondicional a lo largo de este arduo camino. Su constante aliento fue mi motor para alcanzar esta meta.

A mis profesores y asesores, cuya sabiduría y guía fueron fundamentales en cada etapa de este trabajo. Sus conocimientos y paciencia me han ayudado a crecer tanto como estudiante como persona.

Esta tesis está dedicada a cada uno de ustedes, con profundo agradecimiento y cariño.

Índice general

Índice General	III
Índice de figuras	VII
Índice de tablas	VIII
Resumen	IX
Abstract	XI
Antecedentes o Problema de Estudio	1
Justificación	4
Objetivos	6
Introducción	7
1. Revisión bibliográfica	8
1.1. Estado del arte del monitoreo de CO ₂ basado en IoT	8
1.1.1. Países que más han contribuido	9
1.1.2. Países de América Latina que han contribuido	9
1.2. Fundamentos Teóricos	10
1.2.1. Dióxido de carbono	10
1.2.2. Efectos de CO ₂ en el ser humano	11
1.2.3. Concentración de CO ₂ en ambientes abiertos y cerrados	11
1.2.4. Tiempo de muestro para medir una variable de la calidad del aire	12
1.2.5. Internet de las cosas (IoT)	12

1.2.6.	Red de sensores Inalámbricos (WSN)	14
1.2.7.	Longe Range (LoRa)	16
1.2.8.	LoRaWAN	17
1.2.9.	Chirpstack	19
1.2.10.	Frameworks para desarrollo de aplicaciones móviles	19
1.2.11.	Placas de desarrollo	20
1.2.12.	Sensores utilizados en monitoreo de la calidad del aire	24
2.	Desarrollo de la Plataforma de Monitoreo de Niveles de CO₂	27
2.1.	Diseño de nodos de monitoreo	27
2.1.1.	Placa de desarrollo, módulos y sensores	27
2.1.2.	Protocolo de comunicación del sensor MHZ-19B	33
2.1.3.	Caja de aplicación IoT	35
2.1.4.	Placa PCB	36
2.1.5.	Comunicación entre la placa de desarrollo HTCC-AB 01 hacia los sensores de gas de CO ₂	36
2.1.6.	Alimentación del circuito	38
2.2.	Etapa Comunicación LoRa	39
2.2.1.	Topología de red	39
2.2.2.	Registro de Nodos LoRa con ChirpStack	40
2.2.3.	Creación de claves API globales	43
2.2.4.	Comunicación LoRa	44
2.3.	Front-end y Back-end de aplicación de monitoreo	45
2.3.1.	Front-end	45
2.3.2.	Modo Usuario	46
2.3.3.	Modo Administrador	48
2.4.	Desarrollo de Servidor	50
2.4.1.	API REST del servidor de aplicaciones ChirpStack	50
2.4.2.	Back-End	51
2.4.3.	Máquina virtual en Google Cloud	53
2.5.	Desarrollo de Aplicación Móvil	54
2.5.1.	Librerías utilizadas	56
2.5.2.	Widgets de aplicación Flutter	56

2.5.3. Diagrama de clases	59
2.6. Base de datos PostgreSQL	61
2.7. Nombre y icono de la aplicación de monitoreo	61
3. Implementación y Pruebas de Operación de la Plataforma de Monitoreo	63
3.1. Pruebas de Operación: Nodos de CO ₂	63
3.1.1. Comunicación Chirpstack	63
3.1.2. Colocación de nodos	64
3.1.3. Recepción de datos en ChirpStack	66
3.2. Pruebas de operación de servidor	66
3.3. Pruebas de operación de aplicación móvil: Modo Usuario	67
3.3.1. Ubicación de sensores	68
3.3.2. Visualización de valores de CO ₂	69
3.4. Pruebas de operación de aplicación móvil: Modo Administrador	71
3.4.1. Inicio de sesión	71
3.4.2. Cambio de ubicación de nodos de CO ₂	72
3.4.3. Datos en el servidor	73
3.5. Análisis de Resultados	74
3.6. Análisis de Costos	78
4. Conclusiones y Trabajos Futuros	82
4.1. Conclusiones	82
4.2. Trabajos futuros	84
Glosario	86
Referencias	93

Índice de figuras

1.	Trabajos Previos [Fuente: El Autor.]	5
1.1.	Países que más han contribuido [Fuente: El Autor.]	9
1.2.	Países de América Latina que han contribuido [Fuente: El Autor.]	10
1.3.	Topología Estrella. [Fuente: El Autor.]	15
1.4.	Topología Malla. [Fuente: El Autor.]	16
1.5.	Topología Híbrida Estrella-Malla. [Fuente: El Autor.]	16
1.6.	Clases de LoRaWAN. [Fuente: El Autor.]	18
2.1.	Cubecell HTCC-AB 01	28
2.2.	SeedStudio Lipo Rider Pro [Fuente: El Autor.]	29
2.3.	Sensor de gas MG811. [Fuente: El Autor.]	30
2.4.	Sensor de gas MHZ-19B. [Fuente: El Autor.]	33
2.5.	Vista lateral y superior de la caja. [Fuente: El Autor.]	35
2.6.	Vista superior de la tapa para la caja. [Fuente: El Autor.]	36
2.7.	Placa en vista 2D. [Fuente: El Autor.]	37
2.8.	Esquema de comunicación de la placa de desarrollo HTCC-AB 01 hacia los sensores. [Fuente: El Autor.]	38
2.9.	Esquema de alimentación eléctrica. [Fuente: El Autor.]	39
2.10.	Topología de red. [Fuente: El Autor.]	41
2.11.	Creación de perfil de dispositivos. [Fuente: El Autor.]	41
2.12.	Método de conexión OTAA. [Fuente: El Autor.]	42
2.13.	Decodificación de datos. [Fuente: El Autor.]	42
2.14.	Creación de nueva aplicación llamada ServerCO2 [Fuente: El Autor.]	43
2.15.	Registro de dispositivos finales. [Fuente: El Autor.]	43
2.16.	Creación de clave API global. [Fuente: El Autor.]	44

2.17. Prototipo pantallas inicial del usuario. [Fuente: El Autor.]	46
2.18. Prototipo pantallas inicial del usuario. [Fuente: El Autor.]	47
2.19. Prototipo de pantalla de inicio de sesión. [Fuente: El Autor.]	48
2.20. Prototipo pantallas cambio de ubicación de los nodos. [Fuente: El Autor.] . . .	49
2.21. Prototipo pantallas para observar registros de nodos. [Fuente: El Autor.] . . .	50
2.22. Flujograma API REST del servidor de aplicaciones ChirpStack. [Fuente: El Autor.]	51
2.23. Flujograma de funcionamiento de los servicios Front-End. [Fuente: El Autor.] .	52
2.24. Flujograma de funcionamiento Frontend. [Fuente: El Autor.]	55
2.25. Diagrama de clases utilizadas. [Fuente: El Autor.]	58
2.26. Diagrama de clases utilizadas. [Fuente: El Autor.]	60
2.27. Diagrama de la base de datos PostgreSQL. [Fuente: El Autor.]	61
2.28. Icono de aplicación de monitoreo. [Fuente: El Autor.]	62
3.1. Dashboard Chirpstack	63
3.2. Características de Nodos. [Fuente: El Autor.]	64
3.3. Ubicaciones Iniciales de Nodos. [Fuente: El Autor.]	65
3.4. Ubicaciones Iniciales de Nodos. [Fuente: El Autor.]	65
3.5. Tramas de datos recibidas. [Fuente: El Autor.]	66
3.6. Estado de servicio para almacenar datos en archivo de texto. [Fuente: El Autor.]	67
3.7. Estado de servicio para almacenar datos en base de datos. [Fuente: El Autor.] .	67
3.8. Pantallas de Inicio de usuarios. [Fuente: El Autor.]	69
3.9. Pantallas de Valores de Sensores. [Fuente: El Autor.]	70
3.10. Prototipo de pantalla de inicio de sesión del administrador. [Fuente: El Autor.]	71
3.11. Edición de Ubicaciones de sensores de CO ₂ [Fuente: El Autor.] . .	72
3.12. Ingreso a registros de ChirpStack. [Fuente: El Autor.]	73
3.13. Comparación de porcentaje de paquetes perdidos de sensores. [Fuente: El Autor.]	74
3.14. Comparación de porcentaje de datos recibidos y perdidos por día de la semana. [Fuente: El Autor.]	75
3.15. Comparación de sensores.[Fuente: El Autor.]	77

Índice de tablas

1.1. Efectos del CO ₂	12
1.2. Microcontroladores / Tarjetas de Desarrollo	21
1.3. Características de sensores	24
2.1. Especificaciones de energía de SeeedStudio Lipo Rider Pro	29
2.2. Características sensor de gas MG811	30
2.3. Valores analógicos medidos	31
2.4. Características sensor de gas MHZ-19B	33
2.5. Configuración general del sensor de gas MHZ-19B	33
2.6. Lista de comandos	34
2.7. Lectura de concentración de gas CO ₂	34
2.8. Valor de retorno de concentración de gas CO ₂	34
3.1. Ubicaciones de Nodos y Gateway	64
3.2. Analisis de resultados de datos	78
3.3. Presupuesto de nodos con dos sensores	79
3.4. Presupuesto de nodos con sensor MHZ-19B	80
3.5. Presupuesto de nodos con sensor MG811	80
3.6. Resumen costos totales	81

Resumen

Este trabajo analizó la arquitectura de red de sensores inalámbricos usando tecnologías LPWAN que permita la medición y monitoreo remoto de CO₂ en el aire. Así, se define el uso de nodos LoRa y un servidor LoRaWan basado en ChirpStack para la publicación de los datos medidos. La red LoRaWan usada es del grupo de Investigación en Telecomunicaciones y Telemática (GITEL).

A diferencia de trabajos anteriores o similares, se desarrolló también una aplicación móvil de fácil uso que permita monitorear diferentes nodos sensores e identificar su ubicación georeferenciada permitiendo también la revisión histórica del estado de CO₂ en cada nodo. La aplicación móvil fue desarrollada usando el framework Flutter, que ofrece una interfaz intuitiva y amigable para los usuarios. La aplicación permite visualizar en tiempo real los datos recopilados por los sensores, así como acceder a estadísticas y registros históricos. También se implementó funciones de inicio de sesión y edición de ubicaciones, proporcionando una experiencia personalizada.

También, se buscó la posibilidad de identificar sensores de CO₂ que puedan ser adquiridos en corto tiempo y puedan ser comparados al ser usados en los nodos sensores de la red implementada. Para esto, el proyecto involucró el uso de los sensores el MH-Z19B y MG811. Se diseñó una placa PCB personalizada para usar los sensores evaluados y aplicar la adquisición de datos. Además, se implementó una caja de aplicación IoT para proteger los componentes electrónicos y facilitar el uso en exteriores. Así, como resultado de este proyecto se desarrollaron tres nodos que contienen dos sensores capaces de medir CO₂ con exactitud capaces de recolectar datos a distancias de hasta 4.5 Km.

Palabras clave: ChirpStack; CO₂ ; LoRaWan; IoT; Flutter.

Abstract

This work analyzed the architecture of a wireless sensor network using LPWAN technologies to enable the remote measurement and monitoring of CO₂ in the air. Thus, the use of LoRa nodes and a LoRaWan server based on ChirpStack for the publication of the measured data is defined. The LoRaWan network used is from the Telecommunications and Telematics Research Group (GITEL).

Unlike previous or similar works, an easy-to-use mobile application was also developed to monitor different sensor nodes and identify their geo-referenced location allowing also the historical review of the CO₂ status at each node. The mobile application was developed using the Flutter framework, which offers an intuitive and user-friendly interface. The application allows real-time visualization of the data collected by the seigneurs, as well as access to statistics and historical records. Login and location editing functions were also implemented, providing a personalized experience.

Also, the possibility was sought of identifying sensors that could be acquired in a short time and could be compared when used in the sensor nodes of the implemented network. For this, the project involved the use of the MH-Z19B and MG811 sensors. A customized PCB board was designed to use the evaluated sensors and apply data acquisition. In addition, an IoT application box was implemented to protect the electronic components and facilitate outdoor use. Thus, as a result of this project, three nodes were developed containing two sensors capable of accurately measuring CO₂ capable of collecting data at distances up to 4.5 km.

Keywords: ChirpStack; CO₂; IoT; Flutter; LoRaWAN.

Antecedentes

El aumento de concentración de dióxido de carbono (CO₂) en la atmósfera, es uno de los principales gases de efecto invernadero junto con otros contaminantes atmosféricos, ejerce un impacto en la calidad del aire de tal forma que ha generado preocupación a nivel mundial debido a las consecuencias del medio ambiente y en la salud humana [1]-[3].

La contaminación atmosférica afecta a todos por igual, como países desarrollados y países en vías de desarrollo, los países más perjudicados son los de ingresos bajos y medios, particularmente en las regiones del Pacífico Occidental y Asia sur oriental.

Las fuentes principales de contaminación ambiental son: el consumo de energía para cocinar y calefacción, los automóviles, la producción de energía, la agricultura, la incineración de residuos y la industria. Por lo que la calidad del aire está relacionada íntimamente con el clima de nuestro planeta y los ecosistemas del mundo, además muchos de estos agentes que generan contaminación atmosférica son también fuentes de emisión de gases de efecto invernadero [4].

Ecuador no está exento del problema de la contaminación del aire y en sus principales ciudades, donde la contaminación del aire es causada por la quema de combustibles fósiles, efectos invernaderos de la actividad agropecuaria, la generación eléctrica, la incineración de residuos y los parques industriales. Estos tipos de contaminación se caracteriza principalmente en las emisiones de toda clase de gases al exterior sin un tratamiento previo, afectando la salud de la población aledañas [5].

Mientras que, en la Amazonia Ecuatoriana, sufre graves consecuencias ambientales debido a la producción a gran escala causada por la explotación

petrolera. También, la actividad agropecuaria es otro de los factores más contaminantes del medio ambiente de esta zona [5], [6].

En Ecuador se han realizados varios estudios sobre la contaminación de la calidad del aire en las principales ciudades como: Esmeraldas, Guayaquil, Ambato, Cuenca y Quito en esta última ciudad tiene sistemas de monitoreo continuo de la calidad de aire [5], [7], [8].

En Quito, se han implementado varios proyectos de monitoreo ambiental, uno de ellos fue un sistema de monitoreo de bajo costo con redes de sensores inalámbricos visualizados en tiempo real mediante una aplicación web, conformado por varios nodos que envían los datos de CO₂, partículas de polvo (PM 2.5) y la temperatura en el ambiente, para luego ser mostrados en una página web [9].

Mientras que en [10]-[14], se hicieron varios estudios y diseños de monitoreo de la calidad de aire en la ciudad de Cuenca, en donde se implementaron varios modelamiento de dispersión espacial de contaminantes del aire. Estos modelos efectuaron simulaciones meteorológicas, fotoquímicas, monitoreo ambientales en un determinado número de horas y en diferentes lugares de la ciudad. El análisis expuso que la cobertura espacial de información meteorológica no es suficiente para alcanzar a datos concluyentes dado que contaba con información de una resolución muy limitada. Además, que sus sistemas de visualización y notificación dependían de la interacción de los usuarios o de sensores que no recolectan muestras de manera exacta.

También, en [13], [15], se fusiona la integración del concepto de IoT con la tendencia emergente de las ciudades inteligentes y la inteligencia ambiental, con el objetivo de tener un monitoreo, en tiempo real de la calidad del aire de la ciudad de Cuenca.

En estos trabajos se utilizó un nodo móvil inteligente, un servicio web Rest, una plataforma web y una base de datos no relacional (MongoDB), que monitorea en tiempo real un conjunto los niveles de CO₂ temperatura y humedad del ambiente para luego analizar los datos obtenidos.

En la ciudad de Guayaquil, se ha diseñado una red inalámbrica de monitoreo de CO₂ basado en la tecnología Sigfox [13], [16], [17], consta de un

nodo con un módulo sensor de CO₂ , una placa de Arduino y un módulo de transmisión Sigfox. El nodo envía los valores de niveles de CO₂ a las estaciones bases y este los reenvían al servidor de red de la plataforma Sigfox, para llegar a un servidor web.

Basándose en lo previamente examinado, es evidente que aún quedan aspectos por investigar y optimizar en diversas áreas como el procesamiento, el almacenamiento y la representación de los datos que están siendo monitorizados.

La plataforma propuesta tiene el propósito de brindar notificaciones en tiempo real cuando se alcancen niveles elevados de contaminación perjudicial para la salud humana en zonas consideradas como tal. Además, se establecerá una base de datos que permitirá llevar a cabo un análisis exhaustivo del historial de contaminación en la ciudad.

Justificación

De acuerdo a la revisión del estado del arte del monitoreo de calidad, se ha constatado una notable carencia de la ausencia de plataformas inteligentes que incorporen redes de sensores inalámbricos (WSN) basados en Internet de las cosas (IoT) con aplicaciones móviles, con el propósito de monitorizar y alertar a la población en general en relación a los niveles de dióxido de carbono (CO₂) en el ambiente.

En el cual se concisa la ausencia de investigaciones que traten sobre la combinación de redes de sensores IoT en aplicaciones móviles para el monitoreo de la calidad del aire, remarcando la importancia de la propuesta como una solución innovadora y de gran utilidad para plantear una solución al problema de contaminación ambiental y sus efectos adversos para el planeta y en la salud humana.

Por esta razón, la propuesta no solo se enfoca en generar conciencia sobre los niveles de dióxido de carbono (CO₂) en el aire, sino que también busca proveer a los usuarios finales una herramienta con la capacidad de monitorizar los niveles de CO₂ en su entorno más cercano y que permita tomar decisiones y medidas preventivas, para ayudar a mitigar la contaminación del aire. Mediante el uso de una plataforma de monitoreo de niveles de CO₂ que permita la adquisición, almacenamiento y visualización de valores de CO₂ en tiempo real a través de una aplicación móvil.

Esta propuesta radica en su capacidad para generar un efecto beneficioso en la salud, el medio ambiente y la conciencia ciudadana. Al emprender de manera innovadora la problemática de la contaminación del aire, esta investigación contribuye de manera significativa al avance en la comprensión y la disminución de un desafío crítico que afecta a la población en todo el

mundo, integrando tecnologías en desarrollo, como el IoT, con una plataforma de monitoreo y una aplicación móvil, adaptada a las necesidades y expectativas actuales de una sociedad cada vez más digitalizada.

En la figura 1, se observa el resultado de la revisión bibliográfica, donde se ha identificado que no se han encontrado trabajos relacionados con redes de sensores basados en IoT y presentados en una aplicación móvil. Por esto, se propone dar una solución que ayude a la concientización para disminuir los niveles de CO₂ en el aire mediante el uso de una plataforma de monitoreo que permita visualizar y alertar sobre los niveles de CO₂ en el aire. Este trabajo plantea complementar esos puntos, creando una plataforma con un sistema de monitoreo basado en IoT, capaz de recolectar y almacenar datos para poder visualizarlos en una aplicación móvil.

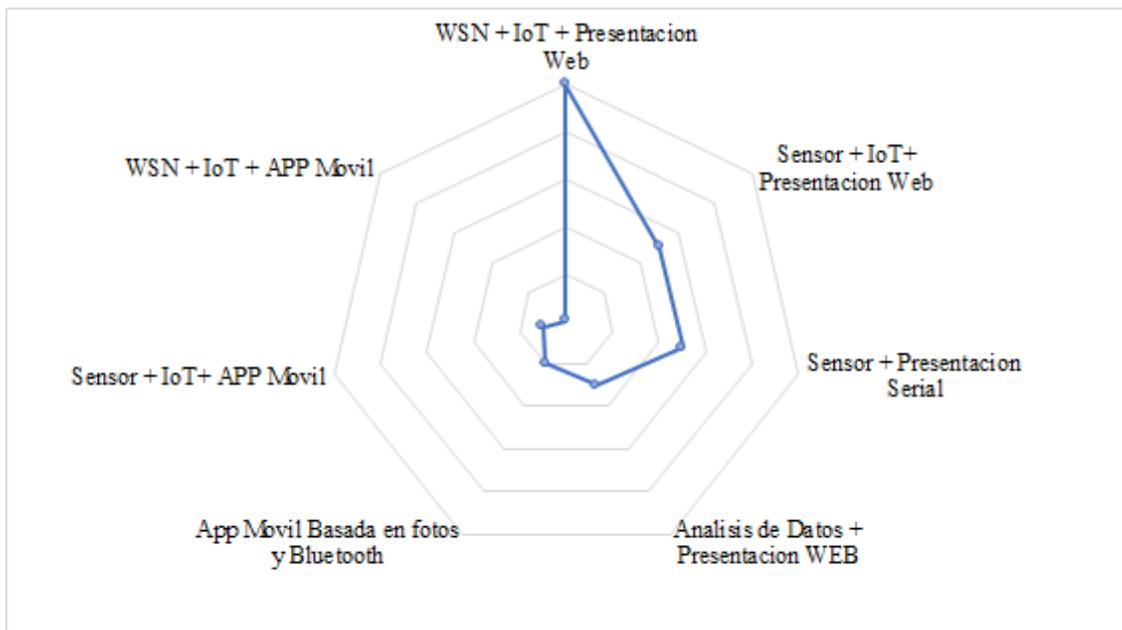


Figura 1: Trabajos Previos [Fuente: El Autor.]

Objetivos

Objetivo General

- Diseñar una plataforma IoT para el monitoreo de niveles de CO₂ .

Objetivos específicos:

1. Estudiar la información científica que soporte y genere un mapa de ruta acerca de sensores y métodos de adquisición para el monitoreo de CO₂ placas de desarrollo para trabajo con redes de sensores inalámbricas y la implementación de plataformas de monitoreo de calidad de aire.
2. Diseñar un sistema de adquisición de datos para la medición de niveles de CO₂ basado en IoT de acuerdo con los métodos y materiales identificados.
3. Diseñar una aplicación móvil para monitorizar los niveles de CO₂ adquiridos.
4. Evaluar el desempeño del sistema diseñado para el adecuado monitoreo de la calidad del aire.

Introducción

Actualmente, la calidad del aire es un factor esencial para el bienestar y la salud de las personas, es necesario apoyarse con herramientas eficientes para medir y analizar los niveles de contaminantes del aire. En respuesta a esta problemática, se presenta la “Plataforma de monitoreo de niveles de CO₂ basada en IoT”, una plataforma diseñada para ofrecer a los usuarios la visualización de niveles de CO₂, para analizar y tomar decisiones informadas sobre la calidad del aire y el impacto en su entorno.

La Plataforma de Monitoreo de niveles de (CO₂) desarrollada combina las tecnologías de Redes de sensores inalámbricos (WSN) con Internet de las cosas (IoT), con una interfaz de usuario amigable de fácil acceso a través del uso de un dispositivo móvil. Así, la aplicación móvil permite a los usuarios acceder a valores de CO₂ en ppm en tiempo real, visualizar gráficos detallados y obtener estadísticas clave sobre los niveles monitorizados. En cualquier ambiente, ya sea en ambientes exteriores o interiores como el hogar, la oficina, la plataforma ofrece una visión integral de la calidad del aire.

En este sentido, los valores medidos por los sensores de CO₂, reflejan la concentración de este gas en diferentes situaciones ambientales. Con la finalidad de comprender el conjunto de valores medidos, se han aplicado varias técnicas para la visualización que permiten extraer conocimientos notables y presentarlos de manera clara y concisa.

En este análisis, se demostró el funcionamiento de diversos sensores de CO₂, además, permite identificar tendencias, patrones clave en los valores es importante para tomar medidas de prevención y efectivas en la administración de la calidad del aire.

Capítulo 1

Revisión bibliográfica

En este capítulo, se amplía la revisión bibliográfica expuesta en los antecedentes de la implementación del Internet de las cosas (IoT, del inglés *Internet of Things*) en el contexto del monitoreo de la calidad del aire, donde el objetivo de esta revisión es identificar las aplicaciones y el funcionamiento de los sensores que han sido utilizados en el monitoreo de la calidad del aire, enfocándonos en la medición de dióxido de carbono (CO₂). Además, se buscarán los métodos más idóneos para la adquisición y el procesamiento de los datos generados por estos sensores.

1.1. Estado del arte del monitoreo de CO₂ basado en IoT

Se obtuvieron resultados de un análisis bibliométrico respaldado en la base de datos SCOPUS. Mediante la aplicación de criterios de búsqueda específicos, a partir de la metodología de un diagrama de flujo prisma, utilizando las palabras clave “monitoreo”, “dióxido de carbono”, “CO₂” y “IoT”, se seleccionó un total de 79 documentos que cumplían con los criterios de estudio establecidos [1].

Se realizó un análisis de la producción de países que más han contribuido y las contribuciones específicas de los investigadores de América Latina en la medición del CO₂ utilizando la tecnología del paradigma de IoT.

La expansión de estos resultados se puede revisar en el artículo [1], donde

se realiza un análisis a profundidad del estudio bibliométrico basado en IoT en lo que respecta al monitoreo de la calidad del aire.

1.1.1. Países que más han contribuido

Como se observa en la figura 1.1 que hasta el año 2022, los países que han liderado en aportes con relación al monitoreo de CO₂ basado en IoT son India e Indonesia, con 20 y 13 publicaciones respectivamente.

En contraste, América ha presentado únicamente 4 publicaciones sobre este tema. En consecuencia, los continentes de Asia y Europa sobresalen como los principales contribuyentes en el campo de interés.

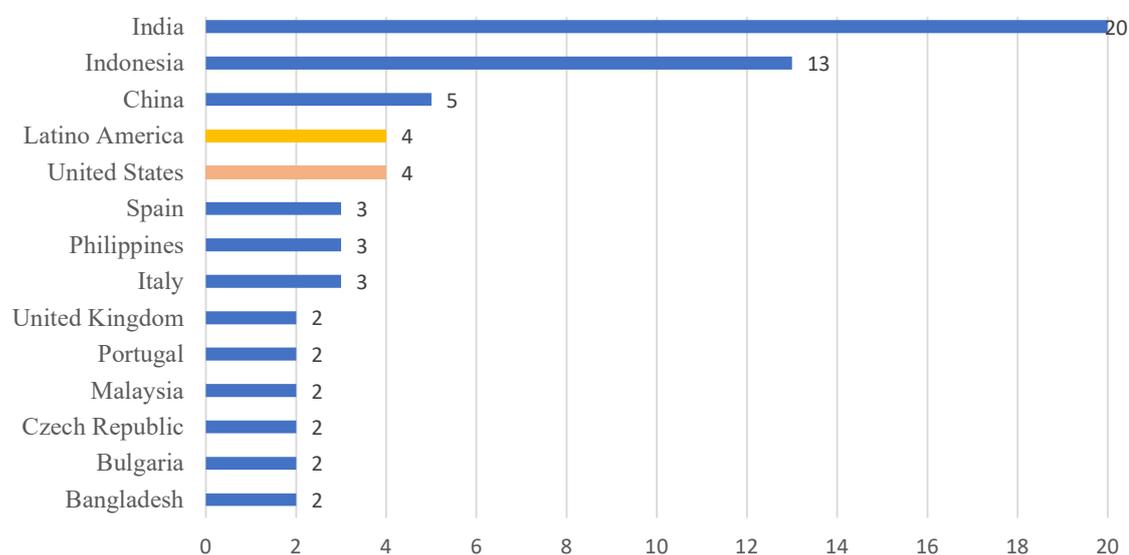


Figura 1.1: Países que más han contribuido [Fuente: El Autor.]

1.1.2. Países de América Latina que han contribuido

Es importante señalar que los países latinoamericanos que han contribuido en la medición y análisis de dióxido de carbono (CO₂) mediante la tecnología IoT incluyen a Brasil, Ecuador, México y Perú, como se observa en la figura 1.2.

El aporte de Latinoamérica en esta área refleja la generación de conocimientos nuevos, identificación de problemáticas locales y globales y colaboración con investigadores de otras partes del mundo.

Estos aportes son importantes para avanzar en la comprensión y abordaje de la contaminación del aire, y para trabajar en la mejora de un ambiente más saludable y sostenible tanto a nivel local como global [1].



Figura 1.2: Países de América Latina que han contribuido [Fuente: El Autor.]

1.2. Fundamentos Teóricos

En esta sección, se realiza una descripción de los conceptos teóricos en los que se basa la plataforma de monitoreo de niveles de CO₂.

1.2.1. Dióxido de carbono

El CO₂ es un gas inodoro e incoloro compuesto por un átomo de carbono y dos átomos de oxígeno, que se intercambia en la biosfera global actuando como termostato al impedir la fuga de la radiación solar que existe en la tierra [18]. No obstante, el crecimiento de CO₂ en la atmósfera conduce a la desestabilización de los sistemas que se relaciona con el calentamiento

global ya que se desprende de la respiración, en las combustiones y algunas fermentaciones [18], [19].

El CO₂ se encuentra naturalmente en la atmósfera del planeta como un gas, tiene una fracción molar de 400 ppm. Además que la concentración actual es de 0,04 % en volumen aun mayor a los niveles preindustriales de 280 ppm [19].

1.2.2. Efectos de CO₂ en el ser humano

A nivel mundial la contaminación de la calidad del aire es actualmente uno de los problemas ambientales más significativos que representa un fenómeno que impacta en la salud del ser humano [20]. Según la Organización Mundial de la Salud (OMS), cada año se producen al menos siete millones de defunciones atribuibles a la contaminación atmosférica, los datos de la OMS indican que al menos el 90 % de la población respira aire con niveles de contaminación que exceden las recomendaciones de sus directrices [21]. Además. Se estima que al menos 4,2 millones de personas fallecen a causa de accidentes cerebro vasculares, cardiopatía coronaria, cáncer de pulmón y enfermedades respiratorias agudas y crónicas, todo ello relacionado con la contaminación del aire [21], [22].

1.2.3. Concentración de CO₂ en ambientes abiertos y cerrados

El dióxido de carbono en ambientes cerrados tiene diferentes implicaciones dependiendo de la concentración puede causar dolor de cabeza, mareos, somnolencia y problemas respiratorios.

En ambientes abiertos se encuentra niveles entre 0 y 400 ppm, alcanzando valores hasta de 550 ppm y como consecuencia se obtiene el calentamiento global [23].

En la tabla 1.1, se observan los valores de CO₂ y los efectos que pueden causar a los seres humanos.

Tabla 1.1: Efectos del CO₂

Nivel de CO ₂ (ppm)	Efectos en Interiores	Efectos en Exteriores
0 - 400	Normal en exteriores e interiores bien ventilados.	Normal en exteriores e interiores bien ventilados.
400 - 1000	Típico en interiores bien ventilados, no se espera impacto en la salud.	Típico en interiores bien ventilados, no se espera impacto en la salud.
1000 - 2000	Puede causar ligera incomodidad en personas sensibles.	Puede causar ligera incomodidad en personas sensibles.
2000 - 5000	Mala calidad del aire interior, posibles síntomas como somnolencia y falta de concentración, especialmente en personas sensibles.	Mala calidad del aire exterior, posible afectación en la salud de personas sensibles.
5000 - 10000	Puede causar problemas de salud más notables en interiores, especialmente en personas con afecciones respiratorias.	Puede afectar la salud de manera notoria en exteriores, especialmente en personas con afecciones respiratorias.
10000+	Niveles preocupantes en interiores, síntomas graves en la mayoría de las personas, incluyendo dificultad para respirar y fatiga.	Niveles preocupantes en exteriores, síntomas graves en la mayoría de las personas, dificultad para respirar, fatiga.

1.2.4. Tiempo de muestro para medir una variable de la calidad del aire

A consecuencia de la contaminación que se presenta en las grandes ciudades, es necesario llevar a cabo acciones para mejorar la calidad del aire, siendo el primer paso medir cuál es la calidad del mismo y poder determinar sus causas y evaluar sus efectos [24].

En áreas urbanas, la evaluación de la calidad del aire se debe realizar las mediciones con mayor frecuencia debido a que la calidad varía significativamente durante el día y la noche.

La calidad del aire se evalúa en la zona urbana donde deben realizarse mediciones con mayor frecuencia dado que la calidad varía significativamente durante el día y la noche [25].

1.2.5. Internet de las cosas (IoT)

Hoy en día, diversas tecnologías de proveedores de servicios para el Internet de las cosas están ganando popularidad en todo el mundo para simplificar los objetos que las personas utilizan día a día. Se estima que en 2020, más de 25 millones de dispositivos están conectados de forma inalámbrica. Debido a la rápida expansión del Internet de las cosas (IoT), las tecnologías relacionadas con las Redes Inalámbricas de Sensores (WSN) se han vuelto muy populares [26], [27]. Además, entre las necesidades que surgen para resolver la comunicación de mediciones de variables de diferente índole, surgen las redes

de largo alcance con baja potencia de consumo (LPWAN, del inglés *Low Power Wide Area Network*), permitiendo a través de una topología física en estrella, cubrir áreas extensas de monitoreo. Dentro de este tipo de redes se encuentran protocolos como, sigfox [28] y LoRa [29].

Es importante tomar en cuenta que, para el paradigma de IoT, debido a que los datos de medición corresponden a paquetes de datos de corta extensión, las tecnologías que permiten el acceso, no usan tasas de transmisión elevadas. En este sentido, es importante tomar en cuenta que, para conseguir transmisiones a larga distancia con bajo consumo de energía, se sacrifica velocidad de transmisión. Así, para el caso de sigfox se tiene una tasa máxima de transmisión de 600bps y para LoRa 5470bps [30].

El paradigma de IoT, se trata de un sistema global de redes informáticas interconectadas que emplean el conjunto de protocolos de Internet (TCP/IP) con el propósito de atender a miles de millones de usuarios en todo el mundo, permitiendo la interconexión de dispositivos cotidianos, instrumentos, vehículos, edificios y otros elementos mediante la integración de electrónica, circuitos, software y sensores, que se conectan a través de una amplia variedad de tecnologías de redes inalámbricas, ópticas y electrónicas, lo que permite a estos dispositivos recopilar y compartir datos [31]. También IoT permite que los dispositivos puedan ser controlados de forma remota a través de la infraestructura de red existente [32].

El número de dispositivos que utilizan internet va en aumento cada día y teniendo todos conectados por cable o inalámbrica es un fuente poderosa de información, por lo que IoT es capaz de interactuar sin la necesidad de la intervención humana, ya que se han desarrollado varias aplicaciones en las industrias como la de salud, transporte y de uso cotidiano. Aun las tecnologías IoT están en etapas iniciales, pero se han desarrollado la incorporación de objetos equipados con sensores en Internet, conocida como IoT, implica diversas consideraciones, incluyendo aspectos de infraestructura, comunicaciones, interfaces, protocolos y estándares. [32], [33].

Aplicaciones de IoT

IoT permite la interacción y comunicación entre los objetos físicos, lo que permite observar su entorno, procesar información, trabajar de manera conjunta y tomar decisiones coordinadas. Esto implica que los objetos físicos pueden "hablar" entre sí, intercambiar datos y colaborar en la toma de decisiones [34].

Las aplicaciones IoT más comunes en las que se utilizan son:

Hogar inteligente: Es un sistema compuesto por varios componentes, incluyendo un controlador central, con una variedad de sensores y actuadores, así como aplicaciones móviles que permiten a los usuarios automatizar las tareas domésticas. Desde esta perspectiva, hay diversas opciones electrónicas disponibles para lograr el objetivo [35].

Aplicación en agricultura: La implementación de una red IoT puede tener un impacto significativo en el progreso de los agricultores, ya que les permite aprovechar de manera eficiente el uso de sensores en el suelo. Estos sensores les brindan la capacidad de planificar el momento adecuado para la siembra, supervisar el crecimiento de las plantas y establecer un horario de riego óptimo y adecuado [36].

IoT en desastres naturales: El IoT desempeña un papel crucial al facilitar la supervisión y la predicción de eventos significativos, como los terremotos. En situaciones de desastres naturales, la implementación de sensores y la rápida comunicación a través de una red de nodos compatibles con IoT permiten la difusión rápida de mensajes desde los nodos ubicados en áreas afectadas [36].

1.2.6. Red de sensores Inalámbricos (WSN)

Una red de sensores inalámbricos (WSN, del inglés *Wireless Sensor Networks*) es una red con dispositivos diminutivos equipado con sensores que permite la posibilidad de crear una red que captura, registra y almacena datos de distintas condiciones, como por ejemplo datos de temperatura, sonido, vibraciones, presión o variables contaminación ambiental [37], [38].

El estándar IEEE 802.15.4 establece regulaciones tecnológicas para la información, las telecomunicaciones y el intercambio de datos. Estas normas

son fundamentales para el funcionamiento y la implementación de las WSN, ya que estas redes se basan en la comunicación mediante señales de radio y utilizan un espectro específico de señales electromagnéticas [39].

Topologías de Redes de sensores Inalámbricos (WSN)

Las redes de sensores inalámbricos por sus siglas en inglés son agrupadas en diferentes maneras que buscan optimizar la colocación de sus nodos para facilitar el envío y recepción de datos, a esta disposición se le conoce como topología [39]. Para entender las diferencias de las topologías se conoce los componentes de WSN.

- Nodos finales: Capturan los datos de sensores [38].
- Routers: Garantiza la cobertura de redes muy extensas, resolviendo problemas de congestión en la emisión de información y posibles fallos [38].
- Puertas de enlace: Reúne los datos de la red que sirven como punto de unión con la red LAN o con internet [38]. .

Topología Estrella: Todos los nodos de sensores están conectados directamente a un nodo central, la comunicación se realiza por medio del mismo usualmente a una distancia de 30 a 100 metros [38].

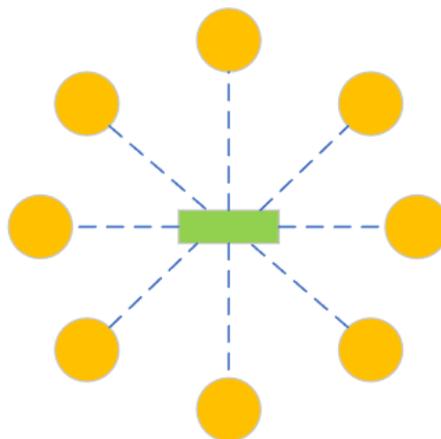


Figura 1.3: Topología Estrella. [Fuente: El Autor.]

Topología Malla: Los nodos de sensores están conectados directamente a varios nodos estructurando una malla. Cada nodo puede enviar y recibir información de otro nodo y de la puerta de enlace [38].

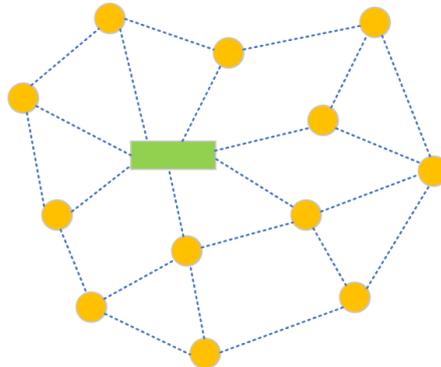


Figura 1.4: Topología Malla. [Fuente: El Autor.]

Topología Híbrida Estrella-Malla: Es una red estrella alrededor de routers pertenecientes a una red de malla [38].

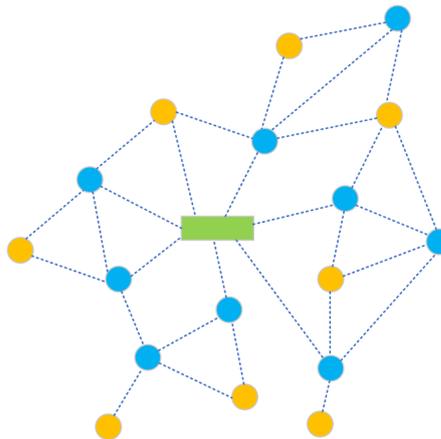


Figura 1.5: Topología Híbrida Estrella-Malla. [Fuente: El Autor.]

1.2.7. Longe Range (LoRa)

LoRa es una tecnología que tiene una conectividad IoT inalámbrica que está ganando campo en sistemas integrados que permite soportar interferencias de ruido, así mismo LoRa se refiere a la capa física de la modulación inalámbrica que permite lograr un enlace de largo alcance [40].

Principales características:

- Ancho de banda: Es capaz de adaptarse a los parámetros de ancho de banda, lo que significa que transmite datos a velocidades bajas [40].

- Capacidad de red: utiliza factores de propagación que permite realizar la difusión de múltiples señales al mismo tiempo y sobre un mismo canal como por ejemplo las redes de área extensa (LoRaWAN) que conecta múltiples dispositivos [40].
- Seguridad: se utiliza para autenticar el nodo de la res, la seguridad de la aplicación protege los datos de la aplicación del usuario final [41].

Arquitectura La red LoRa consta de cuatro elementos básicos:

- El nodo LoRa o puntos finales: Es el corazón del sistema embebido que comprende sensores o aplicaciones que lleva a cabo la detección y el control. Como por ejemplo monitoreo de gas y aplicaciones similares [41].
- Salida:Cada nodo final está conectado a todas las puertas de enlace. Los datos transmitidos por un nodo son enviados a todas las puertas de enlace disponibles. Cada puerta de enlace que recibe una señal la retransmite a un servidor de red ubicado en la nube, utilizando tecnologías como celular, Ethernet, satélite o Wi-Fi [41].
- Servidor de red: Contiene toda la inteligencia.
- Servidor de aplicaciones: Los datos previstos son recibidos del servidor de red [41].

1.2.8. LoRaWAN

Garantiza velocidades de datos que varían desde 0.3 kbps hasta 50 kbps, las cuales son consideradas adecuadas para la transmisión de datos de sensores en tiempo real [42].

Clases LoRaWAN

En la misma red, se pueden identificar tres clases distintas y los dispositivos finales tienen la capacidad de modificar su configuración entre dichas clases [43].

Clase A:

Cada nodo individual toma la decisión de cuándo realizar una transmisión. Durante cada ventana de transmisión, todos los nodos hacen el intento, pero solo uno de ellos logrará tener éxito en su transmisión [44]. Esta clase se considera la más eficiente y es especialmente apropiada para aplicaciones en las que la transmisión de datos es prácticamente la única necesidad [43].

Clase B:

Mediante esta clase, es posible generar ventanas de recepción sin requerir una transmisión previa. Esta funcionalidad permite aumentar la capacidad del dispositivo final para recibir datos de manera más efectiva [43].

Clase C:

Estos dispositivos funcionan en un modo de recepción continua que solo se interrumpe cuando se realiza una transmisión. Esta categoría de dispositivos proporciona la menor latencia de conexión entre los dispositivos finales y las pasarelas, aunque a costa de un mayor consumo de energía [43].

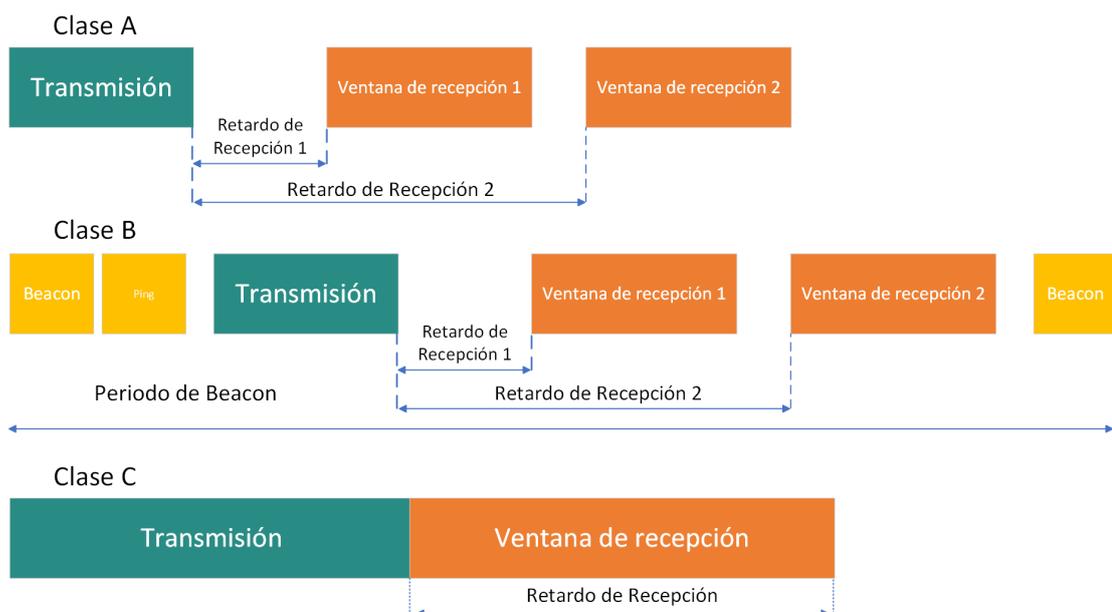


Figura 1.6: Clases de LoRaWAN. [Fuente: El Autor.]

1.2.9. Chirpstack

Se trata de un servidor de código abierto que posibilita la implementación de un número restringido de gateways y dispositivos IoT que forman parte de la red LoRaWAN [45].

La puesta en marcha y ajuste de los servidores Chirpstack involucra la instalación de las dependencias y paquetes de software de Chirpstack, así como la configuración de la organización, perfil de servicio, aplicación, perfil de dispositivo y, por último, la configuración de los dispositivos correspondientes [46].

Características:

- Activación por personalización (ABP): El servidor de red ChirpStack tiene soporte para pre-activar los dispositivos a través de su API [47].
- Activación Sobre el Aire (OTAA): En caso de una activación al servidor de red ChirpStack se le denomina Unirse al servidor referido a la configuración que se encarga de la autenticación del dispositivo y la generación de claves [47].
- Encriptación: Este servidor se encarga de manejar el cifrado y descifrado de los datos de la aplicación transmitidos. Además, es responsable de almacenar y gestionar las claves de aplicación de cada dispositivo, y también administra el proceso de aceptación de la unión cuando se activa el método de autenticación OTAA [47].
- Interfaz web: Esta interfaz web tiene la capacidad de ser utilizada para administrar usuarios, organizaciones, aplicaciones y dispositivos de manera eficiente [47].

1.2.10. Frameworks para desarrollo de aplicaciones móviles

En el desarrollo de aplicaciones móviles , un framework es una composición que tiene un soporte bien definido, tiene módulos de software que pueden ser fácilmente organizado, desarrollado y hace más sencilla la programación de cualquier aplicación [48]. Al utilizar un framework, el

desarrollador se basa en una estructura predefinida que debe seguir. Esta estructura incluye, entre otros elementos, archivos de configuración básica que permanecen constantes, así como componentes previamente desarrollados por los creadores del framework con el propósito de permitir su reutilización [49].

Frameworks utilizados en el desarrollo de aplicaciones móviles:

Flutter: El framework fue creado por Google y está diseñado para el desarrollo de aplicaciones móviles de multiplataforma. Al ser de código abierto, posibilita la construcción de aplicaciones que son compatibles tanto con Android como con iOS [50].

React Native: Representa una herramienta tecnológica de código abierto, cuya comunidad es reconocida por su notable crecimiento en el ámbito del software libre. Se fundamenta en JavaScript y se emplea para desarrollar aplicaciones móviles destinadas a las plataformas iOS y Android [50].

NativeScript: Es una aplicación móvil de código abierto. Mediante el uso de esta herramienta, los desarrolladores pueden acceder a plataformas nativas (Android o iOS), especialmente a través de JavaScript, TypeScript y Angular. Además, esta estructura está diseñada para fomentar la reutilización de código no solo entre plataformas móviles, sino también a nivel web [51].

Xamarin: es un framework de código abierto conocido como Proyecto Mono que fue adquirido por Microsoft en 2016. Permite el desarrollo de aplicaciones móviles mediante el uso de lenguaje diseñado por Microsoft C# [51].

1.2.11. Placas de desarrollo

En la tabla 1.2, se examinan las especificaciones técnicas de las placas de desarrollo con el propósito de seleccionar la opción más adecuada para nuestra plataforma de monitoreo.

Tabla 1.2: Microcontroladores / Tarjetas de Desarrollo

Tarjeta de Desarrollo	Lenguaje de Programación	Microcontrolador	Memoria	LoRa
PYCOM FIPY	MicroPython	Core: Xtensa dual-core 32-bit LX6 microprocessor(s)	RAM: 520KB + 4MB External flash: 8MB	-LoRaWAN stack - Class A and C devices Node range: Up to 40km - Nano-gateway: Up to 22km (Capacity up to 100 nodes)
MODULE RFM9X	MicroPython	LoRa Modem	FLASH: 256 Bytes	168 dB maximum link budget High sensitivity: down to -148 dBm FSK, GFSK, MSK, GMSK, LoRa-TM and OOK modulation
XIAO RP2040	Arduino MicroPython CircuitPython	Core: Dual Cortex M0+ processor cores, up to 133 MHz	Flash: 8MB - SRAM: 264KB	
HTCC-AB 01	Arduino MicroPython CircuitPython	Core: ASR6501 (48 MHz ARM Cortex M0+ MCU)	- Internal SRAM: 16KB Internal Flash: 128KB	LoRaWAN 1.0.2 support
CICERONE	Arduino	Core: STM32 Cortex-M4 32 bit @ 48MHz - frequency up to 168 MHz	SRAM: 128KB Flash: 48KB	LoRaWAN 1.0.2 support

En resumen, se han enumerado las ventajas y desventajas de cada una de las placas de desarrollo mencionadas

PYCOM FIPY:

Ventajas:

- Ofrece soporte para el lenguaje de programación MicroPython, lo que facilita el desarrollo rápido de aplicaciones.
- Cuenta con un microcontrolador que posee un procesador dual Xtensa de 32 bits LX6, brindando un rendimiento adecuado para aplicaciones IoT.
- Dispone de amplia memoria RAM y Flash externa, permitiendo manejar aplicaciones más complejas.
- Ofrece excelente soporte para LoRa, siendo adecuado para proyectos de comunicación a larga distancia.

Desventajas:

- Puede resultar más costoso en comparación con otras opciones de placas de desarrollo.

- Su tamaño y consumo de energía podrían no ser ideales para proyectos con limitaciones de espacio o energía.

MODULO RFM9X

Ventajas:

- Diseñado específicamente como un módem LoRa, lo que lo hace altamente especializado para aplicaciones de comunicación de larga distancia.
- Posee una excelente sensibilidad de recepción y un enlace de alta calidad, mejorando el rendimiento en condiciones de señal débil.
- Ideal para proyectos alimentados por baterías debido a su bajo consumo de energía

Desventajas:

- Limitado en términos de potencia de procesamiento y capacidad de memoria en comparación con placas de desarrollo más completas.

XIAO RP2040

Ventajas:

- Admite varios lenguajes de programación, como Arduino, MicroPython y CircuitPython, lo que lo hace versátil y adecuado para diferentes tipos de desarrolladores.
- Ofrece un buen rendimiento con su núcleo de procesadores Dual Cortex M0+ y memoria Flash y SRAM adecuadas para proyectos pequeños y medianos.

Desventajas:

- La información sobre su soporte y características de LoRa está faltando en la tabla, lo que dificulta evaluar completamente sus ventajas y desventajas.

HTCC-AB 01

Ventajas:

- Permite a los desarrolladores elegir entre varios lenguajes de programación, lo que les brinda opciones cómodas.
- Especializado en LoRaWAN 1.0.2, siendo adecuado para proyectos basados en esta tecnología.
- Apto para aplicaciones de bajo consumo debido a su eficiencia energética.

Desventajas:

- Su capacidad de procesamiento y memoria puede ser limitada en comparación con placas de desarrollo más potentes.

CICERONE

Ventajas:

- Ofrece soporte para el lenguaje de programación Arduino, ampliamente utilizado y conocido.
- Proporciona un rendimiento decente para aplicaciones embebidas gracias a su procesador STM32 Cortex-M4.

Desventajas:

- La memoria Flash y SRAM son más limitadas en comparación con algunas otras placas de desarrollo.

Luego de analizar las ventajas y desventajas de cada placa, se optó por seleccionar la placa HTCC-AB 01 debido a su soporte para varios lenguajes de programación y su especialización en LoRaWAN. Esta elección nos permitirá implementar y desplegar nuestra plataforma de monitoreo de manera efectiva y eficiente. Además, su eficiencia energética nos permitirá mantener el consumo de energía bajo control, lo cual es especialmente importante para asegurar un funcionamiento óptimo y prolongado de la plataforma en aplicaciones de monitoreo continuo.

1.2.12. Sensores utilizados en monitoreo de la calidad del aire

En la tabla 1.3, se presentan las características y costos de los sensores que se evaluarán para su implementación en la plataforma de monitoreo de CO₂. Se describen aspectos como los tipos de gases que procesan, el tiempo de precalentamiento, el rango de medición, el consumo de corriente en mA (que afectará la duración de las baterías del nodo), la interfaz de comunicación con la placa de desarrollo, y además, se incluye información sobre el costo de cada sensor.

Tabla 1.3: Características de sensores

Nombre	Tipo de Gas	Tiempo de Precalentamiento	Rango de medición (ppm)	Consumo medio	Interface	Costo
MH-Z19B	CO ₂	3 min	0 a 2000/5000	60mA	UART PWM	\$35
MICS 6814	- CO - NO2 - H2 - NH3 - CH4		Depende del gas	32 mA	Analógico	\$30.99
MG811	CO ₂	Sobre 48 horas	350 a 10.000	140 ± 20 mA	Analógico	\$70
CCS811	CO	1 min	400 a 29206	30mA	I2C	\$27

Una vez analizadas las características y costos de cada sensor de CO₂, hemos identificado las siguientes ventajas y desventajas asociadas a su uso:

MH-Z19B

Ventajas:

- Específicamente diseñado para medir CO₂, lo que proporciona mediciones precisas y fiables.
- Fácil interpretación de la salida de datos.
- Estabilidad y precisión en las mediciones.

Desventajas:

- Puede tener un costo más elevado en comparación con otros sensores.

- Requiere un tiempo de precalentamiento antes de obtener mediciones precisas.

MICS 6814

Ventajas:

- Es un sensor multifunción que también puede medir otros gases, lo que brinda versatilidad en aplicaciones que requieren la detección de varios gases.
- Costo más económico en comparación con algunos sensores especializados.

Desventajas:

- La medición específica de CO₂ podría no ser tan precisa como en sensores diseñados exclusivamente para este gas.
- Puede requerir calibración adicional para mediciones confiables de CO₂.

MG811

Ventajas:

- Adecuado para aplicaciones de bajo costo y proyectos de aficionados.
- Proporciona mediciones analógicas simples de CO₂.

Desventajas:

- Menos preciso en comparación con sensores más avanzados y especializados.
- Requiere calibración y mantenimiento periódico para obtener mediciones precisas.

CCS811

Ventajas:

- Ofrece mediciones precisas y confiables de la concentración de CO₂.

- Interfaz digital con salida I2C, lo que facilita la integración con microcontroladores.
- Capacidad para compensar la temperatura y la humedad, mejorando la precisión de las mediciones.

Desventajas:

- Puede tener un costo más elevado en comparación con algunos sensores más sencillos.
- Puede requerir calibración inicial para obtener mediciones precisas.

Cabe señalar que la elección del sensor CO₂ dependerá de los requisitos específicos de la plataforma de monitoreo, el presupuesto y las necesidades del proyecto. De acuerdo a las ventajas y desventajas de cada sensor, se decide elegir los sensores MH-Z19B y MG811 para la plataforma. El objetivo es comparar las mediciones de CO₂ y analizarlas en diferentes entornos o regiones.

A la hora de elegir los sensores MH-Z19B y MG811 para la plataforma de monitorización de CO₂, hay que tener en cuenta que el MG811, aunque más caro, también presenta ciertas ventajas, como mediciones analógicas simples de CO₂ y la posibilidad de su implementación en varias aplicaciones Bienes requeridos. Sin embargo, esto significa mayores gastos financieros.

Es importante equilibrar el costo y las capacidades específicas de cada sensor con la precisión requerida para la plataforma de monitoreo. La combinación de los sensores MH-Z19B y MG811 permitirá el análisis comparativo en diferentes ambientes o zonas, proporcionando una visión general más completa de los niveles de CO₂ y permitiendo tomar decisiones para mejorar la calidad del aire en cada zona monitorizada.

Capítulo 2

Desarrollo de la Plataforma de Monitoreo de Niveles de CO₂

Este capítulo presenta una descripción de los sensores y módulos que se usan para la plataforma de monitoreo, el diseño de la placa PCB, caja de aplicación IoT, la comunicación LoRaWAN entre los sistemas, además del diseño de la aplicación móvil.

2.1. Diseño de nodos de monitoreo

En esta sección se presenta un análisis corto de los dispositivos usados en cada nodo.

2.1.1. Placa de desarrollo, módulos y sensores

Después de haber analizado varias placas de desarrollo en el anterior capítulo, se llegó a la conclusión que para la plataforma de monitoreo la placa Heltec Cubecell HTCC-AB 01 es adapta más al objetivo del sistema por las siguientes consideraciones:

Placa de desarrollo Heltec Cubecell HTCC-AB 01

- La placa de desarrollo HTCC-AB 01, está diseñada por la compañía Heltec, incorpora el chip ASR650 x que es compatible con Arduino como se

muestra en la figura 2.1 [52].

- Puede ejecutar el protocolo LoRaWAN de manera estable [53].
- Sistema de administración de energía completo, se puede conectar directamente con un panel solar y baterías de Litio [53].

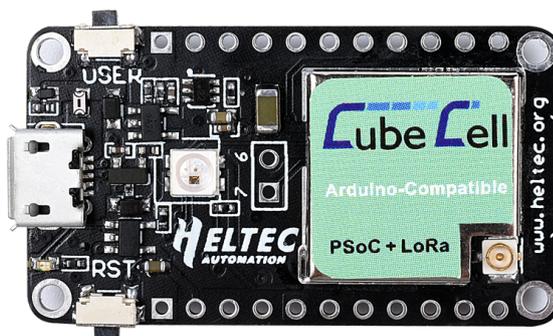


Figura 2.1: Cubecell HTCC-AB 01

SeedStudio Lipo Rider Pro

Lipo Rider Pro de SeedStudio, es un cargador y aumentador de potencia USB, que permite conectar un panel solar y una batería de litio como se muestra en la figura 2.2. Es una solución de energía renovable para el diseño de los nodos de sensores distribuidos en el exterior [54]. Entre algunas de las características de este modulo tenemos:

- Salida de carga máxima de 1A.
- Fuente de alimentación USB estable de 5V independientemente de la fuente.
- Conector de batería y panel solar es JST 2.0 [54].
- Escalable a múltiples baterías de litio y paneles solares grandes o múltiples [54].
- Las dimensiones son 68mm*50mm.
- Algoritmos de carga/recarga integrados en el chip [54].

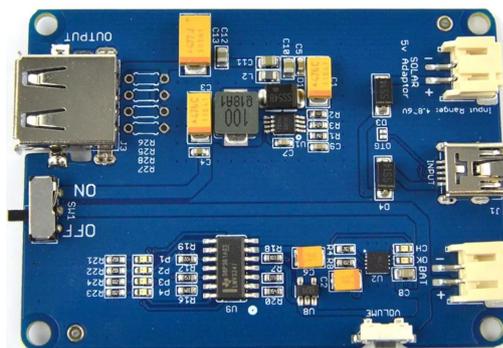


Figura 2.2: SeedStudio Lipo Rider Pro [Fuente: El Autor.]

En la tabla 2.1 se muestran las especificaciones técnicas de energía del modulo Lipo Rider Pro:

Tabla 2.1: Especificaciones de energía de SeedStudio Lipo Rider Pro

Especificación	Min	Normal	Max
$V_{inSolar}$	4,8V	5,0V	6,5V
$I_{inSolar}$	400mA	500mA	600mA
I_{carga}	0mA		1000mA
V_{bat}		4,2V	
$V_{fuenteUSB}$		5,0V	
I_{outUSB}		5,0V	

Sensor de gas CO₂ MG811 (Dióxido de carbono)

En la figura 2.3 se observa el sensor de gas de dióxido de carbono MG811, el cuál permite que un microcontrolador determine cuando se ha alcanzado o excedido un limite de gas de dióxido de carbono predeterminado. El fabricante no proporciona datos de calibración de gas, el usuario es responsable de definir dichos datos, así como la configuración de la alarma cuando la fuente se vuelve excesiva [55].

En la tabla 2.2 se observan las características básicas del sensor de gas MG811:

Calibración de sensor de gas MG811

Una vez encendido el sensor se debe seguir los siguientes pasos para calibrar el sensor de gas[56][57]:



Figura 2.3: Sensor de gas MG811. [Fuente: El Autor.]

Tabla 2.2: Características sensor de gas MG811

Características	Valor
Tipo de sensor	Electrolito sólido
Gas Objetivo	CO ₂
Tiempo de calentamiento	2 horas
Rango de detección	350 ppm - 10000 ppm
Resistencia del calentador	35Ω ± 3Ω

■ Primer paso

- Coloque el sensor en un área bien ventilada de la casa o al aire libre.
- Espere a que se caliente por lo menos dos (02) horas.
- Lea la medida: lo que recibe es un voltaje de referencia de 400 ppm.

■ Segundo paso

- Coloque el sensor en una funda que contenga aire exhalado o expuesto a un escape de un automovil.
- Espere unos minutos.
- Lea la medición: lo que recibe es un voltaje de referencia de 5000 ppm[58].

Una vez realizado los dos pasos vamos a obtener dos valores a y b que para este caso de se observan en la tabla 2.3:

Tabla 2.3: Valores analógicos medidos

PPM	Valor Analógico
400	2569
5000	1729

Con los valores obtenidos se van a ajustar la salida del sensor a partes por millos, con una ecuación de primer grado $ppm = a * Analogico + b$, de la siguiente manera:

$$ppm\ 1 = a * Analogico\ 1 + b \quad (2.1)$$

$$ppm\ 2 = a * Analogico\ 2 + b \quad (2.2)$$

Donde:

$$ppm\ 1 = 400 \quad (2.3)$$

$$ppm\ 2 = 5000 \quad (2.4)$$

$$Analogico\ 1 = 2569 \quad (2.5)$$

$$Analogico\ 2 = 1729 \quad (2.6)$$

Despejando ambas ecuaciones obtenemos:

$$a = \frac{ppm\ 2 - ppm\ 1}{Analogico\ 2 - Analogico\ 1} \quad (2.7)$$

Reemplazamos los valores con los datos obtenidos en las mediciones obteniendo:

$$a = \frac{5000 - 400}{1729 - 2569} \quad (2.8)$$

$$a = -5,47619 \quad (2.9)$$

Una vez obtenido el valor de a , se substituye y se encuentra el valor de b :

$$b = \frac{ppm\ 1}{a * Analógico\ 1} \quad (2.10)$$

$$b = \frac{400}{-5,47619 * 2569} \quad (2.11)$$

$$b = 14468,3 \quad (2.12)$$

Con los valores de a y b , se reemplazan en la ecuación de ajuste de la siguiente manera:

```

1. Inicializar:
a = -5.47619 ; b = 14468.3 ;
2. Leer ADC
v=leerADC(ADC)
3. Calcular valor en ppm
mgCO2=v * a + b ;

```

Sensor MHZ-19B

El sensor de gas MHZ-19B se utiliza para medir la concentración de CO₂ en el aire, utilizando el principio de infrarrojo no dispersivo (**NDIR**, por sus siglas en inglés, Non Dispersive Infrared Detector), no depende del oxígeno y de larga duración.

El sensor consta de una señal de salida analógica y dos salidas digitales con protocolos de comunicación de señal **UART** y **PWM**. El precalentamiento del sensor es de 3 minutos para que el valor de CO₂ se estabilice [59], [60].

El principio del sensor infrarrojo no dispersivo (**NDIR**), funciona cuando un haz de infrarrojos pasa a través del gas CO₂ que calcula midiendo la cantidad de radiación infrarroja absorbidas mediante el receptor IR a la frecuencia requerida [61].

Este sensor que se muestra en la figura 2.4 de tipo común de tamaño pequeño, tienen un mayor consumo de energía que los basados en métodos electroquímicos, son usados para el monitoreo en tiempo real, por su precisión, confiabilidad y durabilidad [62].



Figura 2.4: Sensor de gas MHZ-19B. [Fuente: El Autor.]

Entre las características encontramos la tabla 2.4:

Tabla 2.4: Características sensor de gas MHZ-19B

Características	Valor
Tipo de sensor	NDIR
Gas Objetivo	CO ₂
Tiempo de calentamiento	3 min
Rango de detección	0 - 5000 ppm
Salidas	UART, PWM
Vida Útil	>5 años

2.1.2. Protocolo de comunicación del sensor MHZ-19B

Para leer la concentración de gas de CO₂ se puede hacer directamente vía [UART](#), sin la necesidad de calcular la concentración de ppm. Las configuraciones generales se encuentran en la tabla 2.5:

Tabla 2.5: Configuración general del sensor de gas MHZ-19B

Dato	Valor
Baud rate	9600
Date byte	8 byte
Stop Byte	1 byte

Las características de cada comando o retornos utilizados para la comunicación son:

- Contienen 9 bytes (byte 0 8).
- Byte inicial fijado en 0 x FF.
- El comando contiene el número de sensor (el valor predeterminado de fábrica es 0 x 01).
- Termina con el valor de la prueba de Checksum de comprobación.

Para acceder a los registros de los datos del sensor se cuenta con la lista de direcciones que se detalla en la tabla 2.6

Tabla 2.6: Lista de comandos

Byte	Comando
0 x 86	Concentración de Gas
0 x 87	Calibración de punto cero (Zero)
0 x 88	Calibración de punto span (SPAN)

En la tabla 2.7 se detalla la petición de lectura de concentración de gas CO₂.

Tabla 2.7: Lectura de concentración de gas CO₂

Comando de envío								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Byte de inicio	Sensor No.	Comando	-	-	-	-	-	Comprobar valor
0 x FF	0 x 01	0 x 86	0 x 00	0 x 79				

En la tabla 2.7 se indica las características de la trama (en inglés *frame*) de respuesta que se recibe del sensor MHZ-19B.

Tabla 2.8: Valor de retorno de concentración de gas CO₂

Valor de retorno								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Byte de inicio	Sensor No.	Con. alto nivel	Con. bajo nivel	-	-	-	-	Comprobar valor
0 x FF	0 x 86	0 x 02	0 x 60	0 x 47	0 x 00	0 x 00	0 x 00	0 x D1

Una vez se hayan recibido los valores de retorno, para encontrar la concentración de gas se calcula aplicando la ecuación 2.13:

$$\text{Concentracindegas} : \text{nivelalto} * 256 + \text{nivelbajo} \quad (2.13)$$

2.1.3. Caja de aplicación IoT

Se ha diseñado una caja para la protección de la placa PCB en Inventor para luego imprimirla en 3D. El diseño toma en cuenta el tamaño y la posición de la placa PCB, modulo de gestión de energía, batería, cargador solar, sensor MG-811 y un switch de encendido y apagado además se diseño un agujero para la antena 2.5, donde se diseñó un modelo compacto para colocar los sensores de una manera que tengan una fácil recepción del aire exterior.

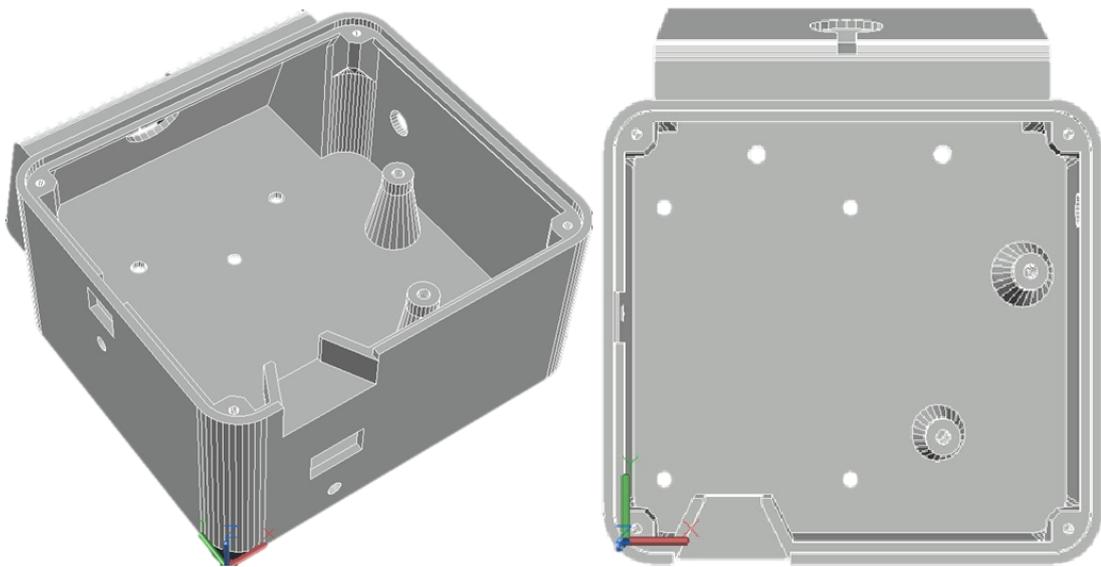


Figura 2.5: Vista lateral y superior de la caja. [Fuente: El Autor.]

De igual manera se diseñó la tapa en acrílico para sellar y proteger los módulos de la intemperie, en la imagen 2.6 se muestra la tapa con una vista superior, donde se realizaron dos agujeros para que los dos sensores tengan salida al aire libre.



Figura 2.6: Vista superior de la tapa para la caja. [Fuente: El Autor.]

La caja fue diseñada para que los sensores de gas MHZ-19B y MG811 además de la antena cuenten con salida. De esta manera los nodos podrán realizar las mediciones de CO₂ y comunicarse con el Puerta de enlace LoRa (*Gateway*).

2.1.4. Placa PCB

El diseño de la placa se desarrolló con el software Autodesk Eagle, la placa PCB (Printed Circuit Board, en inglés) se ubican los siguientes módulos: HTCC-AB 01, MH-Z19B, además de pines para la unión con la placa del modulo de gestión de energía y con el sensor MG-811. El desarrollo con las consideraciones se observa en la imagen 2.7.

Se realizó la placa PCB compacta, de esta manera para que los sensores tengan la opción de tener contacto con la parte exterior, para lograr su objetivo de medir con precisión la calidad del aire.

2.1.5. Comunicación entre la placa de desarrollo HTCC-AB 01 hacia los sensores de gas de CO₂

En la figura 2.8, se observa a la placa HTCC-AB 01 que recoleta dos valores de señal de los dos sensores MHZ-19B y MG811. El sensor MHZ-19B se comunica por medio de un serial (Rx-Tx), la comunicación UART como se

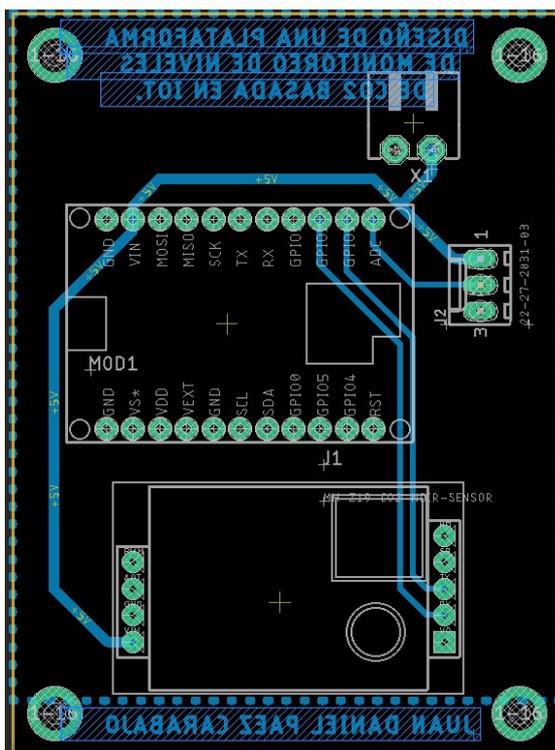


Figura 2.7: Placa en vista 2D. [Fuente: El Autor.]

explicó anteriormente en las tablas 2.7 y 2.8, se usan dos vectores tipo byte donde se envía al sensor para realizar la lectura de concentración y otro para el retorno de la concentración de CO_2 , el sensor MG811 se comunica mediante una entrada analógica con la placa de desarrollo. Los dos valores de señal obtenidos una vez tratados, se guardan en variables tipo flotante.

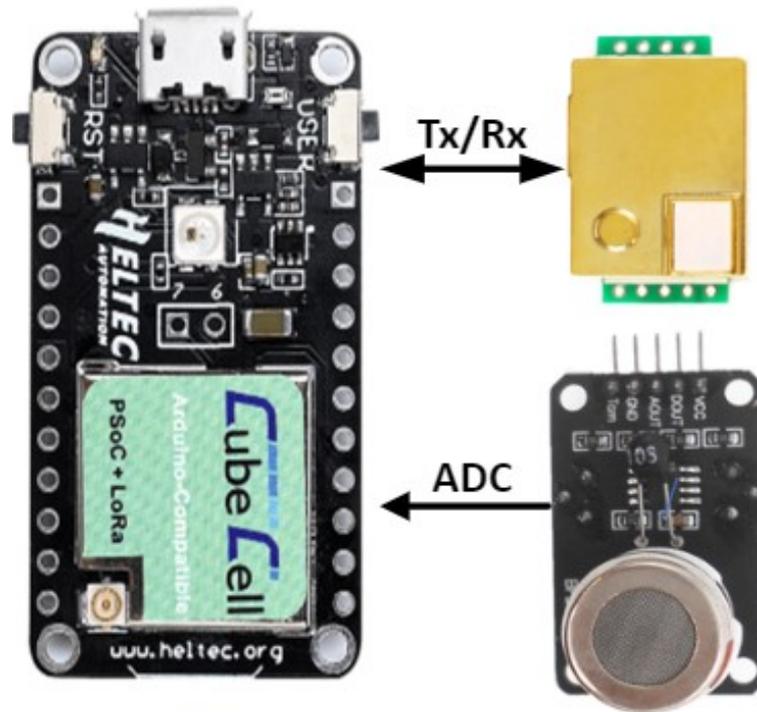


Figura 2.8: Esquema de comunicación de la placa de desarrollo HTCC-AB 01 hacia los sensores. [Fuente: El Autor.]

Para poder alimentar a los dos sensores se utilizó el módulo SeedStudio Lipo Rider Pro que permite tener el voltaje y la corriente necesaria para que funcionen correctamente.

2.1.6. Alimentación del circuito

Para la alimentación de los sensores y placa de desarrollo se utilizó una batería de litio plana de 3.7V y un panel solar de 6V con el modulo SeedStudio Lipo Rider Pro que eleva el voltaje a 5V y permite cargar la batería mediante un cargador normal y energía solar. Los 5V que tiene de salida del modulo alimentan a la placa de desarrollo y los sensores.

El esquema de alimentación se indica en la figura 2.9.

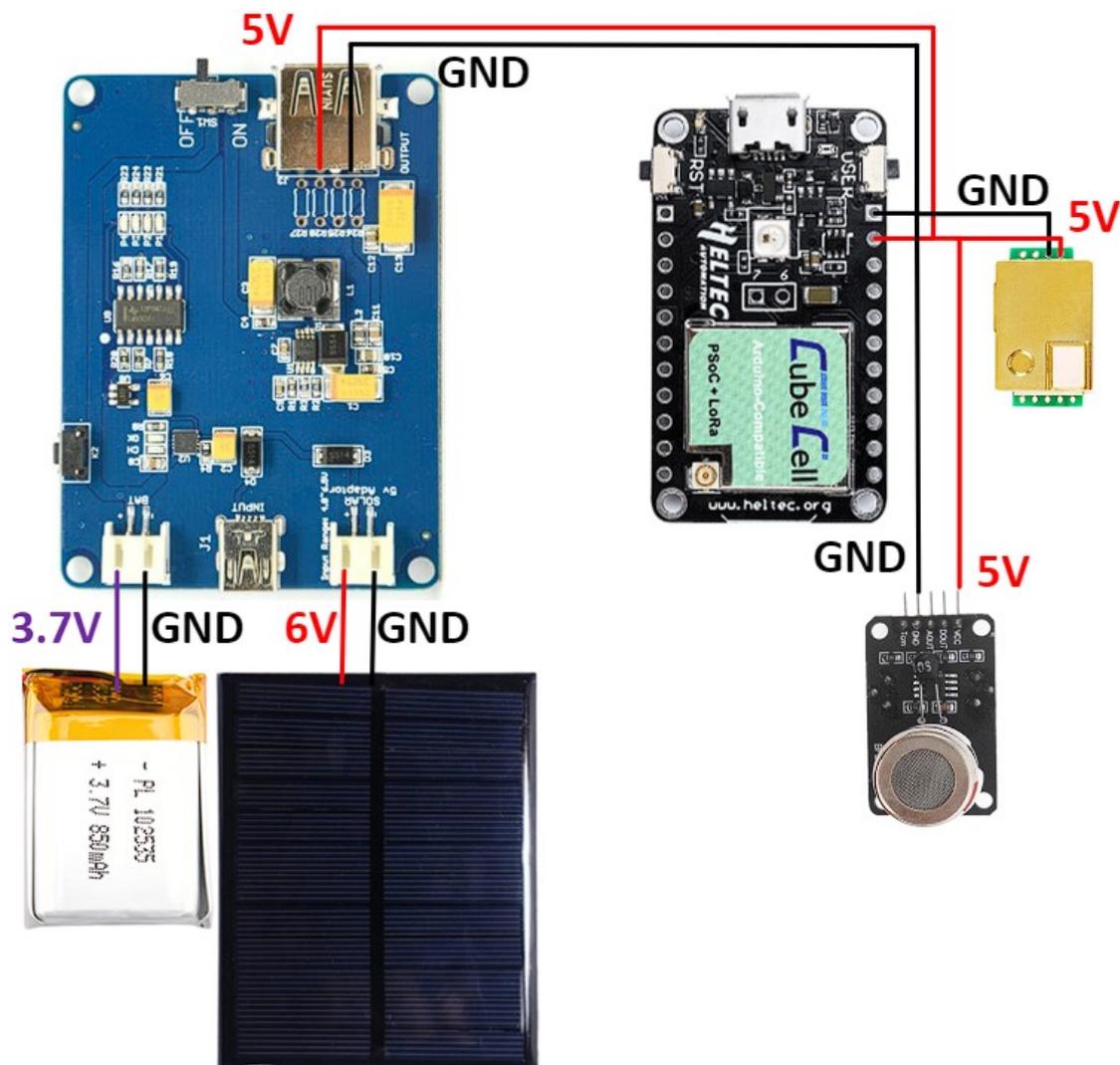


Figura 2.9: Esquema de alimentación eléctrica. [Fuente: El Autor.]

2.2. Etapa Comunicación LoRa

A continuación se presenta la topología de red utilizada, los protocolos de comunicación con el [Gateway](#) y la comunicación LoRa.

2.2.1. Topología de red

La conexión entre los nodos y el Gateway LoRa se usa una red de sensores inalámbricos tipo estrella en la cuál cada nodo recolecta los valores de CO₂ del ambiente y los envía al Gateway mediante la red LoRa en la frecuencia 915MHz. Cuando las variables se receiptan en el gateway este se encarga de enviar al servidor Chirpstack que se encuentra montado en el servidor Amazon

Web Services (AWS).

Una vez que los datos se encuentran en el servidor AWS, se realiza una petición **CURL** del inglés *Client for URLs*, y es una herramienta de línea de comandos para transferir datos a través de la red utilizando protocolos **HTTP**, **HTTPS**, **FTP**, **SFTP**.

CURL puede realizar peticiones como GET, POST y HEAD al servidor, cargar archivos, descargar páginas **HTML**, etc [63], [64].

Donde:

GET: Solicita al servidor una información o recurso concreto.

POST: Método que se encarga de modificar datos.

HEAD: Reside información acerca del documento.

De esta manera se obtiene los valor para la aplicación de monitoreo basada en flutter.

En la imagen 2.10 se observa de manera gráfica la topología de red utilizada para el desarrollo del proyecto, en primera instancia los nodos de CO₂ pueden estar colocados en diferentes escenarios para monitorear la calidad del aire, donde se conectan al Gateway a través de la red LoRaWAN.

El LoRa Gateway actúa como el enlace entre los Nodos CO₂ y la red LoRaWAN, donde se transmiten a un servidor central ChirpStack, que permite la recolección de datos para ser almacenados mediante el la maquina virtual alojada en Google Cloud, para luego se visualizadas en la aplicación de GITEL CO₂.

En las figuras 2.22, 2.23 se explica a profundidad el proceso de adquisición y almacenamiento de datos en el servidor ChirpStack.

2.2.2. Registro de Nodos LoRa con ChirpStack

Para registrar los Nodos en el servidor LoRa se debe crear un perfil de dispositivos que contiene la configuración inicial y las capacidades que necesita el servidor ChirpStack para el acceso de radio LoRaWAN. En la figura 2.11 se observa las configuraciones que utilizará nuestra aplicación.

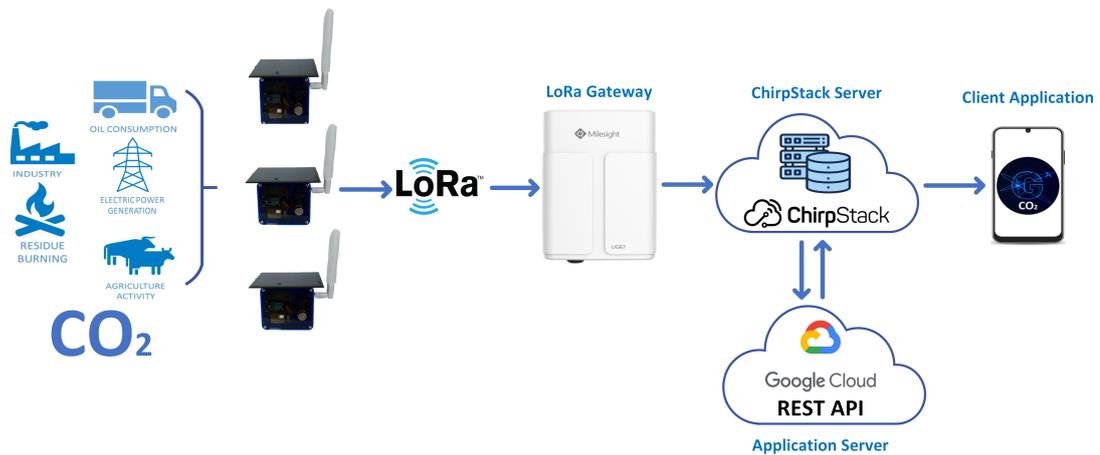


Figura 2.10: Topología de red. [Fuente: El Autor.]

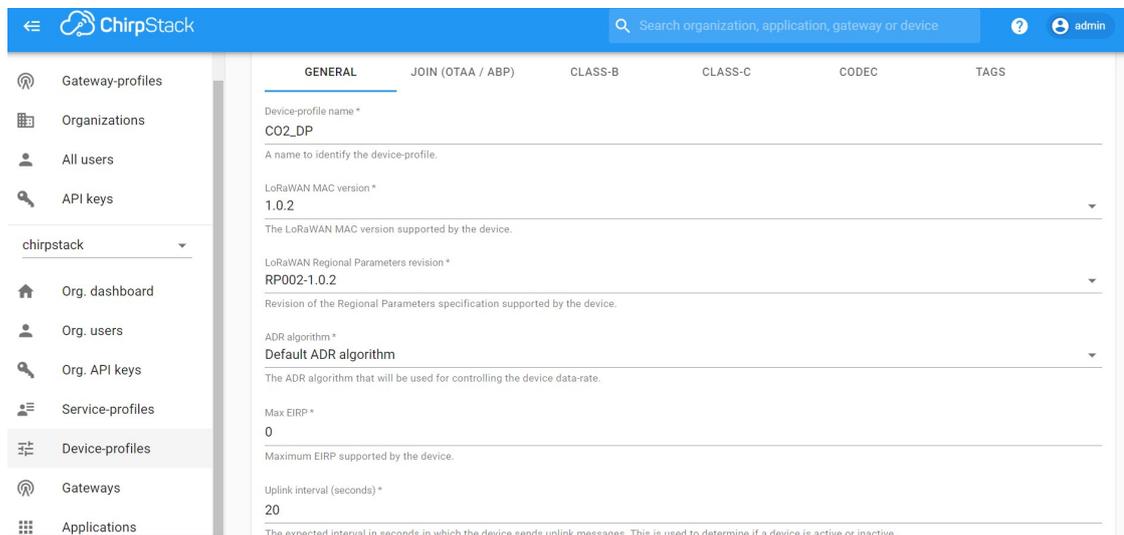


Figura 2.11: Creación de perfil de dispositivos. [Fuente: El Autor.]

Además se deberá activar el método de unión al servidor, existen dos maneras para unirse a la red LoRaWan:

Activación sobre el aire (OTAA) Para poder unirse el dispositivo final realiza una petición de unión con la red LoRaWAN, mediante una dirección dinámica y claves de seguridad[65].

Activación por personalización (ABP) En la activación de **ABP**, requiere claves fijas y de sesión almacenadas en el dispositivo final, el método no enviará una solicitud de unión y no necesita confirmación a través del enlace enlace descendente o en inglés *downlink* [66].

Después de ser analizados los métodos de solicitud de unión a la red

LoRaWAN, se llegó a la conclusión que la mejor opción será OTAA debido a las seguridades que brinda este método, en la figura 2.12 se habilita el método de unión al servidor de los dispositivos finales.

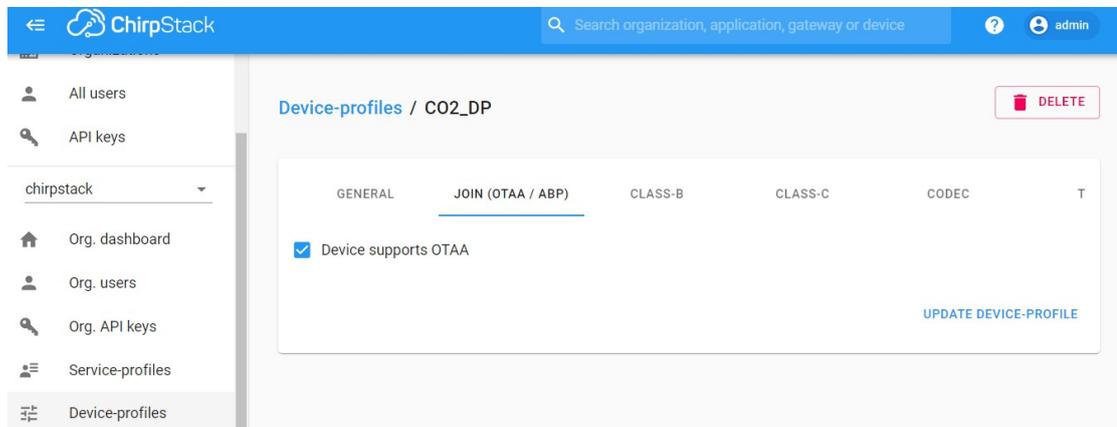


Figura 2.12: Método de conexión OTAA. [Fuente: El Autor.]

Además se deberá configurar el códec de carga útil traducido del inglés "payload codec" que permite decodificar una matriz de bytes a un objeto, lo que nos permite decodificar nuestra trama de datos que en inglés es *frame* de datos enviados por los dispositivos final, en la figura 2.13 se muestra el código utilizado para la decodificación.

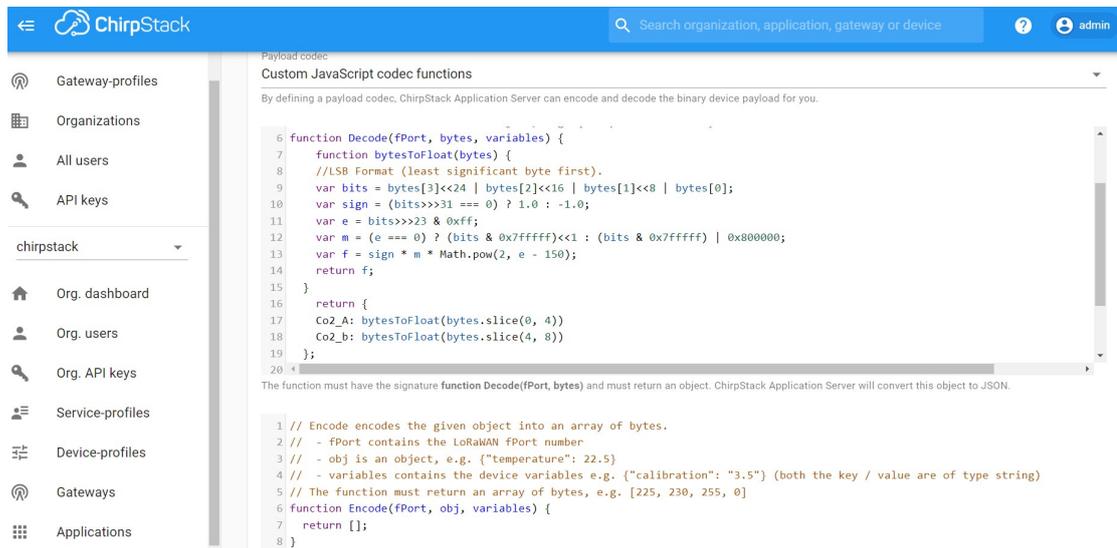


Figura 2.13: Decodificación de datos. [Fuente: El Autor.]

Una vez creado el perfil de dispositivos, se puede registrar los dispositivos finales para esto vamos a la opción de aplicación como se muestra

en la figura 2.14, se crea una nueva aplicación donde se coloca el nombre la descripción y el perfil de servicio.

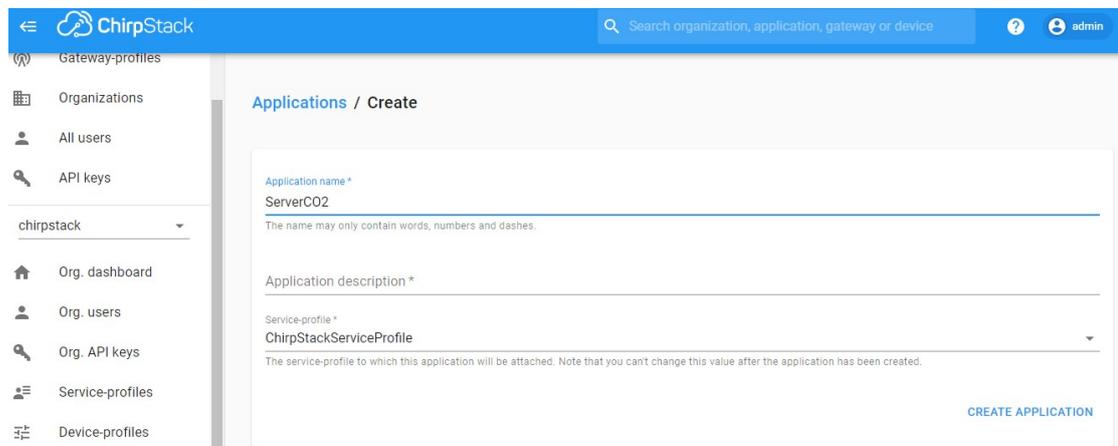
The screenshot shows the ChirpStack web interface. On the left is a navigation sidebar with options like Gateway-profiles, Organizations, All users, API keys, and various org and device profiles. The main content area is titled 'Applications / Create'. It contains a form with three main sections: 'Application name *' with the value 'ServerCO2' and a note 'The name may only contain words, numbers and dashes.'; 'Application description *' which is currently empty; and 'Service-profile *' with a dropdown menu showing 'ChirpStackServiceProfile' and a note 'The service-profile to which this application will be attached. Note that you can't change this value after the application has been created.' A blue 'CREATE APPLICATION' button is located at the bottom right of the form.

Figura 2.14: Creación de nueva aplicación llamada ServerCO2 [Fuente: El Autor.]

Una vez creada la aplicación se van a agregar los dispositivos finales, en la figura 2.15 se muestra parámetros necesarios para registrar: El nombre del dispositivo, descripción del dispositivo, deviceEUI que será un identificador único para cada dispositivo y el el perfil de dispositivos que fue creado en el paso anterior.

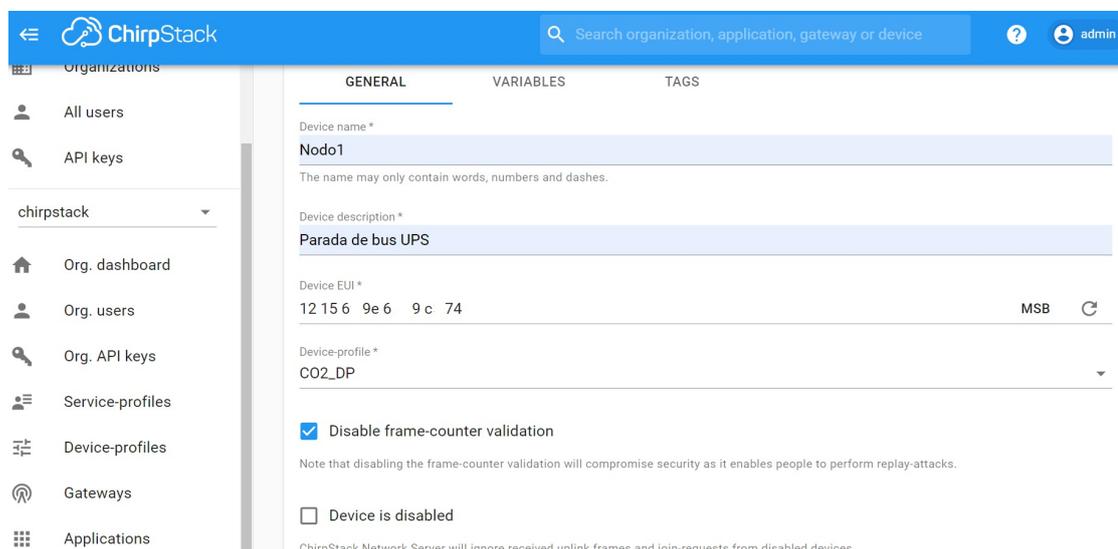
The screenshot shows the 'GENERAL' tab of the device registration form in ChirpStack. The form fields are: 'Device name *' with value 'Nodo1' and a note 'The name may only contain words, numbers and dashes.'; 'Device description *' with value 'Parada de bus UPS'; 'Device EUI *' with value '12 15 6 9e 6 9 c 74' and a 'MSB' button; 'Device-profile *' with a dropdown menu showing 'CO2_DP'. There are two checkboxes: 'Disable frame-counter validation' which is checked, and 'Device is disabled' which is unchecked. A note below the checkboxes states: 'Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.' and 'ChirpStack Network Server will ignore received uplink frames and join-requests from disabled devices.'

Figura 2.15: Registro de dispositivos finales. [Fuente: El Autor.]

2.2.3. Creación de claves API globales

Para acceder a la base de datos se deberá crear una clave API global, ChirpStack permite crear de una manera sencilla, por lo que en la figura 2.16, se

muestra el único paso que se debe realizar para crearla. Se coloca el nombre y se agrega para tener una clave global.



Figura 2.16: Creación de clave API global. [Fuente: El Autor.]

2.2.4. Comunicación LoRa

La comunicación entre los nodos y el gateway se realiza cuando el nodo solicita una petición **OTAA** para poder unirse mediante a la red LoRaWAN. **OTAA** es un método que intercambia dos mensajes entre el nodo y gateway para el procedimiento de conexión. Los nodos solo se podrán unir a la red LoRaWAN después del procedimiento de conexión [67].

El procedimiento para la conexión y envío de datos se describen a continuación.

Conexión de nodos

1. Inicializa sistema, LoRaWan, Softserial, (GPIO 2-TX, GPIO 1-RX),
byte devEUI [], appKey [], leerMhCO2, abcON, rangeID, v, a, b;
 2. PreparaFrame, Inicializa Serial(9600 baudios), Inicializa
SoftSerial(9600 baudios), deviceState=0;
 3. Caso 1: deviceState==0
Imprime parametros LoRaWan
Realiza enlace
deviceState=1
 4. Caso 2: deviceState==1
Enlace LoRaWan establecido
deviceState=2
- Conexión de nodos

```
5. Caso 3: deviceState==2
   preparaTxFrame(appPort=2)
   Envío datos por LoRaWan
   deviceState=3

6. Caso 4: deviceState==3
   Setea siguiente tiempo de envío
   deviceState=4

7. Caso 5: deviceState==4
   Modo bajo consumo
   deviceState=2

8. deviceState=0
```

2.3. Front-end y Back-end de aplicación de monitoreo

A continuación se describe el proceso de desarrollo del Front-end y Back-end de la plataforma de monitoreo.

2.3.1. Front-end

Flutter es el framework de desarrollo para la aplicación móvil de monitoreo, la interfaz de la aplicación debe ser fácil de utilizar y intuitiva para que los usuarios interactúen.

Para la interfaz se realizó un prototipo de aplicación móvil, es una de las técnicas más fáciles de usar para anticipar cómo se desarrollará la aplicación en términos de diseño sin tener que comenzar a programar o escribir ningún código. Los prototipos que se realizaron fueron para un Administrador que le permite cambiar la ubicación de los Nodos LoRa y otra para los usuarios finales, a continuación se presenta en las secciones los prototipos iniciales de las pantallas de la aplicación móvil.

2.3.2. Modo Usuario

El modo de usuario ofrece únicamente las funciones para observar los valores de los sensores de CO₂.

Pantalla Usuario

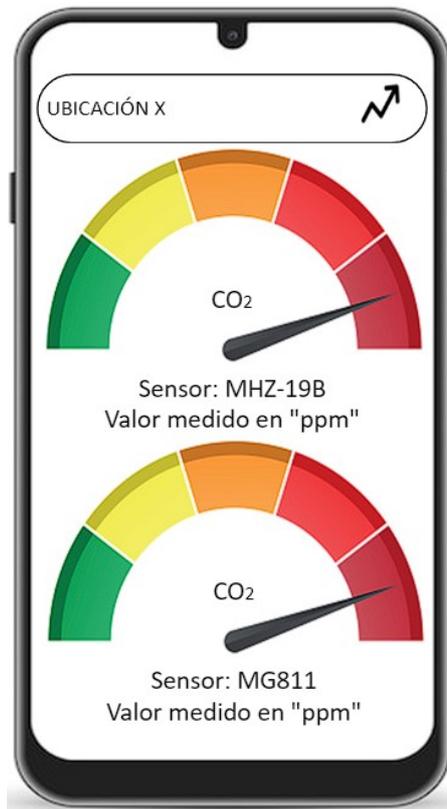
El prototipo de la interfaz inicial del usuario se muestra en la figura 2.17, el usuario tendrá dos pantallas en las cuales podrá revisar las ubicaciones de los nodos de manera gráfica en el mapa o desplegar un listado con las ubicaciones donde se encuentran. Para poder cambiar de pantalla entre el mapa 2.17(a) y el listado 2.17(b), la aplicación deberá tener un botón que permita realizar el intercambio de las pantallas.



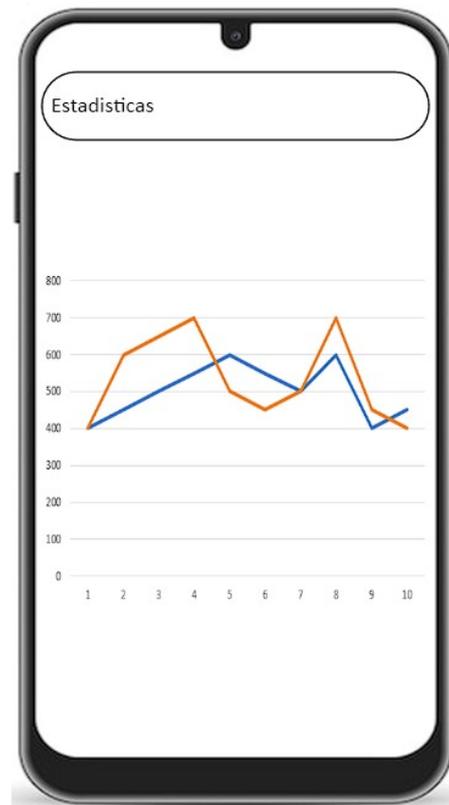
Figura 2.17: Prototipo pantallas inicial del usuario. [Fuente: El Autor.]

Después que el usuario seleccione un nodo de referencia, podrá observar los niveles de CO₂ de los nodos como se muestra en la figura 2.18, primero se desplegará la interfaz como se muestra en la figura 2.18(a), donde podrá ver

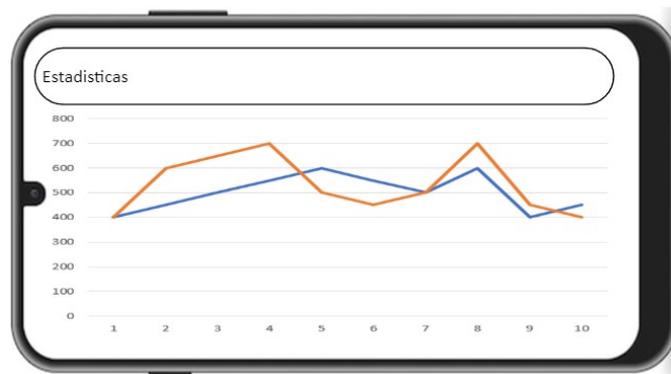
los valores de los dos sensores de CO₂ del nodo, además tendrá la opción de visualizar los valores de manera lineal por lo que podrá cambiar de pantalla y observar como en la figura 2.18(b), para poder observar de mejor manera los valores lineales se podrá girar la pantalla como se muestra en la figura 2.18(c).



((a)) Valores de CO₂ de los sensores



((b)) Estadísticas de CO₂



((c)) Interfaz para la muestra de niveles de CO₂

Figura 2.18: Prototipo pantallas inicial del usuario. [Fuente: El Autor.]

2.3.3. Modo Administrador

El modo administrador ofrece las funciones de inicio de sesión, visualización de datos recibidos desde el servidor, reubicación de ubicaciones de sensores.

Inicio de Sesión

Para poder ingresar al modo administrador, se debe iniciar sesión como se indica en la figura 2.19, esto solo aplica para los administradores o técnicos, con una cuenta proporcionada anteriormente, para que puedan realizar un cambio de ubicación de los nodos y observar los datos recibidos del servidor.



Figura 2.19: Prototipo de pantalla de inicio de sesión. [Fuente: El Autor.]

Cambio de Ubicación de Nodos

Solo el administrador tendrá acceso a la edición de ubicaciones como se muestra en la figura 2.20, para poder editar la ubicación de un nodo el

administrador debe ingresar al listado de ubicaciones como se indica en la figura 2.20(a), una vez que se haya ingresado al nodo se podrá cambiar la ubicación de dos maneras, la primera puede cambiar con la ubicación en tiempo real o la segunda colocando manualmente la latitud y la longitud de la nueva ubicación donde se vaya a colocar el nodo, luego deberá presionar el botón editar para cambiar y guardar la nueva ubicación como se observa en la figura 2.20(b).



Figura 2.20: Prototipo pantallas cambio de ubicación de los nodos. [Fuente: El Autor.]

Datos recibidos desde el servidor

Para saber si el nodo tiene conexión se tiene la opción de revisar los registros enviados al servidor desde la aplicación sin la necesidad de ingresar al servidor ChirpStack, en la figura 2.21 se muestra la forma en la que se debe ingresar a los registros.

Para ingresar a los registros se debe ingresar al sensor que se quiere revisar como en la figura 2.21(a) y luego al registro del nodo como en la figura 2.21(b), donde se observarán los datos enviados desde los nodos como los

valores de CO₂ de los sensores y la hora de envío.

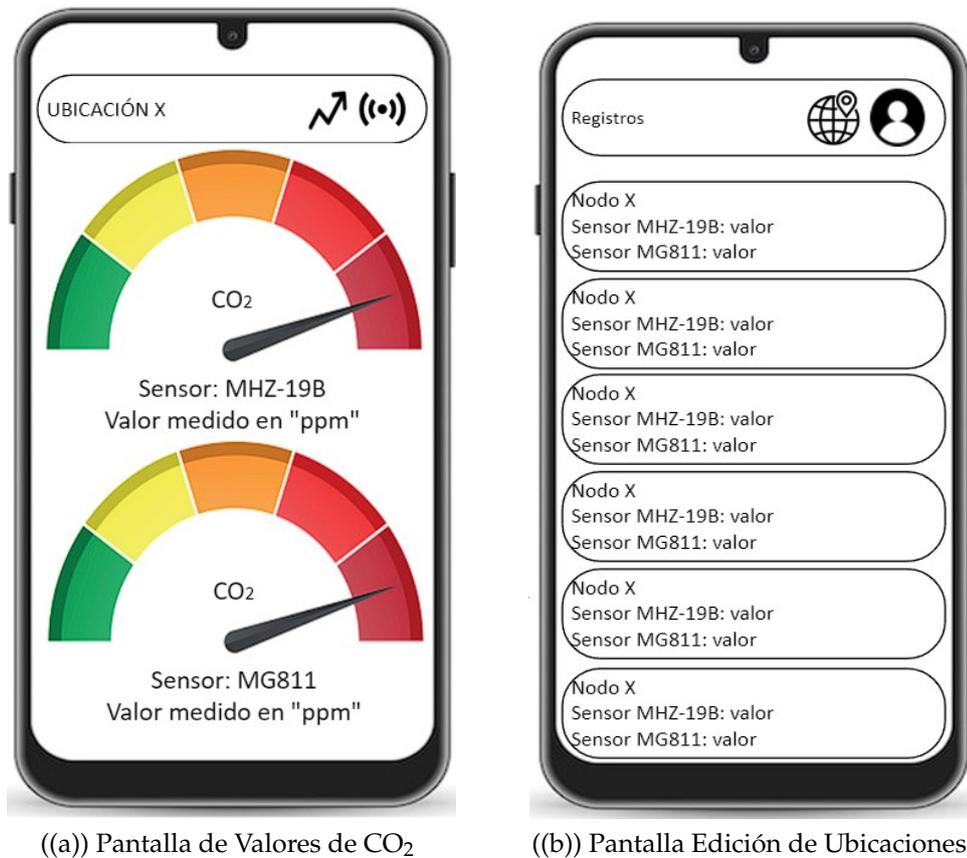


Figura 2.21: Prototipo pantallas para observar registros de nodos. [Fuente: El Autor.]

2.4. Desarrollo de Servidor

A continuación se explica el proceso utilizado para el desarrollo del back-end de la aplicación de monitoreo, se divide en dos procesos:

2.4.1. API REST del servidor de aplicaciones ChirpStack

Una API REST es utilizada para construir la integración con la aplicación de monitoreo y proporciona una solución para leer las tramas de datos enviadas al servidor en nuestro caso.

Utilizamos una API REST debido a que ChirpStack no almacena los datos recibidos, ya que principalmente está diseñada para la gestión de las redes LoRaWAN, y su enfoque es la transmisión de los datos a dispositivos finales, por lo que necesariamente se necesita crear un servidor que permita almacenar

datos para luego ser consumidos en nuestra aplicación.

En la Figura 2.22 se muestran los pasos para crear un API REST del servidor de aplicaciones ChirpStack:

1. Para la autenticación se debe crear una clave " API TOKEN " obtenida mediante la interfaz web.
2. Se debe ingresar a la interfaz API REST del servidor de aplicaciones ChirpStack por ejemplo: `http://localhost:8080/api`
3. A continuación vamos al apartado `/api/devices/dev_eui/events` y seguimos los pasos para crear una dirección CURL para obtener las tramas de datos de los nodos de CO₂, para este caso lo único paso que debemos seguir es la colocación del `dev_eui` de cada nodo.
4. Como resultado se genera una dirección tipo CURL para la obtención de tramas de datos.

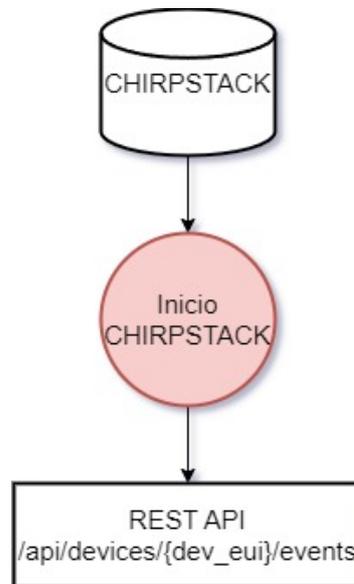


Figura 2.22: Flujograma API REST del servidor de aplicaciones ChirpStack. [Fuente: El Autor.]

2.4.2. Back-End

Como se explicó en la sección anterior, ChirpStack no almacena los datos de los sensores por lo que se creó una máquina virtual para el servicio de almacenado de datos.

Donde se creó dos servicios el primero daemon `sensor.service` permite la recolección y almacenamiento de los datos de CO₂. Y el segundo daemon `sensorpy.service` que lee los datos recolectados y envía a guardarlos a una base de datos, para este caso se ha reutilizado la base de datos de ChirpStack basada en MySQL.

En la figura 2.23 se observa los dos servicios realizados para la lectura y almacenado de datos en la base de datos, a continuación se describen los dos procesos:

daemon `sensor.service` Este servicio se encarga de leer las tramas de datos de los nodos del servidor ChirpStack mediante el uso del proceso "CURL", como se mencionó previamente. Los datos obtenidos se almacenan posteriormente en un archivo de texto.

daemon `sensorpy.service` Este servicio utiliza un script de Python para leer los datos recolectados por el primer servicio y luego los envía para que sean almacenados en la base de datos de ChirpStack.

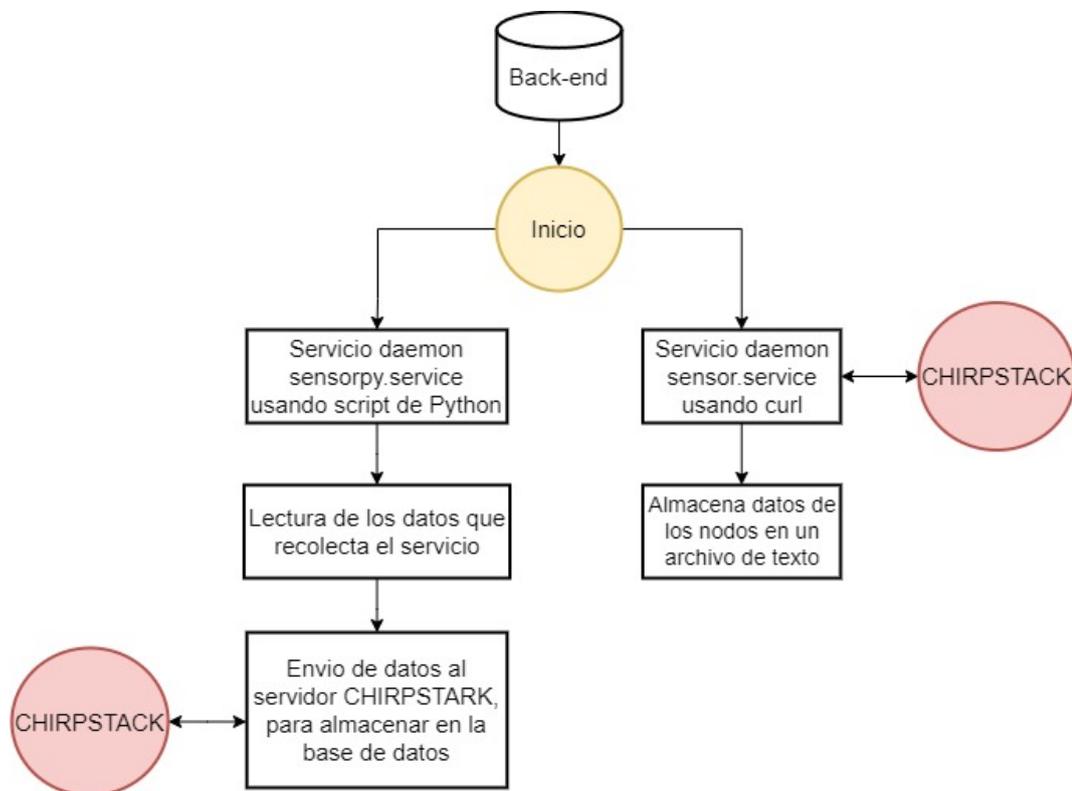


Figura 2.23: Flujograma de funcionamiento de los servicios Front-End. [Fuente: El Autor.]

2.4.3. Máquina virtual en Google Cloud

El servidor se encuentra alojado en una máquina virtual de Google Cloud, una vez que se haya creado la máquina virtual, se deben seguir los siguientes pasos para poner en línea el servidor:

1. Ingresar como root

```
sudo su
```

2. Instalar librerías de python3

```
sudo apt install libpq-dev python3-dev
```

```
sudo apt install python3-pip
```

```
pip3 install psycopg2
```

3. Copiar carpeta server a `/usr/local/bin/`

```
//tar -xf filename.tar.gz//
```

```
cp -R server /usr/local/bin/
```

4. Abrir directorio

```
cd /usr/local/bin/server
```

5. Permisos de ejecución sh

```
chmod +x init.sh
```

6. Copiar servicios a `/etc/systemd/system/`

```
cp sensor.service /etc/systemd/system/
```

```
cp sensorpy.service /etc/systemd/system/
```

7. Restaurar variable de servicios

```
systemctl daemon-reload
```

8. Iniciar servicios y comprobar que se ejecutaron correctamente

```
sudo systemctl start sensor
```

```
sudo systemctl status sensor
```

```
sudo systemctl start sensorpy
```

```
sudo systemctl status sensorpy
```

2.5. Desarrollo de Aplicación Móvil

El propósito de la aplicación es visualizar los valores de concentración de CO₂ de una manera gráfica y fácil de usar, observando los valores de CO₂ en tiempo real y un historial.

La aplicación se encuentra desarrollada en el framework flutter, en la figura 2.24 se observa el proceso de funcionamiento de la aplicación. el proceso de la aplicación se describe a continuación:

- **Inicio:** La aplicación se inicializa con el logo del grupo Gitel.
- **Ver Mapa:** Por defecto se va a observar el mapa de Google, con un PIN señalando las ubicaciones de los nodos de CO₂. Se puede escoger entre dos vistas, la primera ya mencionada el mapa de Google y la segunda se podrá en listar las ubicaciones.
- **Gráficos:** Cuando se ingrese a un nodo se van a observar:
 - Visualizar en tiempo real dos manómetros con la calidad de aire, equivalentes a los dos sensores de CO₂ colocados en cada nodo.
 - También se puede elegir la opción de visualizar en un gráfico lineal con los últimos 1000 registros de CO₂ de los dos sensores.
- **Inició sesión:** Para los usuarios que tengan acceso al usuario se van a desbloquear funciones especiales como:
 - Edición de ubicaciones de los nodos de CO₂ en el mapa.
 - Acceso y visualización de las 180 últimas tramas de datos de los sensores.
- **No inició sesión:** Solo tendrá acceso a los gráficos, no podrá editar ubicaciones y tampoco observar las tramas de datos.
- **Servicio ChirpStack:** El usuario y contraseña se encuentran creados en la base de datos de ChirpStack, también el valor de los sensores son consultados desde la base de datos.

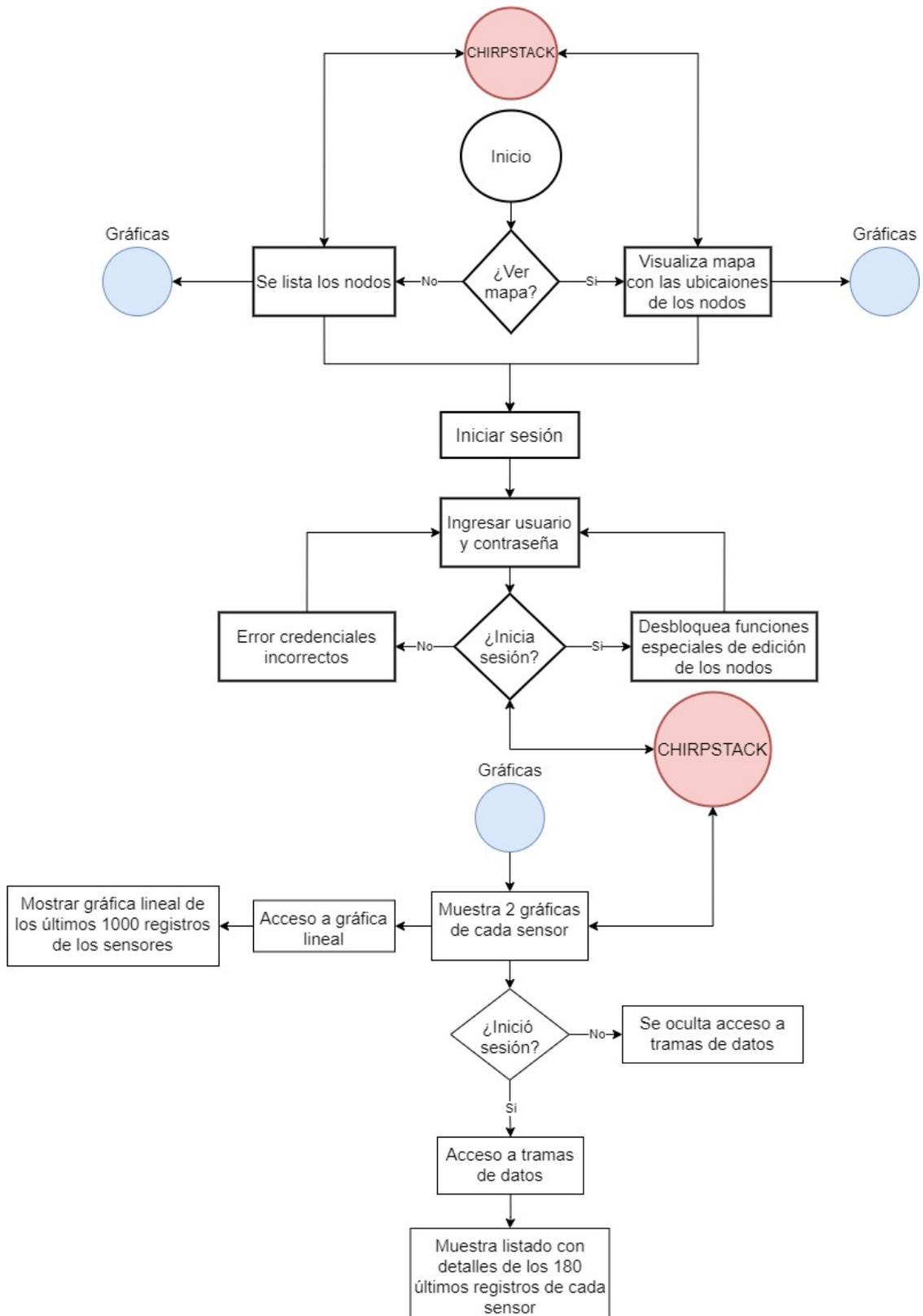


Figura 2.24: Flujograma de funcionamiento Frontend. [Fuente: El Autor.]

2.5.1. Librerías utilizadas

Las librerías utilizadas para el desarrollo de la aplicación son las siguientes:

cupertino_icons 1.0.2: Widgets de Cupertino que contiene activos de iconos utilizados basado en estilo de iconos de Apple.

postgres 2.6.1: Permite consultar y conectarse a bases de datos PostgreSQL.

syncfusion_flutter_gauges 21.2.9: Widgets de visualización de datos, utilizado para visualizar datos de manera gráfica de CO₂.

graphic 2.2.0: Complemento que permite la reproducción en tiempo real de imágenes fijas.

google_maps_flutter 2.2.0: Proporciona un widget de Google Maps.

geolocator 9.0.1: Complemento de geolocalización que permite obtener la ubicación actual del dispositivo.

crypto 3.0.3: Conjunto de funciones para la conversión de datos a otro formato.

shared_preferences 2.1.2: Permite almacenar o guardar datos en los diferentes dispositivos que se ejecuta la aplicación.

flutter_launcher_icons 0.13.1: Permite actualizar el icono de inicio de la aplicación.

2.5.2. Widgets de aplicación Flutter

Flutter es un framework de código abierto desarrollado por Google que permite a los desarrolladores crear aplicaciones multiplataforma de alta calidad para dispositivos móviles, web y escritorio. Los "widgets" se emplean como elementos de interfaz de usuario (UI) en Flutter, cumpliendo un papel fundamental en la construcción de la interfaz gráfica de la aplicación. En otras palabras se hace referencia a una representación gráfica o esquemática de la jerarquía de widgets en una aplicación, ofreciendo una visión visual de cómo se organizan y anidan los widgets en la interfaz de usuario.

En el proceso del diseño de la interfaz gráfica de la aplicación de la

plataforma de monitoreo, se emplea una estructura jerárquica de widgets, como se muestra en la figura 2.25. Estos widgets se anidan unos dentro de otros con el propósito de lograr la disposición y apariencia deseadas. En esta estructura, se utiliza diversos elementos, como contenedores, elementos de diseño, botones, listas, texto, mapas, entre otros, que desempeñan un papel esencial en la creación de la apariencia y la interactividad de la aplicación.

Como se explicó en la figura 2.25, se observa la estructura jerárquica de widgets que se utilizan en la aplicación, donde se utiliza un widget Scaffold que básicamente es una estructura básica que encapsula varios elementos comunes, entre los cuales se encuentran los siguientes:

ListView.builder: Widget que genera una lista desplazable que muestra elementos de manera dinámica y eficiente. En este caso va a generar una lista con todos los nodos que tenemos registrados con su Nombre, descripción y un icono, también se va muestra un icono para la opción de editar ubicaciones.

Center: Widget que se utiliza para centrar a su único "hijo", en los dos ejes cartesianos "X y Y".

GoogleMap: El widget que mediante del paquete "google_maps_flutter" permite la integración de mapas interactivos de Google Maps dentro de la aplicación, para esto se debe obtener una API key válida de Google Maps.

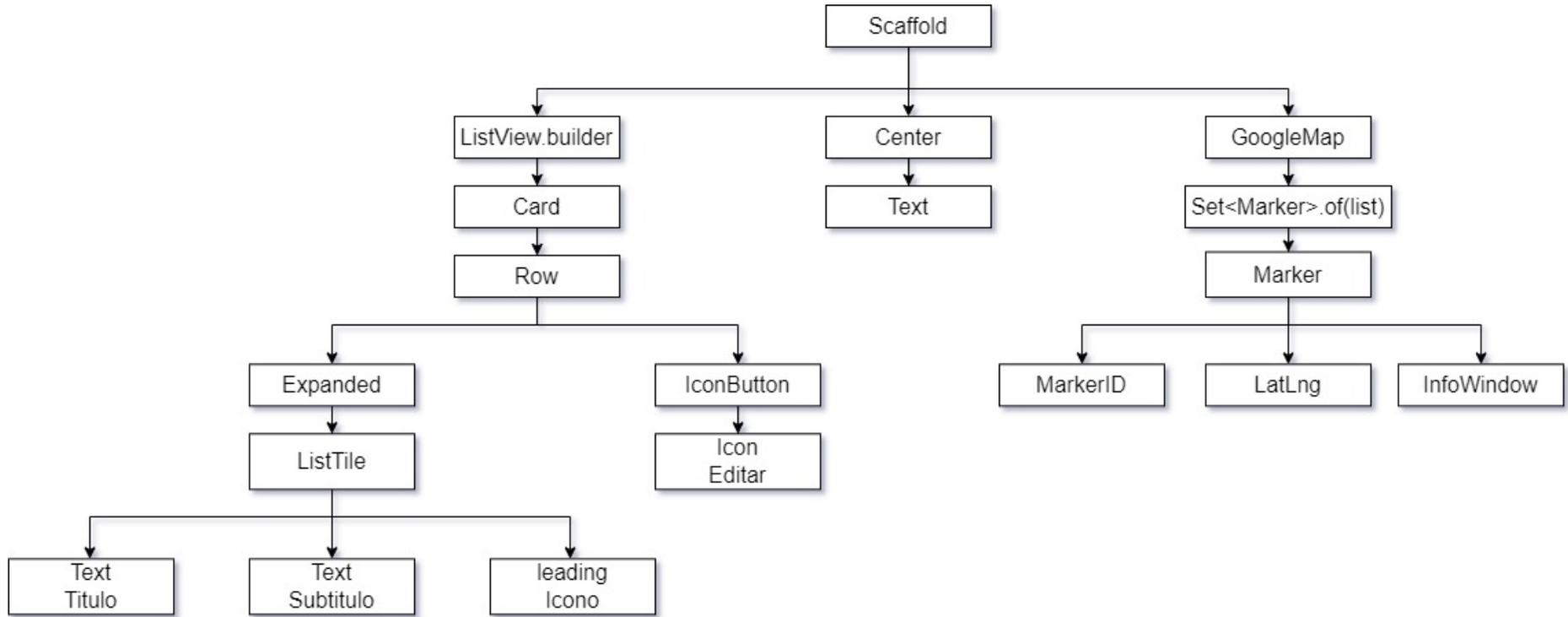


Figura 2.25: Diagrama de clases utilizadas. [Fuente: El Autor.]

2.5.3. Diagrama de clases

En la figura 2.26, se representa el diagrama de clases que se utilizó para la creación de la plataforma de monitoreo, a continuación se describen las clases y cual es su objetivo y a que clases las llaman.

1. Main:

- **Objetivo:** Es el punto de entrada de la aplicación. Crea una instancia de "MyApp" y configura un MaterialApp con un título y un tema, luego define "MyHomePage" como la pantalla de inicio de la aplicación.
- **Llama a clase:** "Home".

2. Home:

- **Objetivo:** Implementa la pantalla principal de la aplicación. Muestra un mapa con ubicaciones de sensores y una lista de sensores disponibles. El usuario puede alternar entre ver el mapa y la lista utilizando un botón de acción, y también puede acceder a la pantalla de inicio de sesión para gestionar su sesión.
- **Llama a clase:** "Login", " Edit" , "Graphic" .

3. Login:

- **Objetivo:** Representa la pantalla de inicio de sesión de la aplicación. Permite a los usuarios iniciar sesión o cerrar sesión utilizando sus credenciales. También muestra un mensaje de bienvenida cuando el usuario ha iniciado sesión con éxito.
- **Llama a clase:** Ninguna clase.

4. Edit:

- **Objetivo:** Implementa una pantalla de edición que permite al usuario editar la latitud y longitud de un sensor específico, así como visualizar su nombre y descripción. También permite obtener la ubicación actual del dispositivo para establecer valores predeterminados para la latitud y longitud.

- Llama a clase: Ninguna clase.

5. Graphic:

- Objetivo: Muestra dos gráficos radiales con información en tiempo real de los sensores "MH-Z19Bz "MG811", mostrando la concentración de CO₂ en partes por millón. Además, permite al usuario ver más detalles en forma de estadísticas y registros, según la disponibilidad de datos del usuario actual.
- Llama a clase: Records", "Statistics".

6. Statistics:

- Objetivo: Representa la pantalla que muestra las estadísticas de un sensor específico en forma de gráfico. Se actualiza automáticamente cuando los datos de estadísticas están disponibles. El gráfico muestra las lecturas de CO₂ a lo largo del tiempo, agrupadas por una variable llamada "group".
- Llama a clase: Ninguna clase.

7. Records:

- Objetivo: Representa la pantalla que muestra los registros de un sensor específico en forma de lista. Se actualiza automáticamente cuando los datos están disponibles.
- Llama a clase: Ninguna clase.

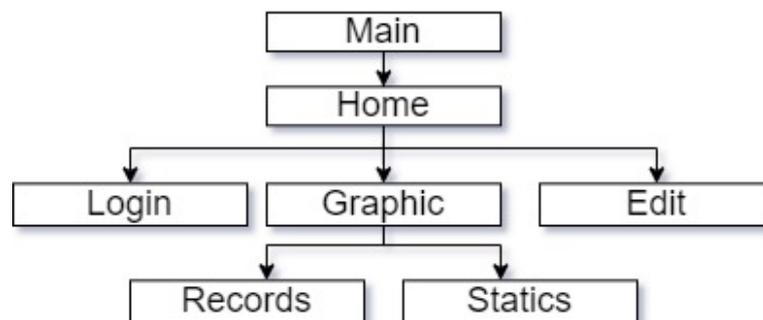


Figura 2.26: Diagrama de clases utilizadas. [Fuente: El Autor.]

2.6. Base de datos PostgreSQL

El servidor Chirpstack utiliza una base de datos relacional PostgreSQL para almacenar y gestionar la información con respecto a la red, las aplicaciones y los dispositivos que utiliza la plataforma.

En la figura 2.27, se observa el diagrama final de la base de datos, donde observamos la tabla de “sensor” depende de “device”, donde estarán siendo almacenados los valores medidos de los sensores de CO₂, mediante el servicio creado en el back-end.

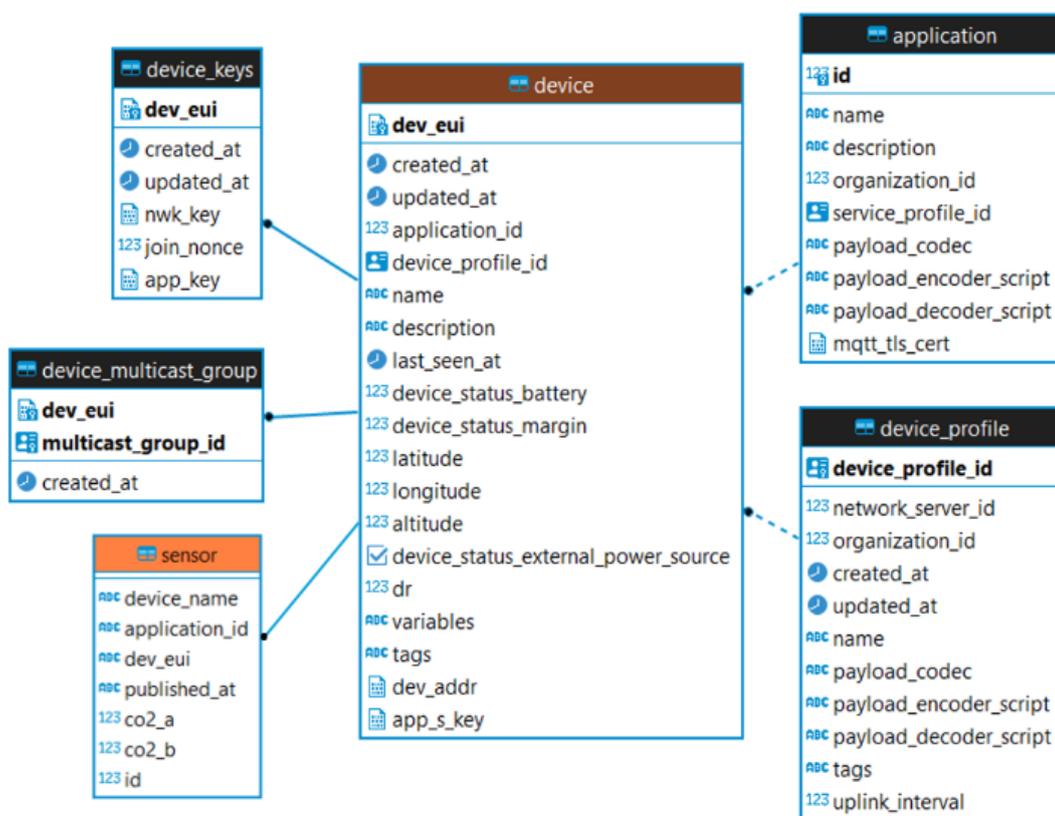


Figura 2.27: Diagrama de la base de datos PostgreSQL. [Fuente: El Autor.]

2.7. Nombre y icono de la aplicación de monitoreo

El icono de la aplicación de monitoreo, como se observa en la figura 2.28, se debe colocar la figura en la carpeta “assets”, donde el archivo **pubspec.yaml**, utiliza la propiedad para especificar la ruta del archivo de icono de la siguiente manera:

```
flutter_icons:  
  android: "launcher_icon"  
  ios: true  
  image_path: "assets/icon/iconGitel.jpg"
```



Figura 2.28: Icono de aplicación de monitoreo. [Fuente: El Autor.]

Además, el nombre de la aplicación va a ser **"Gitel CO₂"** y para cambiar el nombre en la aplicación ingresamos al archivo **AndroidManifest.xml**, y en el apartado "android" cambiamos el nombre de la siguiente manera:

```
android:label="Gitel CO2"
```

Capítulo 3

Implementación y Pruebas de Operación de la Plataforma de Monitoreo

3.1. Pruebas de Operación: Nodos de CO₂

3.1.1. Comunicación Chirpstack

En la figura 3.1, observamos el dashboard del servidor Chirpstack, donde podemos encontrar los Gateways activos, donde observamos que una porción del gráfico se encuentra en color verde significa que el gateway está en funcionamiento.

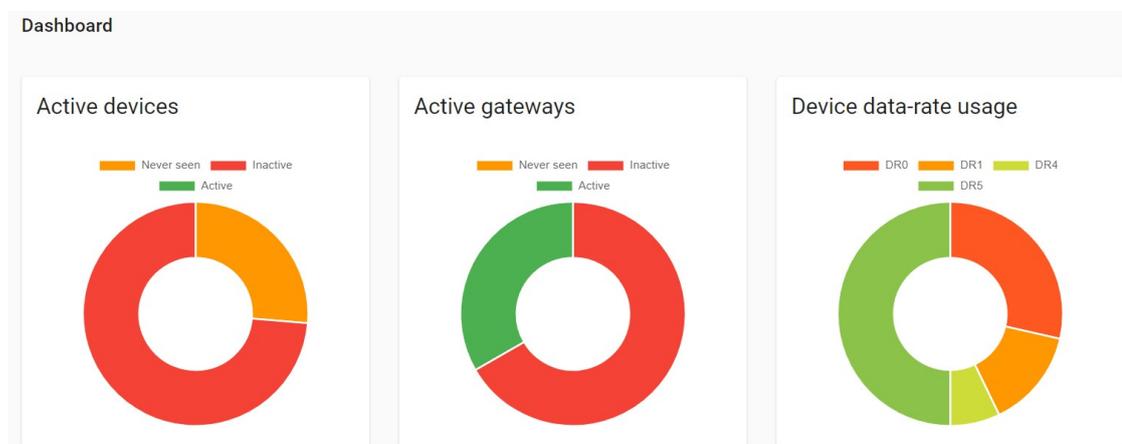


Figura 3.1: Dashboard Chirpstack

Cuando se hayan seguido los pasos correctamente del capítulo 2 para configurar los nodos de CO₂ en el servidor Chirpstack, en la figura 3.2 se observa el apartado de aplicaciones que se creó para el servicio de nodos de CO₂ se encuentran conectados y enviando tramas de datos.

Last seen	Device name	Device EUI	Device profile	Link margin	Battery
<input type="checkbox"/> a few seconds ago	Nodo1	8 2ddc8f 00fd82	CO2_DP	n/a	n/a
<input type="checkbox"/> a few seconds ago	Nodo2	5 2306e6 17b976	CO2_DP	n/a	n/a
<input type="checkbox"/> a few seconds ago	Nodo3	1 b98fef 19653f	CO2_DP	n/a	n/a

Figura 3.2: Características de Nodos. [Fuente: El Autor.]

3.1.2. Colocación de nodos

Inicialmente se planteó colocar los nodo como se muestran en la figura 3.3, también se colocaron en las posiciones indicadas en la figura 3.4, las ubicaciones se muestran en la tabla 3.1:

Tabla 3.1: Ubicaciones de Nodos y Gateway

NODO	LATITUD	LONGITUD
GATEWAY	2°53'12.21"S	78°59'26.57"W
NODO 1	2°53'32.20"S	79° 0'21.20"W
NODO 2	2°53'12.29"S	78°59'21.11"W
NODO 3	2°53'32.20"S	78°59'54.50"W

Se tomo en cuenta para ubicar a los nodos de CO₂ las siguientes consideraciones:

Ubicación Un punto donde haya gran concurrencia de transporte tanto publico como privado.

Concentración Se tomo en cuenta donde hayan un mayor numero de concentración de personas.



Figura 3.3: Ubicaciones Iniciales de Nodos. [Fuente: El Autor.]

En la figura se muestra los nodos colocados en las ubicaciones correspondientes:



Figura 3.4: Ubicaciones Iniciales de Nodos. [Fuente: El Autor.]

3.1.3. Recepción de datos en ChirpStack

En ChirpStack, podemos observar las tramas de datos enviadas por cada nodo, para esto debemos ingresar a la aplicación luego al nodo y en la ventana "Device Data." observaremos lo datos recibidos como se observa en la figura 3.5.

The screenshot shows the ChirpStack web interface for a device named 'Nodo3'. The 'DEVICE DATA' tab is active, displaying a list of received frames. Each frame includes a timestamp, status, frequency, SF, BW, FCnt, FPort, and a 'Confirmed' status. The interface also includes navigation tabs, a 'DELETE' button, and control buttons for HELP, PAUSE, DOWNLOAD, and CLEAR.

Timestamp	Status	Frequency	SF	BW	FCnt	FPort	Confirmed
Jul 24 10:47:14 PM	up	918.2 MHz	SF12	BW125	FCnt: 11303	FPort: 2	Confirmed
Jul 24 10:46:07 PM	up	917.8 MHz	SF12	BW125	FCnt: 11302	FPort: 2	Confirmed
Jul 24 10:45:13 PM	up	917 MHz	SF12	BW125	FCnt: 11301	FPort: 2	Confirmed
Jul 24 10:44:05 PM	up	917.4 MHz	SF12	BW125	FCnt: 11300	FPort: 2	Confirmed
Jul 24 10:43:05 PM	up	918 MHz	SF12	BW125	FCnt: 11299	FPort: 2	Confirmed
Jul 24 10:42:04 PM	up	917 MHz	SF12	BW125	FCnt: 11298	FPort: 2	Confirmed
Jul 24 10:41:11 PM	up	917.4 MHz	SF12	BW125	FCnt: 11297	FPort: 2	Confirmed
Jul 24 10:40:04 PM	up	917.8 MHz	SF12	BW125	FCnt: 11296	FPort: 2	Confirmed

Figura 3.5: Tramas de datos recibidas. [Fuente: El Autor.]

3.2. Pruebas de operación de servidor

Cuando se haya instalado el servicio vamos a comprobar si se ejecutaron correctamente, en la figura 3.6, observamos que las tramas de datos se pueden leer y almacenar en un archivo de texto.

Para poder revisar el estado del servicio " daemon sensor.service " ingresamos al servidor y colocamos el siguiente código para la consulta:

```
sudo systemctl status sensor
```

3.3. PRUEBAS DE OPERACIÓN DE APLICACIÓN MÓVIL: MODO USUARIO67

```
danielpaezc@monitoreo-1:~$ sudo su
root@monitoreo-1:/home/danielpaezc# sudo systemctl status sensor
● sensor.service - Sensor CO2
   Loaded: loaded (/etc/systemd/system/sensor.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-07-15 16:10:04 UTC; 1 weeks 2 days ago
     Main PID: 286128 (init.sh)
       Tasks: 4 (limit: 4693)
      Memory: 5.2M
         CPU: 15min 18.134s
       CGroup: /system.slice/sensor.service
              └─286128 /bin/bash /usr/local/bin/servidor/init.sh
                 └─286129 curl -X GET --header Accept: application/json --header Grpc-Metadata-Authorization: Bear
                    └─286130 curl -X GET --header Accept: application/json --header Grpc-Metadata-Authorization: Bear
                       └─286131 curl -X GET --header Accept: application/json --header Grpc-Metadata-Authorization: Bear

Jul 25 03:50:42 monitoreo-1 init.sh[286130]: [79B blob data]
Jul 25 03:50:43 monitoreo-1 init.sh[286131]: [79B blob data]
Jul 25 03:50:43 monitoreo-1 init.sh[286129]: [79B blob data]
Jul 25 03:50:43 monitoreo-1 init.sh[286130]: [79B blob data]
Jul 25 03:50:44 monitoreo-1 init.sh[286131]: [79B blob data]
Jul 25 03:50:44 monitoreo-1 init.sh[286129]: [79B blob data]
Jul 25 03:50:44 monitoreo-1 init.sh[286130]: [79B blob data]
Jul 25 03:50:45 monitoreo-1 init.sh[286131]: [79B blob data]
Jul 25 03:50:45 monitoreo-1 init.sh[286129]: [79B blob data]
Jul 25 03:50:45 monitoreo-1 init.sh[286130]: [79B blob data]
lines 1-23/23 (END)
```

Figura 3.6: Estado de servicio para almacenar datos en archivo de texto. [Fuente: El Autor.]

También debemos revisar que las tramas de datos se están almacenando en la base de datos de ChirpStack, por lo que en el servicio se revisa el estado del servicio “daemon sensorpy.service” como se muestra en la figura 3.7.

```
sudo systemctl status sensorpy
```

```
root@monitoreo-1:/home/danielpaezc# sudo systemctl status sensorpy
● sensorpy.service - Sensor python CO2
   Loaded: loaded (/etc/systemd/system/sensorpy.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-06-22 19:09:24 UTC; 1 months 1 days ago
     Main PID: 8917 (python3.9)
       Tasks: 1 (limit: 4693)
      Memory: 35.4M
         CPU: 7h 26min 42.402s
       CGroup: /system.slice/sensorpy.service
              └─8917 /usr/bin/python3.9 /usr/local/bin/servidor/sensor.py

Warning: journal has been rotated since unit was started, output may be incomplete.
```

Figura 3.7: Estado de servicio para almacenar datos en base de datos. [Fuente: El Autor.]

Donde observamos que los servicios se encuentran activos y funcionando correctamente, para poder ser utilizados en los siguientes procesos.

3.3. Pruebas de operación de aplicación móvil: Modo Usuario

En el modo Usuario solo se puede observar en el mapa las ubicaciones de los sensores de CO₂ una lista con todos los sensores de la red y los valores

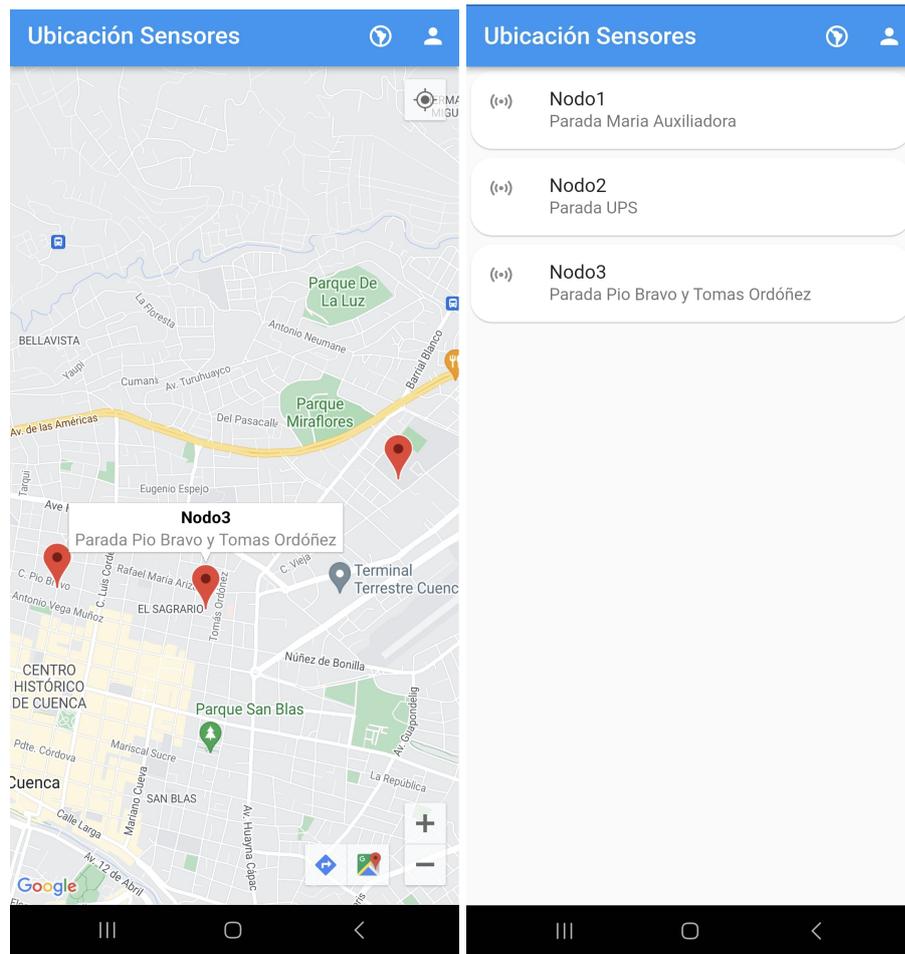
de CO₂ de los sensores.

3.3.1. Ubicación de sensores

En la figura 3.8 se observa las pantallas iniciales que pueden escoger los Usuarios, en donde se podrán cambiar entre dos pantallas se que describen a continuación:

Pantalla Ubicación de Sensores en el mapa En la figura 3.8(a) observamos la ubicaciones de los sensores en el mapa, los pines de color rojo es un sensor, cuando se presiona en el pin se despliega el nombre y descripción del sensor y para acceder a los valores se debe presionar en el nombre donde direccionara hacia la pantalla de valores de CO₂.

Pantalla Ubicación de Sensores en lista Además en la figura 3.8(b), el usuario podrá observar todos los sensores en una lista con el nombre y la descripción de cada nodo de CO₂ para acceder a los valores se deberá presionar en el sensor que se desea observar.



((a)) Pantalla Inicial Usuarios

((b)) Pantalla Lista de Sensores

Figura 3.8: Pantallas de Inicio de usuarios. [Fuente: El Autor.]

3.3.2. Visualización de valores de CO₂

Cuando se haya seleccionado un nodo ingresará hacia los valores de CO₂ donde se podrán observar los valores de manera gráfica y lineal, cabe recalcar que se podrán observar datos en tiempo real y también un historial de 24 horas. En la figura 3.9, se observa las maneras en las que se puede visualizar los valores de CO₂ de los sensores colocados en cada nodo descritos a continuación:

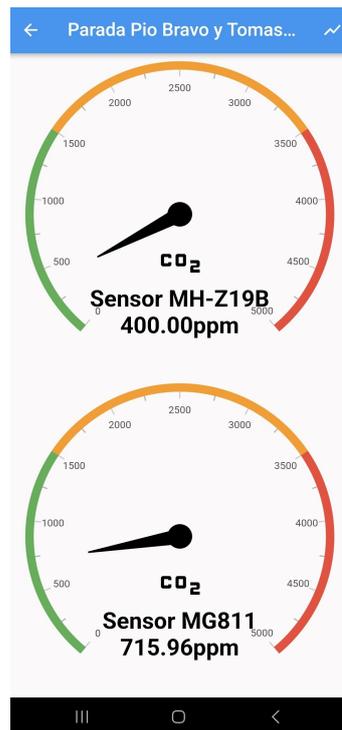
Valores de CO₂ gráfico: En la figura 3.9(a), se observa el valor en tiempo real de los sensores de CO₂ colocados en cada nodo, estos valores van desde 0 a 500ppm. El gráfico circular contiene tres rangos de colores verde, amarillo y rojo, que se indican como alertas de nivel de CO₂ bajo, medio y alto.

Valores de CO₂ lineal: La figura 3.9(b) muestra valores en tiempo real y

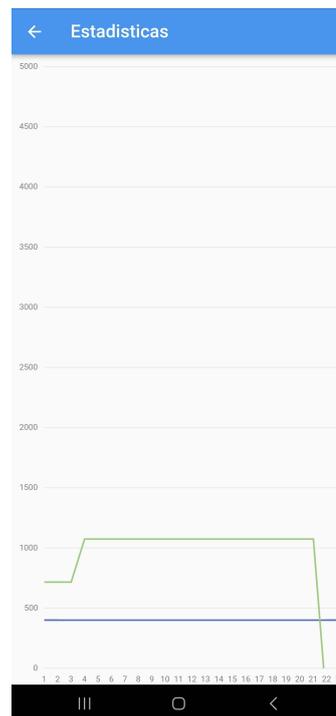
3.3. PRUEBAS DE OPERACIÓN DE APLICACIÓN MÓVIL: MODO USUARIO70

además los valores de las últimas 24 horas que el sensor monitorizó. En la figura se van a observar dos valores lineales de los dos sensores, donde la línea de color azul es el sensor MH-Z19B y la de color verde es el sensor MG811.

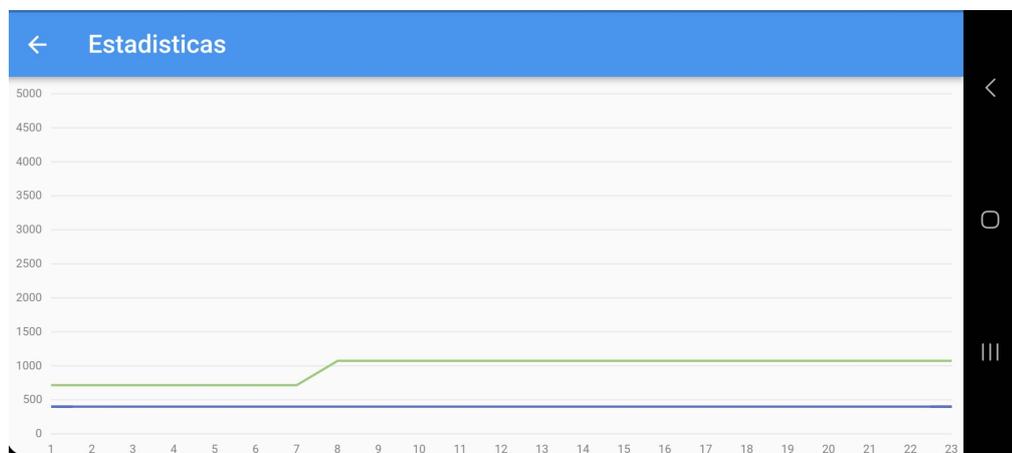
Para una mejor visualización de los valores de CO₂ se recomienda girar el celular como se observa en la figura 3.9(c).



((a)) Pantalla Valor de Sensores Gráfico



((b)) Pantalla 1 Valor de Sensores Lineal



((c)) Pantalla 2 Valor de Sensores Lineal

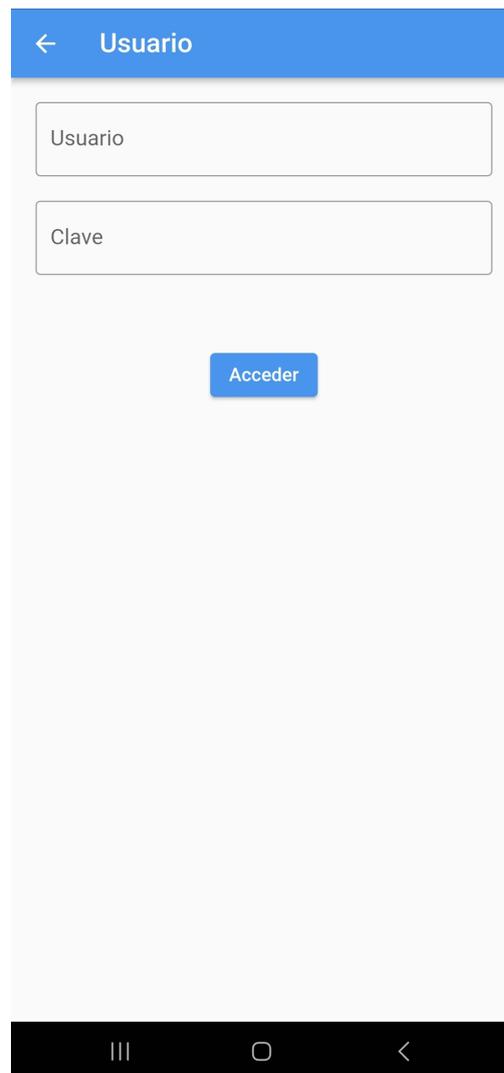
Figura 3.9: Pantallas de Valores de Sensores. [Fuente: El Autor.]

3.4. Pruebas de operación de aplicación móvil: Modo Administrador

El modo administrador se utiliza para la edición de ubicaciones de los sensores de CO₂, además el usuario permitirá observar las tramas recibidas en el servidor ChirpStack.

3.4.1. Inicio de sesión

Para entrar en modo de administrador se debe ingresar mediante un usuario y contraseña preestablecidos como se muestra en la figura 3.10, las credenciales de usuario deberán ser entregadas por el administrador del servicio.



El prototipo de pantalla de inicio de sesión del administrador muestra una interfaz de usuario con un encabezado azul que contiene un ícono de retroceso y el título 'Usuario'. Debajo del encabezado, hay dos campos de entrada de texto: el primero está etiquetado como 'Usuario' y el segundo como 'Clave'. Debajo de los campos de entrada, hay un botón azul con el texto 'Acceder'. La pantalla tiene un fondo gris claro y una barra de navegación negra en la parte inferior con los íconos de menú, inicio y retroceso.

Figura 3.10: Prototipo de pantalla de inicio de sesión del administrador. [Fuente: El Autor.]

3.4.2. Cambio de ubicación de nodos de CO₂

Inicialmente la ubicación de los nodos de CO₂ vendrán por defecto, para poder modificarlos tendremos que entrar en la lista de "Ubicación de Sensores", y seleccionar en el símbolo de edición, en la figura 3.11 se podrán cambiar las ubicaciones mediante dos opciones:

- **Edición por ubicación en tiempo real:** En la figura 3.11(a) se observa símbolo de "Mi locación" en la parte superior derecha, cuando se presione se obtendrá la latitud y longitud en tiempo real.
- **Edición manual:** Para la edición se tendrá que ingresar la latitud y longitud manualmente en los campos como se observa en la figura 3.11(b).

Luego de haber ingresado la latitud y longitud se deberá presionar el botón **Editar** y se guardaran los cambios realizados.

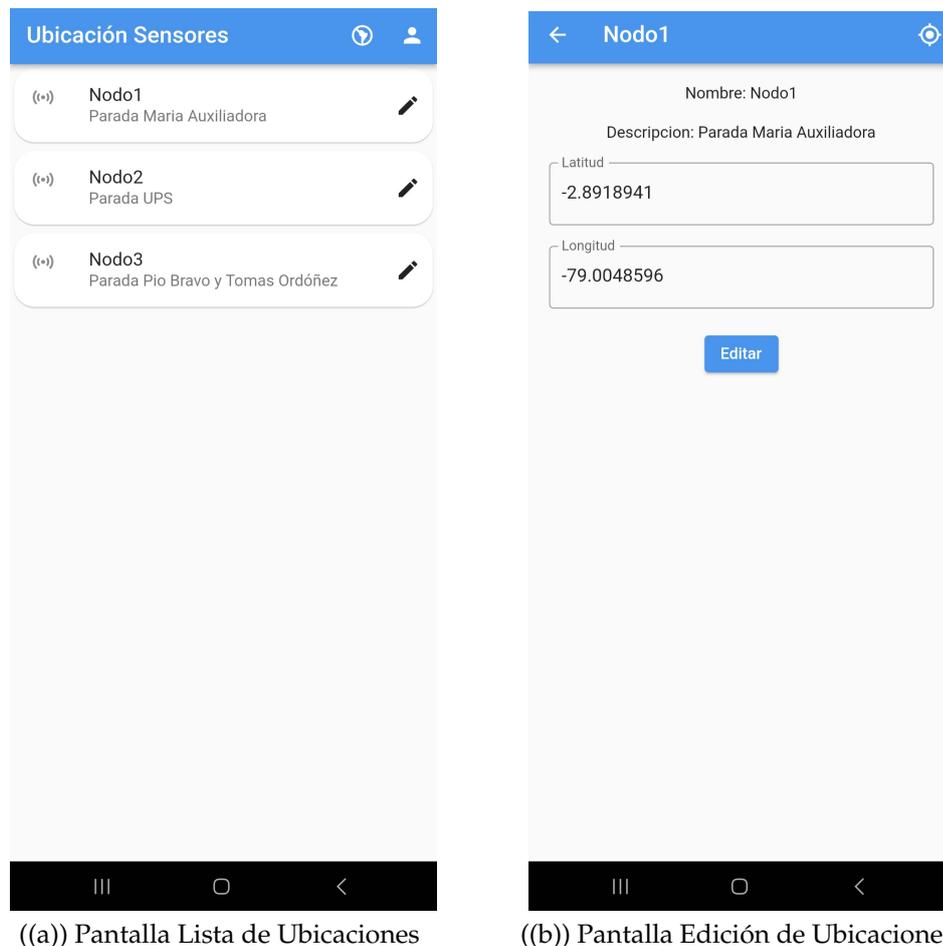
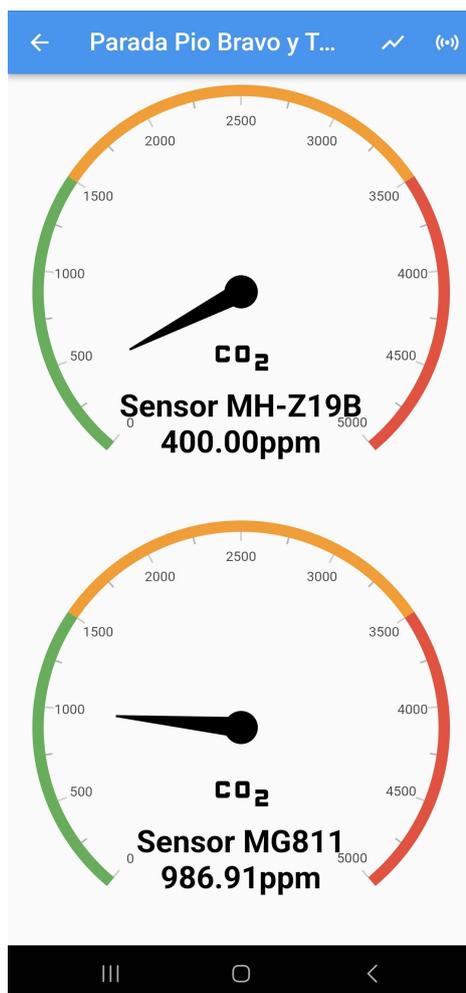


Figura 3.11: Edición de Ubicaciones de sensores de CO₂ [Fuente: El Autor.]

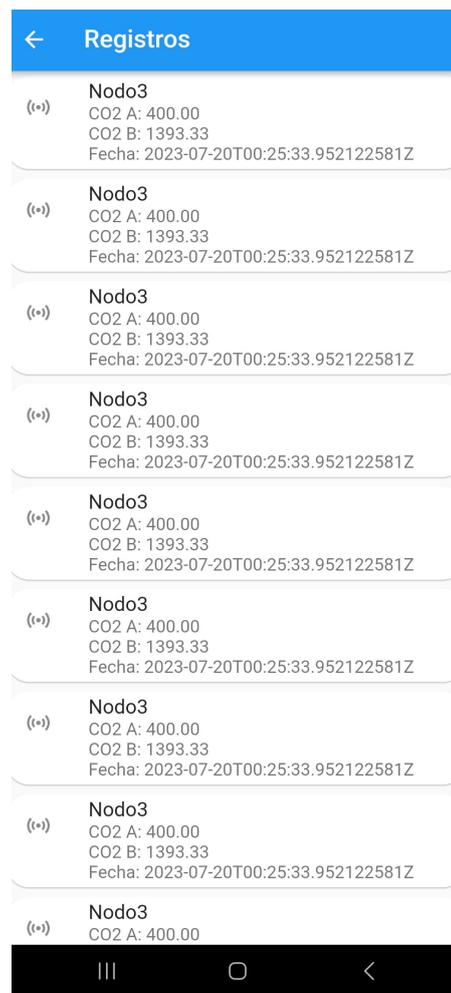
3.4.3. Datos en el servidor

Según se muestra en la figura 3.12, el acceso a los registros que se enviaron al servidor ChirpStack requiere seguir un conjunto de pasos específicos:

1. Desde la pantalla de valores de sensores, es necesario presionar el icono de sensores ubicado en la parte superior derecha, tal como se muestra en la figura 3.12(a).
2. Una vez que se ha presionado el icono de sensores, se podrán visualizar los registros de las tramas de datos que fueron enviados al servidor ChirpStack. Estos registros pueden ser observados en la figura 3.12(b).



((a)) Pantalla Valores de Sensores



((b)) Pantalla Registros de ChirpStack

Figura 3.12: Ingreso a registros de ChirpStack. [Fuente: El Autor.]

3.5. Análisis de Resultados

Los nodos de CO₂ estuvieron recolectando datos durante 4 semanas, en la figura 3.15 observamos los resultados obtenidos de los sensores MG-811 y MHZ-19B que fueron colocados en la ubicación Pío Bravo y Tomas Ordoñez.

Los resultados muestran lo siguiente:

- Se comprobó la funcionalidad de los dos sensores
- La información proveniente del Nodo 3 consistió en 1,611,093 datos, de los cuales 1,234,348 fueron recibidos y 376,745 no fueron recibidos exitosamente. La representación visual de estas cifras se muestra en la figura 3.13 mediante un gráfico de pastel, indicando que el 76.82 % de los paquetes fueron recibidos con éxito, mientras que el 23.2 % de los paquetes se perdieron.

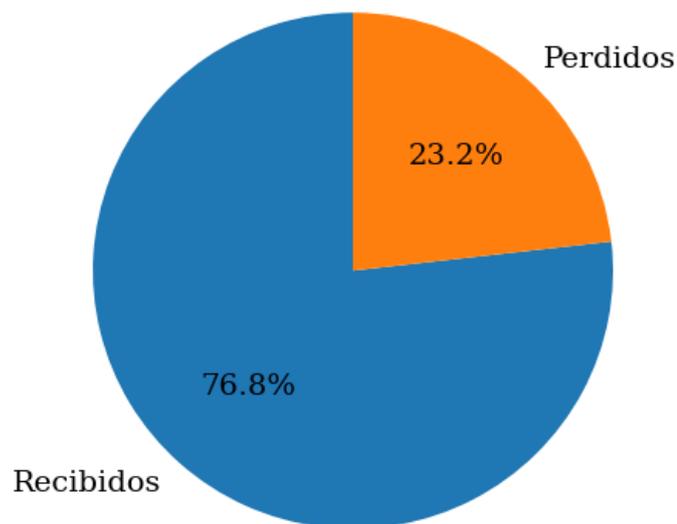


Figura 3.13: Comparación de porcentaje de paquetes perdidos de sensores. [Fuente: El Autor.]

- En la figura 3.14, se presenta un cuadro comparativo que muestra las pérdidas de paquetes para cada día de la semana. Se puede observar que el día con la menor cantidad de pérdidas fue el lunes, con solo un 12.26 % de paquetes perdidos, mientras que el día con las pérdidas más significativas fue el sábado, alcanzando un alarmante 98.49 %.

Estas pérdidas pueden ser atribuidas a diversos factores, tales como la des-calibración de sensores, el mal funcionamiento del GATEWAY sin

servicio de internet o incluso factores climáticos adversos. Cada uno de estos elementos podría contribuir a la pérdida de paquetes y afectar la calidad y fiabilidad de la transmisión de datos.

Es esencial investigar y abordar estos problemas para mejorar la confiabilidad del sistema y garantizar una transmisión de datos más estable y efectiva, especialmente en días críticos con altos índices de pérdidas como el sábado.

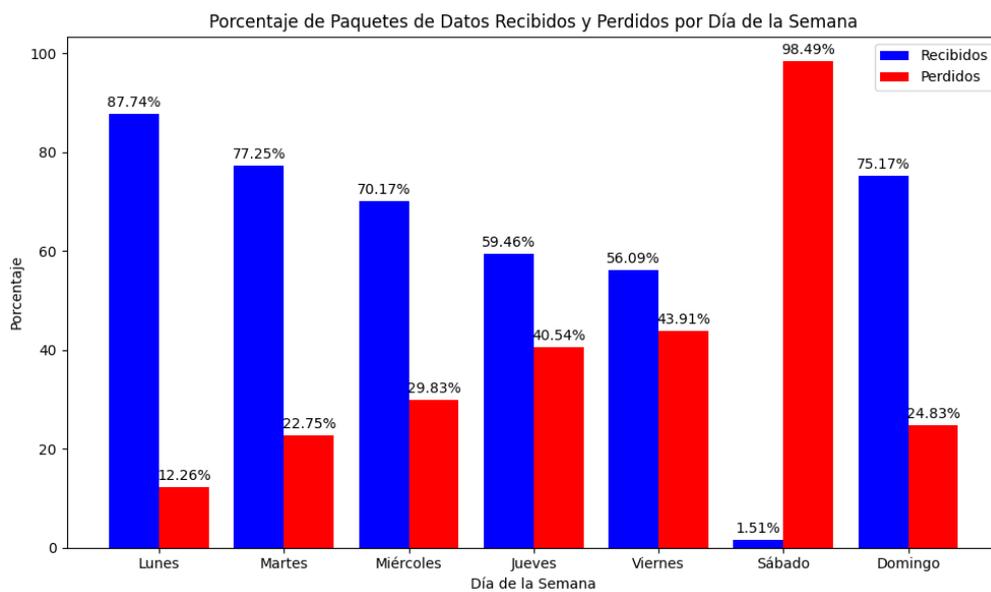


Figura 3.14: Comparación de porcentaje de datos recibidos y perdidos por día de la semana. [Fuente: El Autor.]

- En la figura 3.15, se muestran los datos recibidos de los dos sensores de CO₂ MG-811 y MHZ-19B, durante aproximadamente 4 semanas. Esta representación visual permite observar el comportamiento y rendimiento de cada uno de estos sensores. En específico, se detallan los niveles de variación de los sensores en relación con los niveles de CO₂.

Al analizar la figura, se puede notar que el sensor MG-811 demostró ser más sensible a los cambios de CO₂, presentando picos más altos de partes por millón (PPM). Por otro lado, el sensor MHZ-19B se mantuvo más estable, sin experimentar muchos cambios significativos.

Estos resultados sugieren que el sensor MHZ-19B muestra un mejor funcionamiento en entornos interiores, mientras que el sensor MG-811

puede ser más adecuado para entornos exteriores o situaciones donde es necesario detectar rápidamente cambios significativos en los niveles de CO₂.

Estas observaciones son valiosas para comprender las características y limitaciones de cada sensor, lo que puede ser útil al seleccionar el sensor más adecuado para distintos propósitos y entornos de monitoreo de CO₂.

Además, esto destaca la importancia de utilizar el sensor adecuado según el contexto y los requisitos específicos de la aplicación.

En la primera gráfica de la figura 3.15, se observa la calibración de los sensores con respecto al CO₂ del medio ambiente, se realizaron varias pruebas en diferentes escenarios para la calibración y adquisición de valores.

En la tabla 3.2 de análisis de resultados, se resumen los resultados presentados en la figura anterior 3.15, donde se derivan los siguientes hallazgos a partir de las pruebas efectuadas:



Figura 3.15: Comparación de sensores.[Fuente: El Autor.]

Tabla 3.2: Analisis de resultados de datos

Fecha de inicio	MG-811	MHZ-19B
2023-06-26	- Calibración del sensor - Tiempo de precalentamiento de 48 horas	- No se expone directamente a gas CO ₂ - Se realiza pruebas en ambiente interior - No varía considerablemente debido al nivel de exposición
2023-07-03	- No se realiza pruebas	- Exposición a ambientes internos y externos - Exposición a diversos ambientes de pruebas - Cuando está expuesto en ambientes internos se obtienen picos altos mayores a 5000ppm. - Cuando está expuesto en ambiente externos se observa picos no superiores a 4000ppm. - Se comprueba funcionamiento, tiene un mejor rendimiento en ambientes interiores.
2023-07-17	- Pruebas en ambientes externos - Es mucho más sensible a la variación de CO ₂ .	- En ambientes externos depende de la ubicación del sensor con respecto a la exposición con el gas CO ₂ . - Permanece estable a 400ppm
2023-07-24	- Se demuestra el funcionamiento del sensor, es preferible utilizarlo en ambientes externos donde su perceptibilidad mejora.	- Permanece estable a 400ppm

3.6. Análisis de Costos

A continuación se presentan cuadros de costos para los diferentes casos de que se desea el nodo de CO₂

1. Nodo con ambos sensores (MG811 y MHZ-19B):

- En la tabla 3.3 se detalla el costo total para este nodo es de 260.875 USD.
- Este es el tipo de nodo más costoso, ya que requiere la adquisición de ambos sensores, MG811 y MHZ-19B.
- La ventaja de este nodo es que ofrece la posibilidad de utilizar ambos sensores, lo que permite adaptarse a diversas situaciones y aplicaciones, cubriendo diferentes niveles de sensibilidad y estabilidad en la detección de CO₂.

Tabla 3.3: Presupuesto de nodos con dos sensores

Cantidad	Descripción de Item	Valor Unitario	Valor total
1	Módulo HTCC-AB 01	20	20
1	Sensor MG811	70	70
1	Sensor MHZ-19B	26	26
1	Placa PCB	39.5	39.5
1	Modulo gestión de energía	17.9	17.9
1	Conector peineta pcb	1.4	1.4
2	Conector molex 2p	0.65	1.3
1	Conector molex 3p	0.75	0.75
1	Conector molex 4p	0.95	0.95
1	Interruptor 2 pines	0.75	0.75
1	Cable para sensor	2.5	2.5
1	Mini Panel solar 6v 200mA	6.5	6.5
1	Bateria recargable 3.7V 2500mAH	16.875	16.875
1	Caja impresa + componentes estructurales	52	52
1	Antena	4.45	4.45
		TOTAL	260.875

2. Nodo con sensor MHZ-19B

- En la tabla 3.4 se observa el costo total para este nodo es de 190.875 USD.
- El sensor MHZ-19B es el componente más costoso de este nodo, con un valor unitario de 26 USD. Al igual que en el caso anterior, otros componentes también contribuyen al costo total.
- Este nodo es más económico que el anterior y puede ser una opción adecuada cuando se prioriza un costo más bajo y se puede tolerar una menor sensibilidad a los cambios de CO₂, pero mayor estabilidad en ambientes interiores.

3. Nodo con sensor MG811

- En la figura 3.5 el costo total para este nodo es de 234.875 USD.
- El sensor MG811 es el componente más costoso, con un valor unitario de 70 USD. Los otros componentes, como el módulo HTCC-AB 01, la placa PCB, el módulo de gestión de energía, entre otros, también contribuyen al costo total del nodo.

Tabla 3.4: Presupuesto de nodos con sensor MHZ-19B

Cantidad	Descripción de Item	Valor Unitario	Valor total
1	Módulo HTCC-AB 01	20	20
1	Sensor MHZ-19B	26	26
1	Placa PCB	39.5	39.5
1	Modulo gestión de energía	17.9	17.9
1	Conector peineta pcb	1.4	1.4
2	Conector molex 2p	0.65	1.3
1	Conector molex 3p	0.75	0.75
1	Conector molex 4p	0.95	0.95
1	Interruptor 2 pines	0.75	0.75
1	Cable para sensor	2.5	2.5
1	Mini Panel solar 6v 200mA	6.5	6.5
1	Bateria recargable 3.7V 2500mAH	16.875	16.875
1	Caja impresa + componentes estructurales	52	52
1	Antena	4.45	4.45
		TOTAL	190.875

- Este tipo de nodo puede ser una opción preferible cuando se requiere una mayor sensibilidad a los cambios de CO₂ en ambientes exteriores y se puede justificar el costo adicional.

Tabla 3.5: Presupuesto de nodos con sensor MG811

Cantidad	Descripción de Item	Valor Unitario	Valor total
1	Módulo HTCC-AB 01	20	20
1	Sensor MG811	70	70
1	Placa PCB	39.5	39.5
1	Modulo gestión de energía	17.9	17.9
1	Conector peineta pcb	1.4	1.4
2	Conector molex 2p	0.65	1.3
1	Conector molex 3p	0.75	0.75
1	Conector molex 4p	0.95	0.95
1	Interruptor 2 pines	0.75	0.75
1	Cable para sensor	2.5	2.5
1	Mini Panel solar 6v 200mA	6.5	6.5
1	Bateria recargable 3.7V 2500mAH	16.875	16.875
1	Caja impresa + componentes estructurales	52	52
1	Antena	4.45	4.45
		TOTAL	234.875

En la tabla 3.6 podemos observar que el costo más bajo es para el nodo con sensor MHZ-19B (190.875 USD) y el costo más alto es para el nodo con ambos sensores MG811 y MHZ-19B (260.875 USD). El nodo con el sensor MG811 se encuentra en un punto intermedio en términos de costo.

Es importante destacar que estos costos están basados en los componentes específicos mencionados en las tablas proporcionadas previamente. Otros factores como el mantenimiento, la vida útil de los sensores y la infraestructura necesaria también deben ser considerados al tomar una decisión final.

En función de las necesidades y el presupuesto del proyecto, se puede seleccionar el tipo de nodo que mejor se ajuste a los requerimientos específicos. Si se busca una mayor sensibilidad y adaptabilidad en la detección de CO₂ el nodo con ambos sensores puede ser la mejor opción, aunque también es el más costoso. Por otro lado, si se prioriza un costo más bajo y se puede tolerar una menor sensibilidad a los cambios de CO₂ el nodo con el sensor MHZ-19B podría ser más adecuado.

Tabla 3.6: Resumen costos totales

Cantidad	Descripción de Item	Valor Unitario	Valor total
3	Nodos de CO ₂ + 2 Sensores	260.875	782.625
3	Nodos de CO ₂ + 1 sensor MHZ-19B	190.875	572.625
3	Nodos de CO ₂ + 1 sensor G-811	234.875	704.625

Capítulo 4

Conclusiones y Trabajos Futuros

4.1. Conclusiones

En este trabajo, luego de haber realizado la revisión del estado del arte, se reconoce la creciente importancia del Internet de las Cosas (IoT) en el monitoreo y gestión de la calidad del aire. Así, la capacidad de adquisición y transmisión de datos en tiempo real desde nodos de CO₂ distribuidos en diferentes locaciones, podría generar un cambio significativo en la manera que se aborda y comprende la calidad del aire en ambientes tanto urbanos como rurales. Por tanto, de acuerdo a los trabajos revisados, permite una mejor comprensión de la calidad del aire por medio de un análisis más preciso de las tendencias y patrones de contaminación, lo cual, resulta fundamental para la toma de decisiones informadas.

Un aspecto fundamental de la investigación ha sido la evaluación de diferentes tipos de sensores de CO₂ para aplicaciones del monitoreo. Tras el análisis podemos deducir la importancia de considerar la precisión, durabilidad y el costo para seleccionar los sensores para proyectos específicos. La selección adecuada de sensores asegura que las mediciones sean confiables y los resultados precisos en el monitoreo de la calidad del aire. En este sentido, se puede decir que el sensor MG811 tiene mejores resultados en ambientes exteriores, además, requiere calibración y mantenimiento cada cierto tiempo para obtener mediciones precisas. Mientras que, el sensor MH-Z19B sus mediciones son más precisas y fiables en ambientes internos, no necesita

calibración manual y es más estable y de fácil interpretación de la salida de valores de CO₂.

Una plataforma de monitoreo basada en IoT puede ser implementada en cualquier sistema de almacenamiento en la nube, el de este trabajo, opera con un servicio en la nube para la gestión de LoRaWAN a través de un servidor ChirpStack que está alojado en AWS. En general, la mayoría de compañías ofrecen este servicio de computación en la nube por un año de uso gratis, sin embargo, después de este periodo de tiempo, la prueba gratuita termina teniendo un costo mensual que ronda los \$10 al mes, valor que puede no ser alto si se mantienen varios entornos LoRaWAN haciendo que una plataforma de este tipo pueda ser auto sustentable en un negocio de IoT. No obstante, si la cantidad de datos que se registran se eleva exponencialmente, el costo de alojamiento también sube. En este punto, es necesario evaluar si el costo de alojamiento es rentable vs la implementación de Chipstack en un alojamiento local con gestión completa por parte del administrador de la plataforma tomando en cuenta que, en este caso habría que pagar por el acceso a una IP pública para que los servicios de LoRaWAN puedan ser accesibles desde cualquier lugar del mundo.

La interacción de la aplicación móvil con el servidor de LoRaWAN, permite la conexión de dispositivos IoT a larga distancia y bajo consumo energético. Esto se presenta como una alternativa especialmente en aplicaciones para ciudades inteligentes, como el monitoreo ambiental o la agricultura inteligente, entre otras. La combinación de tecnologías, interfaces intuitivas y herramientas de análisis de datos proporciona una experiencia completa para los usuarios, lo que es indispensable para la adopción y el impacto de la plataforma. Además, el análisis de costos presentado en las tablas permite tomar decisiones informadas al seleccionar el tipo de nodo más apropiado para el proyecto en función de las necesidades específicas y las restricciones presupuestarias. Lo que contribuye a una administración eficiente de los recursos y la optimización de aplicaciones LoRaWAN.

Proporcionar información precisa y fácilmente accesible sobre la calidad del aire no solo puede reducir y mitigar los riesgos para la salud asociados con la exposición a niveles elevados de CO₂, sino que puede fomentar una mayor

conciencia ambiental y la adopción de prácticas más sostenibles. Sin embargo, el problema para conseguir esta conciencia ambiental no pasa solo por el uso de la tecnología, sino también es necesario de aplicación de una política pública que de alguna forma aproveche los datos de monitoreo de CO₂ para la reducción de la generación excesiva de este gas.

4.2. Trabajos futuros

Como trabajo futuro, hay varias opciones para mejorar la plataforma como la optimización de la duración de la batería implementando funciones que permitan al administrador monitorizar la duración de la batería de los nodos y recibir notificaciones cuando sea necesario recargarlos, para mantener la operación continua de los dispositivos.

Inicio de sesión con otras plataformas, permitiendo utilizar a los usuarios sus cuentas de mail o redes sociales como Facebook, Twitter, Instagram, Etc.

La personalización de notificaciones permitiendo a los usuarios configurar notificaciones en función de sus necesidades, tendrán un mayor control sobre la información que reciben.

Glosario

ABP Activación por personalización – Activation By Personalization.

API Interfaz de programación de aplicaciones – Application Programming Interface.

CO₂ Dioxido de carbono – Carbon dioxide.

CURL Cliente URL – Client URL.

FTP Protocolo de Transferencia de Archivos – file transfer protocol .

Gateway Puerta de enlace – Gateway.

HTML Lenguaje de Marcado de Hipertexto – HyperText Markup Language.

HTTP Protocolo de Transferencia de Hipertexto – HyperText Transfer Protocol.

HTTPS Protocolo de transferencia de hipertexto seguro – Hypertext Transfer Protocol Secure.

IoT Internet de las Cosas – Internet of Things.

LoRaWAN Largo Alcance Red de Area Amplia de Baja Potencia – Long Range Low Power Wide Area Network.

LPWAN Red de Área Amplia de Baja Potencia – Low Power Wide Area Network.

NDIR Sensor de infrarrojo no dispersivo – Non Dispersive Infrared Detecto.

OMS Organización Mundial de la Salud.

OTAA Activación sobre el aire – Over-The-Air Activation.

PCB Placa de Circuito Impreso – Printed Circuit Board.

PWM Modulación por ancho de pulso – Pulse Width Modulation.

SFTP Protocolo de transferencia segura de archivos – Secure File Transfer Protocol.

TCP/IP Protocolo de control de transferencia/Protocolo de Internet – Transfer Control Protocol/Internet Protocol.

UART Transmisor / Receptor Universal Asíncrono – Universal Asynchronous Receiver / Transmitter.

WSN Redes de Sensores Inalambricos – Wireless Sensor Networks.

Referencias

- [1] J.-D. Páez-Carabajo, M.-K. Huerta, D. Rivas-Lalaleo, G. Vayas, R. Clotet-Martinez y J.-I. Castillo-Velazquez, «Bibliometric study for CO2 measurement using IoT: Looking for the Latin American contributions,» en *2022 IEEE 40th Central America and Panama Convention (CONCAPAN)*, IEEE, 2022, págs. 1-6.
- [2] M. Planelles, «Nuevo record de concentración de CO2 en la atmósfera,» *El país*, págs. 13-16, 2017.
- [3] J. Granados y C. Corner, «Respuesta de las selvas tropicales al incremento de CO2 en la atmósfera,» *Revista Forestal Iberoamericana*, vol. 1, n.º 1, págs. 63-70, 2004.
- [4] *Contaminación atmosférica*, https://www.who.int/es/health-topics/air-pollutiontab=tab_2.
- [5] Á. F. Moreira-Romero, «Contaminación del aire en el medio ambiente por las emisiones de gases tóxicos de empresas industriales en Ecuador,» *Polo del conocimiento*, vol. 3, n.º 7, págs. 299-306, 2018.
- [6] D. Leon y J. C. Cárdenas, «Latinoamérica y el Caribe: Riqueza natural y degradación ambiental en siglo XXI,» *Covid19| Serie De Documentos De Política Pública Pnud*, vol. 14, págs. 1-35, 2020.
- [7] J. D. Burgos Tatés, «Determinación de los niveles de contaminación a la atmósfera por emisiones de CO en el Distrito Metropolitano de Quito,» Tesis de mtría., Ecuador: Latacunga: Universidad Técnica de Cotopaxi (UTC), 2023.
- [8] M. V. R. Moncayo, M. A. C. Núñez, O. H. Á. Hernández y S. V. Pinta, «Emisión de dióxido de carbono de vehículos automotores en la ciudad de Loja, Ecuador,» *CEDAMAZ*, vol. 8, n.º 1, págs. 23-29, 2018.
- [9] D. R. Ortiz Espinosa y col., «Sistema de monitoreo de contaminación del aire con redes de sensores inalámbricos en tiempo real,» B.S. thesis, Quito, 2018.

- [10] A. ROMERO, «M. MODELACIÓN DE DISPERSIÓN ESPACIAL DE CONTAMINANTES DEL AIRE EN LA CIUDAD DE CUENCA,» UNIVERSIDAD SAN FRANCISCO DE QUITO. QUITO, 2012.
- [11] M. E. Muñoz Tello y E. G. Vásquez Córdova, «Estimaciones del potencial de captura de carbono en los parques urbanos y emisiones de CO₂ vehicular en Cuenca, Ecuador,» B.S. thesis, 2020.
- [12] A. Espinoza Molina y col., «Diseño de un sistema de información geográfica para la Red de Monitoreo Ambiental de la ciudad de Cuenca.,» B.S. thesis, Quito, 2011., 2011.
- [13] M Quiñones, V González, R Torres y M Jumbo, «Sistema de monitoreo de variables medioambientales usando WSN y una plataforma cloud,» *Enfoque UT*, vol. 7, n.º 1, págs. 329-343, 2017.
- [14] M. Quiñones-Cuenca, V. González-Jaramillo, R. Torres y M. Jumbo, «Sistema de monitoreo de variables medioambientales usando una red de sensores inalámbricos y plataformas de Internet de las Cosas,» *Enfoque UTE*, vol. 8, págs. 329-343, 2017.
- [15] L. A. Chuqui Nieto, «Desarrollo de una plataforma abierta para el almacenamiento y gestión de información de nodos móviles para internet de las cosas de la célula inteligente de la Universidad Politécnica Salesiana,» B.S. thesis, 2016.
- [16] J. C. Maza Pacheco, J. Collantes y col., «Diseño de una red inalámbrica de monitoreo de co₂ basado en la tecnología sigfox,» B.S. thesis, Espol, 2018.
- [17] D. R. Ortega Segura, «Implementación de un prototipo para el Monitoreo de dióxido de carbono (CO₂), monóxido de carbono (CO) y metano (CH₄) en lugares residenciales utilizando el sistema operativo Android y Arduino,» B.S. thesis, 2017.
- [18] M. Arroyo y A. Ramírez-Monroy, «Dióxido de carbono, sus dos caras,» *Anales de Química de la RSEQ*, vol. 116, n.º 2, págs. 81-87, 2020.
- [19] J. W. Sancarranco Estela, «El dióxido de carbono como agente de contaminación ambiental en la región Piura,» 2022.

- [20] M. Romero Placeres, F. Diego Olite y M. Álvarez Toste, «La contaminación del aire: su repercusión como problema de salud,» *Revista cubana de higiene y epidemiología*, vol. 44, n.º 2, págs. 0-0, 2006.
- [21] «Contaminación atmosférica.» (2023), dirección: https://www.who.int/es/health-topics/air-pollution#tab=tab_1 (visitado 23-05-2023).
- [22] «Contaminación atmosférica.» (2023), dirección: https://www.who.int/es/health-topics/air-pollution#tab=tab_2 (visitado 23-05-2023).
- [23] M. J. B. Subils y F. B. Domínguez, «NTP 549: El dióxido de carbono en la evaluación de la calidad del aire interior,» *España: Centro Nacional De Condiciones De Trabajo*, 2000.
- [24] «Estrategia de monitoreo,» *Instituto de Hidrología, Meteorología y Estudios Ambientales*, 2015.
- [25] A. F. Montaña, C. G. López y V. D. Belitskaya, «4. La calidad de aire en ciudades,» *Martha Georgina Orozco M. Coordinación y Edición*, pág. 75,
- [26] H. Sharma y S. Sharma, «A review of sensor networks: Technologies and applications,» *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, págs. 1-4, 2014.
- [27] M. K. Huerta, A. Garcia-Cedeno, J. C. Guillermo y R. Clotet, «Wireless Sensor Networks Applied to Precision Agriculture: A worldwide literature review with emphasis on Latin America,» *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, n.º 4, págs. 209-222, 2021.
- [28] C. Y. F. Banchón y J. C. P. Maza, «Diseño de una red inalámbrica de monitoreo de CO2 basado en la tecnología Sigfox,» 2018.
- [29] R. S. Sinha, Y. Wei y S.-H. Hwang, «A survey on LPWA technology: LoRa and NB-IoT,» *Ict Express*, vol. 3, n.º 1, págs. 14-21, 2017.
- [30] P. Levchenko, D. Bankov, E. Khorov y A. Lyakhov, «Performance Comparison of NB-Fi, Sigfox, and LoRaWAN,» *Sensors*, vol. 22, 24 dic. de 2022, ISSN: 14248220. DOI: 10.3390/s22249633.
- [31] S. Madakam, V. Lake, V. Lake, V. Lake y col., «Internet of Things (IoT): A literature review,» *Journal of Computer and Communications*, vol. 3, n.º 05, pág. 164, 2015.

- [32] G. Pradyumna, B. Omkar y B. Sagar, «Introduction to IOT,» *International Advanced Research Journal in Science, Engineering and Technology*, vol. 5, n.º 1, págs. 41-44, 2018.
- [33] M. U. Farooq, M. Waseem, S. Mazhar, A. Khairi y T. Kamal, «A review on internet of things (IoT),» *International journal of computer applications*, vol. 113, n.º 1, págs. 1-7, 2015.
- [34] E. Serrano Hernández y col., «Análisis y desarrollo de los servicios de seguridad post-cuánticos en IOT,» Tesis de mtría., Benemérita Universidad Autónoma de Puebla, 2022.
- [35] L. B. Estrada Bolívar, «Confiabilidad de los sistemas de seguridad del hogar inteligente basado en IoT,» 2021.
- [36] D. E. Almeida Correa, «Diseño e implementación de un sistema de localización y rastreo GPS de lanchas pesqueras utilizando comunicación inalámbrica,» B.S. thesis, 2022.
- [37] N. Aakvaag y J.-E. Frey, «Redes de sensores inalámbricos,» *Revista ABB*, vol. 2, n.º 2006, págs. 39-42, 2006.
- [38] R. F. Martínez, J. O. Meré, F. J. M. de Pisón Ascacíbar, A. G. Marcos y F. A. Elías, «Redes inalámbricas de sensores: teoría y aplicación práctica,» *Grupo de Investigación EDMANS-Universidad de la Rioja, España*, 2009.
- [39] C. C. Nayibe, T. B. D. Alexander, A. Z. L. Cristina y C. O. L. Miguel, «Redes de sensores inalámbricos,» *Universidad Nacional de Colombia, Bogotá*, 2014.
- [40] E. P. ESPINEL PACHECO, «SISTEMAS DE MONITOREO Y SEGUIMIENTO PARA PERSONAS CON SÍNDROME DE DOWN UTILIZANDO TECNOLOGÍA LONG RANGE,» Tesis de mtría., Quito, Ecuador: Universidad Tecnológica Israel, 2022.
- [41] S. Devalal y A. Karthikeyan, «LoRa technology-an overview,» en *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, IEEE, 2018, págs. 284-290.
- [42] M. A. Moya Quimbita, «Evaluación de pasarela LoRa/LoRaWAN en entornos urbanos,» 2018.
- [43] I. Ordóñez Monfort, «Estudio de la arquitectura y el nivel de desarrollo de la red LoRaWAN y de los dispositivos LoRa.,» 2017.

- [44] J. P. M. Álvarez, V. I. R. Abdalá, F. M. R. M. Barboza y F. R. C. Soria, «Estudio descriptivo de LoRaWAN y aplicaciones específicas,» *Difu100ci@, Revista de difusión científica, ingeniería y tecnologías*, vol. 15, n.º 1, págs. 8-17, 2021.
- [45] A. C. Cuji Torres y D. O. Sarumeño Avila, «Integración de ChirpStack con Node Red para alojar datos en Google Cloud Platform,» B.S. thesis, 2022.
- [46] F. Delgado Ferro, J. Navarro Ortiz, N. Chinchilla Romero, J. J. Ramos Muñoz y col., «Arquitectura LoRaWAN para entornos sin cobertura,» 2021.
- [47] A. Jimeno-Beltrán, «Análisis de Soluciones Multicloud. Diseño e Implementación de una red privada LoRaWAN,» 2022.
- [48] J. R. M. Ríos, N. M. L. Mora, M. P. Z. Ordóñez y E. L. L. Sojos, «Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python,» *Archivo de la revista Latinoamericana de Ingeniería de Software*, vol. 4, n.º 4, págs. 201-207, 2016.
- [49] N. Simmonds Samper y col., «Análisis de frameworks para desarrollo de aplicaciones móviles y web,» 2019.
- [50] R. I. Bravo Rivera, «Análisis comparativo entre frameworks, para el desarrollo de aplicaciones móviles multiplataformas,» B.S. thesis, BABAHOYO: UTB, 2021, 2021.
- [51] K. S. Zurita Núñez, «Frameworks para el desarrollo de aplicaciones móviles multiplataforma compiladas de forma nativa–estudio comparativo y ejemplos prácticos,» B.S. thesis, PUCE-Quito, 2020.
- [52] *CubeCell Series – Heltec Automation [Online]*, Available: https://heltec.org/cubecell_overview/, 2023.
- [53] *CubeCell – Tablero de desarrollo – Heltec Automation*, Available: <https://heltec.org/project/htcc-ab01/>, 2023.
- [54] *LipoRider Pro | Wiki [Online]*, Available: https://wiki.seeedstudio.com/Lipo_Rider_Pro/, 2023.
- [55] M. Aziz, P Saptiani, M Iryanti y A Aminudin, «Design of carbon dioxide level measures on peat soil with MG 811 sensor,» en *Journal of Physics: Conference Series*, IOP Publishing, vol. 1280, 2019, pág. 022 061.
- [56] S. Abraham y X. Li, «A cost-effective wireless sensor network system for indoor air quality monitoring applications,» *Procedia Computer Science*, vol. 34, págs. 165-171, 2014.

- [57] K. Q. Abdelfadeel, Y. Samarawickrama y V. Cionca, «How to Conduct LoRaWAN Site Surveys,» vol. 2019-Octob, IEEE, 2019, págs. 133-138, ISBN: 9781728133164. DOI: 10.1109/WiMOB.2019.8923593.
- [58] C.-T. Chiang, M. Chung y M.-Y. Huang, «Design of a gas sensor transducer circuitry with calibration ability for CO₂ concentration detection,» *IEEE Sensors Journal*, vol. 16, n.º 16, págs. 6367-6373, 2016.
- [59] A Martín-Garín, J. Millán-García, A Bañri, J Millán-Medel y J. Sala-Lizarraga, «Environmental monitoring system based on an Open Source Platform and the Internet of Things for a building energy retrofit,» *Automation in Construction*, vol. 87, págs. 201-214, 2018.
- [60] M Toat, S Soekirno y F Faisal, «Monitoring system for carbon dioxide gas concentration using NDIR sensors,» en *AIP Conference Proceedings*, AIP Publishing, vol. 2604, 2023.
- [61] S. Ali, C. M. Kumar, G. R. Kullayappa y V. Saritha, «Development of cost-effective measurement system for outdoor and indoor environment parameters using IoT Technology,» *Solid State Technology*, vol. 63, n.º 5, págs. 8503-8520, 2020.
- [62] J. Kwon, G. Ahn, G. Kim, J. C. Kim y H. Kim, «A study on NDIR-based CO₂ sensor to apply remote air quality monitoring system,» en *2009 Iccas-Sice*, IEEE, 2009, págs. 1683-1687.
- [63] A. Curl, J. D. Nelson y J. Anable, «Does accessibility planning address what matters? A review of current practice and practitioner perspectives,» *Research in Transportation Business & Management*, vol. 2, págs. 3-11, 2011.
- [64] B. Christudas y B. Christudas, «cURL and Postman,» *Practical Microservices Architectural Patterns: Event-Based Java Microservices with Spring Boot and Spring Cloud*, págs. 847-855, 2019.
- [65] A. M. Da Rocha, M. A. De Oliveira, P. J. FM y G. G. H. Cavalheiro, «ABP vs. OTAA activation of LoRa devices: an Experimental Study in a Rural Context,» en *2023 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2023, págs. 630-634.
- [66] E. Gresak y M. Voznak, «Protecting gateway from abp replay attack on lorawan,» en *AETA 2018-Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*, Springer, 2020, págs. 400-408.

- [67] S. Na, D. Hwang, W. Shin y K.-H. Kim, «Scenario and countermeasure for replay attack using join request messages in LoRaWAN,» en *2017 international conference on information networking (ICOIN)*, IEEE, 2017, págs. 718-720.