



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no
Renovables

Ingeniería en Telecomunicaciones

**Diseño e implementación de una red inalámbrica de sensores de bajo costo,
para la medición de parámetros de la calidad del aire**

**Trabajo de Integración Curricular,
previo a la obtención del título de
Ingeniero en Telecomunicaciones**

AUTOR:

Paul Alexander Aponte Ortiz

DIRECTOR:

Ing. Marco Augusto Suing Ochoa. Mg. Sc

Loja – Ecuador

2023

Educamos para Transformar

Certificación

Loja, 30 de agosto del 2023

Ing. Marco Augusto Suing Ochoa. Mg. Sc

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo proceso de la elaboración del Trabajo de Integración Curricular denominado: **Diseño e implementación de una red inalámbrica de sensores de bajo costo, para la medición de parámetros de la calidad del aire**, previo a la obtención del título de **Ingeniero en Telecomunicaciones**, de la autoría del estudiante **Paul Alexander Aponte Ortiz** , con **cédula de identidad Nro. 1105456287** , una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Ing. Marco Augusto Suing Ochoa. Mg. Sc

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Yo, **Paul Alexander Aponte Ortiz**, declaro ser el autor del presente Trabajo de Integración Curricular y eximo expresamente a la Universidad nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi Trabajo de Integración Curricular en el Repositorio Institucional – Biblioteca Virtual.

Firma:



Cédula de identidad: 1105456287

Fecha: 14 de diciembre del 2023

Correo electrónico: paul.aponte@unl.edu.ec

Teléfono: 0989789125

Carta de Autorización por parte del autor, para la consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular.

Yo, **Paul Alexander Aponte Ortiz**, declaro ser el autor del Trabajo de Integración Curricular denominado: **diseño e implementación de una red inalámbrica de sensores de bajo costo, para la medición de parámetros de la calidad del aire**, como requisito para optar por el título de **Ingeniero en Telecomunicaciones**, autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Integración Curricular que realice un tercero.

Para constancia de esta autorización, suscribo, en la ciudad de Loja, a los catorce días del mes de diciembre del dos mil veintitrés.

Firma:



Autor: Paul Alexander Aponte Ortiz

Cedula de identidad: 1105456287

Dirección: Loja, (Esteban Godoy)

Correo electrónico: paul.aponte@unl.edu.ec

Teléfono: 0989789125

DATOS COMPLEMENTARIOS:

Director de Trabajo de Integración Curricular: Ing. Marco Augusto Suing Ochoa. Mg. Sc

Dedicatoria

Al culminar mi Trabajo de Integración Curricular me permito agradecer de manera especial a las personas que me apoyaron de principio a fin, a superar y lograr este objetivo; con respeto, amor y humildad lo dedico:

A Dios, por permitirme la vida, sus bendiciones e iluminar mi camino, darme la inteligencia y sabiduría, para poder lograr cada uno de mis propósitos en mi vida.

A mis padres Hilario Aponte y Gloria Ortiz, por ser el ejemplo y pilar fundamental en mi vida, en toda mi educación, tanto académica, como personal; su incondicional apoyo perfectamente mantenido a través del tiempo. Por su comprensión, amor brindado, su ayuda en los momentos difíciles, y con los recursos necesarios para estudiar.

A mis hermanas Ing. Liliana Ortiz, Ing. Fernanda Aponte y Juliana Aponte, por estar siempre presente con su apoyo y cariño en todo momento.

A mis sobrinos Sofía y Aarón, por ser la alegría y motivación para cumplir cada una de mis metas.

A mi Abuelita Mariana Granda por brindarme su amor y bendiciones.

A mi novia Ing. Katty Rentería, por su apoyo, amor y comprensión.

Paul Alexander Aponte Ortiz

Agradecimiento

El presente Trabajo de Titulación primeramente me gustaría agradecer a Dios por sus inmensas bendiciones para hacer realidad este sueño anhelado.

A la Universidad Nacional de Loja, por darme la oportunidad de estudiar y ser un profesional. A mi director de Trabajo de Integración Curricular, Ing. Marco Suing, por su esfuerzo y dedicación, quienes, con sus conocimientos, sus experiencias, su paciencia y sus motivaciones han logrado en mí que pueda terminar mis estudios con éxito.

También me gustaría agradecer a mis profesores durante toda mi carrera profesional porque todos han aportado con un granito de arena a mi formación.

Agradezco a mis padres porque estuvieron a mi lado motivándome y dándome fuerzas para seguir adelante, a mis hermanas por su comprensión y estar siempre a mi lado. A mis sobrinos por ser mi motivación.

Finalmente, a mis familiares y amigos por su apoyo incondicional durante mi vida universitaria.

Paul Alexander Aponte Ortiz

Índice de contenidos

Portada	i
Certificación	ii
Autoría	iii
Carta de Autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de contenidos	vii
Índice de tablas.....	x
Índice de figuras	xi
Índice de anexos	xvi
1. Título	1
2. Resumen	2
Abstrac.....	3
3. Introducción	4
3.1. Objetivos.....	4
4. Marco teórico	6
4.1. El Aire y su Importancia	6
4.1.1. Definición del Aire.....	6
4.1.2. Importancia.....	6
4.2. Calidad del Aire.....	6
4.2.1. Calidad el Aire en el Ecuador.....	6
4.2.2. Índices de la calidad del Aire	7
4.2.3. Contaminantes.....	8
4.2.4. Efectos en la Salud	10
4.3. Red de Sensores Inalámbricos Para Monitoreo de la Calidad de Aire	11
4.3.1. Características de una WSN.....	12

4.3.2.	Topologías de Red.....	13
4.3.3.	Ventajas y desventajas.....	14
4.3.4.	Fases de despliegue de las WSN.	14
4.3.5.	Protocolos de Comunicación Inalámbrica para una WSN	15
4.3.6.	Estándares para WSN.....	17
4.3.7.	Seguridad WSN.....	18
4.3.8.	Red General de Medición de la Calidad del Aire.....	19
4.4.	Redes Low Power Wide Area Networking (LPWAN).....	19
4.4.1.	Características	20
4.4.2.	Aplicaciones de las LPWAN.....	20
4.4.3.	Requisitos de la Aplicación LPWAN.....	21
4.4.4.	Requisitos de tráfico.....	21
4.4.5.	Capacidad y Densificación.....	22
4.4.6.	Operaciones energéticamente eficientes y fuentes de baja potencia	22
4.4.7.	Cobertura.....	23
4.4.8.	Identificación de ubicación	23
4.4.9.	Seguridad y privacidad.....	23
4.4.10.	Rentabilidad.....	24
4.4.11.	Elección de la tecnología LPWAN.....	25
4.5.	LoRa y LoRaWAN.....	27
4.5.1.	LoRa	27
4.5.2.	LoRaWAN	27
4.6.	Plataforma IoT	29
4.6.1.	Ubidots	30
4.6.2.	ThinkSpeak.....	31
4.6.3.	My Devices Cayenne	32
4.6.4.	Comparativa de plataformas Web	34

5. Metodología.....	35
5.1. Investigación y análisis de los parámetros más importantes de la calidad del aire ...	35
5.2. Evaluar y caracterizar el por qué el uso de la tecnología LPWAN, LoRA para el desarrollo del proyecto, referente a su, aplicación, alcance, funcionamiento, arquitectura y protocolos.	35
5.3. Diseño de una red de sensores basada en la tecnología LoRa	36
5.3.1. Diseño de la red LoRaWAN	36
5.3.2. Despliegue de la red LoRaWAN.....	37
5.3.3. Nodo Gateway.....	37
5.3.4. Configuración para The Things Network.....	39
5.3.5. Nodos Sensores	46
5.3.6. Diseño de la conexión de los Nodos Sensores	48
5.3.7. Configuración de los Módulos Heltec Cubecell.....	49
5.3.8. Calibración de sensores	71
5.3.9. Materiales de software.....	73
5.3.10. Materiales de hardware.....	73
6. Resultados	74
7. Discusión	116
8. Conclusiones	119
9. Recomendaciones	121
10. Bibliografía	123
11. Anexos	126

Índice de tablas:

Tabla 1. Interpretación del índice de la calidad del aire.....	7
Tabla 2. <i>Valores fijados por la NECA de los contaminantes</i>	9
Tabla 3. Principales características de una WSN	12
Tabla 4. <i>Comparativa de las Tecnologías LPWAN</i>	24
Tabla 5. <i>Comparativa de las Tecnologías LPWAN en Términos IoT</i>	26
Tabla 6. <i>Comparativa de Plataformas Web</i>	34
Tabla 7. Características de los sensores del Nodo Sensor.....	48
Tabla 8. Materiales de Software	73
Tabla 9. Materiales de Hardware.....	73

Índice figuras:

Figura 1. Infecciones respiratorias agudas en el Ecuador.	7
Figura 2. <i>Red de sensores inalámbricos.</i>	12
Figura 3. Topologías de Redes de Sensores.	13
Figura 4. Protocolos de una WSN.	16
Figura 5. Comparativa de Tecnologías Inalámbricas.	24
Figura 6. Gráfica Comparativa.	26
Figura 7. Arquitectura LoRaWAN.	29
Figura 8. Ubidots.	30
Figura 9. ThinkSpeak.	32
Figura 10. My Devices Cayenne.	33
Figura 11. Red LoRaWAN para el Monitoreo de Gases Ambientales.	36
Figura 12. Materiales que Componen el nodo Gateway.	37
Figura 13. Inicio de sesión para ingresar al Gateway	38
Figura 14. Configuración inicial "Quick Set"	39
Figura 15. Configuración LoRa.	40
Figura 16. Configuración Server.	40
Figura 17. Servidor de red LoRaWAN (LNS) añadido.	41
Figura 18. Dispositivo LoRa (Concentrador).	41
Figura 19. Agrega o elimina distintos LNS a los que reenviar los paquetes LoRa.	42
Figura 20. Habilitar el Gateway	42
Figura 21. Tráfico de mensajes del gateway	43
Figura 22. Selección de clúster para integración de gateway y nodos sensores a la TTN.	43
Figura 23. Pestaña de ingreso y selección de Dispositivos, nodos y gateway.	44
Figura 24. Ventana para agregar el Gateway. Fuente: TTN	44
Figura 25. Configuración de parámetros del Gateway.	45
Figura 26. Configuración y registro del Gateway.	46
Figura 27. Módulo Heltec Cube Cell	47
Figura 28. Conexión de los nodos sensores.	49
Figura 29. IDE de Arduino.	50
Figura 30. Link de Descarga. json.	50
Figura 31. Gestor de placas.	51
Figura 32. Instalación de cubecell en el IDE de arduino	51

Figura 33. Elección del módulo y modelo a utilizar	52
Figura 34. Configuración de los parámetros de los módulos	52
Figura 35. Proceso de selección del código	53
Figura 36. Código para trabajar con los módulos Cubecell mediante OTAA en TTN.....	54
Figura 37. Página TTN.....	54
Figura 38. Completamos los parámetros y creamos la aplicación	55
Figura 39. Aplicación creada para operar con los módulos.	55
Figura 40. Registro de los módulos.....	55
Figura 41. Aplicación creada con los módulos registrados.....	56
Figura 42. Parámetros del módulo 1 en la TTN.....	56
Figura 43. Parámetros del módulo 2 en la TTN.....	57
Figura 44. Código Configurado para el módulo 1 en la TTN.....	57
Figura 45. Código Configurado para el módulo 2 en la TTN.....	58
Figura 46. Recepción y envío de datos del módulo 1.	59
Figura 47. Envío y recepción de datos del módulo 2.	59
Figura 48. Configuración del payload formatters. Fuente: TTN.....	60
Figura 49. Datos mostrados en Live Data. Fuente TTN	60
Figura 50. Creación del Plugin de datos.	61
Figura 51. Plugin de datos activo.	61
Figura 52. Integración de TTN.....	64
Figura 53. Token Ubidots.	64
Figura 54. Variables enviadas de TTN a Ubidots.	64
Figura 55. Ventana para la creación de eventos.....	64
Figura 56. Eventos creados parada cada gas contaminante.	65
Figura 57. Busque en Telegram de BotFather.....	66
Figura 58. Chat en de Telegram, elección del nombre del bot	66
Figura 59. Token de Telegram	67
Figura 60. Configuración del complemento de Telegram en Ubidots	68
Figura 61. URL de Telegram para crear las alertas.	68
Figura 62. Configuración de las alertas para Telegram en Ubidots.....	69
Figura 63. widget disponible en Ubidots.	70
Figura 64. Tablero Web que presenta los valores de los medidores de calidad de aire.....	70
Figura 65. Calentamiento de los Sensores.	71
Figura 66. Estructura interna del Nodo Sensor	72

Figura 67. Estructura externa del Nodo Sensor.....	72
Figura 68. a) Ubicación del nodo sensor 1 mercado mayorista b) Ubicación en google maps	74
Figura 69. a). Ubicación del nodo sensor 2 Terminal Terrestre Loja b). Ubicación en google maps	75
Figura 70. Dashboard de Ubidots de los estándares de visualización.....	76
Figura 71. Dashboard de Ubidots de los periodos de tiempo para la visualización de los datos	77
Figura 72. Dashboard de eventos de Ubidots de configuración de alertas	77
Figura 73. Dashboard de Ubidots de los servidores de envío de alertas.....	78
Figura 74. Dashboard de visualización de los datos del nodo 1 ubicado en el mercado mayorista	79
Figura 75. Widgets de medición del gas CO2 en el mercado mayorista	79
Figura 76. Valores mínimos obtenidos del CO2	80
Figura 77. Valores máximos obtenidos del CO2	80
Figura 78. Valores promedio obtenidos del CO2.....	81
Figura 79. Widgets de Visualización de los datos obtenidos del O3	81
Figura 80. Valores mínimos obtenidos por el O3	82
Figura 81. Valores máximos obtenidos del O3	82
Figura 82. Valores promedio obtenidos del O3	83
Figura 83. Dashboard de los gráficos a partir de los valores obtenidos del NO2	83
Figura 84. Valores mínimos obtenidos del NO2.....	84
Figura 85. Valores máximos obtenidos del NO2	84
Figura 86. Media de los valores obtenidos del NO2	85
Figura 87. Widgets de visualización de los datos obtenidos del PM2,5	85
Figura 88. Valores mínimos obtenidos del PM2,5.....	86
Figura 89. Valores máximos obtenidos del PM2,5	86
Figura 90. valores promedios obtenidos del PM2,5.....	87
Figura 91. Widgets de visualización de los datos registrados del PM10	87
Figura 92. Valores mínimos obtenidos del PM10.....	88
Figura 93. Valores máximos obtenidos del PM10	88
Figura 94. Valores promedio obtenidos del PM10	89
Figura 95. Notificaciones de alertas del CO2; a) Telegram, b) SMS y c) Correo electrónico	90
Figura 96. Notificaciones de alertas del O3; a) Telegram, b) SMS y c) Correo electrónico ..	91

Figura 97. Notificaciones de alertas del NO ₂ ; a) Telegram, b) SMS y c) Correo electrónico	92
Figura 98. Notificaciones de alertas del PM _{2,5} ; a) Telegram, b) SMS y c) Correo electrónico	93
Figura 99. Notificaciones de alertas del PM ₁₀ ; a) Telegram, b) SMS y c) Correo electrónico	94
Figura 100. Dashboard de Ubidots para visualizar los datos de los gases.	95
Figura 101. Widgets de medición del gas CO ₂ en el terminal terrestre.	95
Figura 102. Valores mínimos de los valores arrojados de CO ₂ .	96
Figura 103. <i>Valores máximos arrojados por el CO₂</i>	96
Figura 104. <i>Valor promedio del CO₂</i>	97
Figura 105. Widgets de visualización de los datos de O ₃ .	97
Figura 106. <i>Valores mínimos obtenidos de O₃</i>	98
Figura 107. Valores máximos almacenados de O ₃	98
Figura 108. <i>Valores promedios obtenidos de O₃</i>	99
Figura 109. Widgets de visualización de los datos de NO ₂ .	99
Figura 110. Valores mínimos obtenidos de NO ₂	100
Figura 111. Valores máximos obtenidos de NO ₂	100
Figura 112. Promedio de los valores arrojados de NO ₂	101
Figura 113. Widgets de visualización de los datos	101
Figura 114. Valores mínimos obtenidos del SO ₂	102
Figura 115. Valores máximos obtenidos del SO ₂ .	102
Figura 116. Valores promedios obtenidos de SO ₂ .	103
Figura 117. Widgets de visualización de los datos de PM _{2,5} .	103
Figura 118. Valores mínimos obtenidos del PM _{2,5} .	104
Figura 119. Valores máximos obtenidos del PM _{2,5}	104
Figura 120. Valores promedio obtenidos del PM _{2,5}	105
Figura 121. Widgets de visualización de los datos del PM ₁₀ .	105
Figura 122. Valores mínimos obtenidos del PM ₁₀ .	106
Figura 123. Valores máximos obtenidos del PM ₁₀	106
Figura 124. Valores promedios obtenidos del PM ₁₀ .	107
Figura 125. Notificaciones de alertas del CO ₂ ; a) Telegram, b) SMS y c) Correo electrónico	108
Figura 126. Notificaciones de alertas del O ₃ ; a) Telegram, b) SMS y c) Correo electrónico	109

Figura 127. Notificaciones de alertas del gas NO ₂ , a) Telegram, b) SMS y c) Correo electrónico	110
Figura 128. Notificaciones de alertas del PM _{2,5} ; a) Telegram, b) SMS y c) Correo electrónico	111
Figura 129. Notificaciones de alertas del PM ₁₀ ; a) Telegram, b) SMS y c) Correo electrónico	112
Figura 130. Datos almacenados en Excel de los gases obtenidos por el NODO 2	113
Figura 131. Datos almacenados en Excel de los gases obtenidos por el NODO 1	114
Figura 132. Visualización de los datos en tabla y gráfica dinámica en Excel	114
Figura 133. Visualización de los datos en tabla y gráfica dinámica en Excel	115
Figura 134. Conexión de los dispositivos	126
Figura 135. Conexión I2C.....	131
Figura 136. Visualización del envío y recepción de los datos mediante I2C.....	131
Figura 137. Proceso de Calibración en el IDE de Arduino.....	139
Figura 138. Proceso de calibración en el IDE de arduino.....	143
Figura 139. Proceso de calibración en el IDE de arduino.....	147
Figura 140. Diseño del circuito digital en proteus.....	170
Figura 141. Impresión del circuito en la vaquelita.....	170

Índice de anexos:

Anexo 1. Conexión de los Sensores con el Arduino Uno	126
Anexo 2. Conexión I2C entre el Arduino UNO y el Modulo Cubecell.....	131
Anexo 3. Calibración sensor MQ7.....	139
Anexo 4. Calibración sensor MQ136.....	143
Anexo 5. Calibración sensor MQ131	147
Anexo 6. Calibración Sensor DSM501	151
Anexo 7. Código para el envío de los datos del módulo 2.....	161
Anexo 8. <i>Diseño del circuito</i>	170
Anexo 9. Construcción de los nodos sensores	171
Anexo 10. Ubicación del Gateway.....	172
Anexo 11. Certificado de traducción del resumen.....	190

1. Título

**Diseño e implementación de una red inalámbrica de sensores de bajo costo,
para la medición de parámetros de la calidad del aire**

2. Resumen

El presente proyecto pretende diseñar e implementar una red de sensores con tecnología LoRa, con el fin de monitorear los principales factores contaminantes del aire en la ciudad de Loja. Partiendo de la investigación y evaluación de la tecnología LoRa, tanto en su estructura, protocolo y funcionamiento se propondrá realizar el diseño e implementación de la red mencionada. Dicha red contiene diferentes componentes IoT, hardware (sensores) y software (almacenamiento en la nube), los cuales serán utilizados para el desarrollo y visualización de los datos, la información será mostrada mediante una plataforma web. De esta manera, se contribuye al proyecto de convertir a la ciudad de Loja en una Ciudad inteligente (Smart City).

Se analizará el comportamiento de la contaminación a partir de las mediciones de concentraciones de Dióxido de Azufre (SO₂), Ozono (O₃), Dióxido de Carbono (CO₂), Dióxido de Nitrógeno (NO₂) y Material Particulado (MP), medidos en porcentajes por millón (ppm). Las mediciones serán extraídas por una red de sensores inalámbricos ubicados en lugares estratégicos de la ciudad de Loja: el Terminal Terrestre y el Mercado Mayorista.

La red inalámbrica basada en el estándar LoRaWAN, se compone de un nodo sensor (prototipo móvil) y un Gateway. Los nodos sensores envían los datos de forma inalámbrica hacia el Gateway, el cual a través de una plataforma Web (Ubidots), muestra el comportamiento que describen la medición y almacenamiento de la información en esta plataforma Web.

Palabras Claves: *LoRa, IoT, LoRaWAN, ciudad inteligente, Calidad aire, Ubidots, Gateway, Nodos.*

Abstrac

This project aims to design and implement a network of sensors with LoRa technology, in order to monitor the main air pollutants in the city of Loja. Starting from the research and evaluation of LoRa technology, both in its structure, protocol and operation, the design and implementation of the mentioned network can be carried out. This network contains different IoT components, hardware (sensors) and software (cloud storage), which will be used for the development and visualization of data, the information will be displayed through a web platform. In this way, it contributes to the project of turning the city of Loja into a Smart City.

With the above, the behavior of pollution will be analyzed from measurements of concentrations of Sulfur Dioxide (SO₂), Ozone (O₃), Carbon Monoxide (CO), Nitrogen Dioxide (NO₂) and Particulate Matter (PM), measured in percentages per million (ppm). The measurements will be taken by a network of wireless sensors located in strategic places in the city of Loja: The Terrestrial Terminal and Wholesale Market.

The wireless network based on the LoRaWAN standard is made up of a sensor node (mobile prototype) and a Gateway. The sensor nodes send the data wirelessly to the Gateway, which through a Web platform (Ubidots), shows the behavior that describes the measurement and storage of information on a Web platform.

Keywords: *LoRa, IoT, LoRaWAN, smart city, air quality, Ubidots, Gateawy, Nodes.*

3. Introducción

La contaminación del aire representa un importante riesgo para la salud y el medioambiente. Los impactos en la salud por la contaminación del aire han sido documentados en Ecuador se encontró vínculos entre contaminación atmosférica y enfermedades respiratorias, déficits neuroconductuales, condiciones cardiovasculares y altos costos relacionados con estas enfermedades.

En la ciudad de Loja, actualmente no se cuentan con prototipos inteligentes para medir la calidad del aire, donde los principales contaminantes del aire son: material particulado fino (PM 2,5), dióxido de carbono (CO₂), hidrocarburos (HC), óxidos de nitrógeno (NO_X) dióxido de azufre (SO₂) y oxidantes fotoquímicos expresados como ozono (O₃), todos estos se derivan fundamentalmente por procesos de combustión.

El presente proyecto tiene como objetivo desarrollar dos prototipos inteligentes, mediante la comunicación inalámbrica con la tecnología LoRa y el protocolo LoRaWAN, para el envío y recepción de los datos, que correspondan a los parámetros ambientales antes mencionados, estos prototipos serán ubicados en puntos estratégicos de la ciudad de Loja, permitiendo el almacenamiento y visualización de los datos en tiempo real; habilitando su uso a autoridades, especialistas, ambientalista entre otros, para la toma de decisiones.

El proyecto consta de 4 secciones. En la primera sección cuenta con tres apartados, título, resumen e introducción. La segunda sección se realiza el marco teórico donde se describe conceptos importantes como: el aire y su importancia, Red de Sensores Inalámbricos para Monitoreo de la Calidad de Aire, Redes Low Power Wide Area Networking (LPWAN), Lora/LoRaWAN e Interfaz/base de datos. En la tercera sección se desarrolla e implementa la red de sensores, la plataforma web, base de datos. Finalmente, en la última sección se describen, conclusiones, recomendaciones y anexos lo que nos permitirá conocer los resultados del desarrollo de los prototipos para medir los parámetros de calidad de aire.

3.1. Objetivos

Con la finalidad de realizar un correcto desarrollo del presente proyecto de titulación se han establecido los siguientes objetivos:

Objetivo General

- Implementar una red inalámbrica de sensores utilizando la tecnología LoRa para evaluar la contaminación del aire en la ciudad de Loja.

Objetivos específicos

- Realizar un estudio bibliográfico con el fin de identificar los parámetros más importantes de la calidad del aire.
- Evaluar y caracterizar el porqué del uso de la tecnología LPWAN, LoRa para el desarrollo del proyecto.
- Diseñar una red de sensores basado en el protocolo LoraWAN.
- Implementar una interfaz web que permita el acceso permanente a la información tomada de los sensores.
- Desarrollar una base de datos que permita el registro de los datos y posterior almacenamiento de forma inmediata, con la finalidad de mostrar registros históricos para toma de decisiones.

4. Marco teórico

4.1. El Aire y su Importancia

4.1.1. Definición del Aire

“El aire ambiente o atmosférico, se refiere a cualquier parte no limitada de la atmósfera, y se caracteriza como una combinación de gases en la que el contenido típico, es al menos un veinte por ciento (20%) de oxígeno, aproximadamente un setenta y nueve por ciento (79%) de nitrógeno, junto con cantidades variables de gases inertes y vapor de agua, en proporciones que varían en términos de volumen.” (OMS, 2021).

4.1.2. Importancia

El recurso natural primordial más fundamental para los organismos vivos es el aire. Su importancia radica en los beneficios fundamentales que aporta, como ser esencial para el funcionamiento adecuado del planeta y para el desarrollo humano y económico. Asimismo, desempeña un papel vital en las funciones esenciales de los seres vivos, tales como la respiración y el proceso de fotosíntesis. Además, es crucial para la polinización y actúa como fuente de energía renovable en la forma de energía eólica. El aire también ofrece protección contra los rayos solares, regula la temperatura y provee agua para el consumo de los organismos. Es un componente esencial en procesos industriales y productivos, además de estar relacionado con el efecto invernadero y el equilibrio en la naturaleza. También es responsable de la existencia del viento, los fenómenos naturales como el fuego, las nubes, las lluvias, y otros acontecimientos naturales. (Boffetta et al., 1993).

4.2. Calidad del Aire

4.2.1. Calidad el Aire en el Ecuador

En la actualidad, en Ecuador, existe una escasez de investigaciones que aborden los efectos de la contaminación del aire en la salud de las personas. De manera similar, los problemas medioambientales son más pronunciados en las áreas urbanas densamente pobladas, como Quito, Guayaquil y Cuenca. La Organización Mundial de la Salud sostiene que el acceso al aire limpio es un derecho innegable de la población para garantizar su bienestar y salud. Por lo tanto, es esencial que Ecuador adopte las normas de prevención y control de la contaminación atmosférica propuestas en el Plan Nacional de Calidad del Aire (GEO-Loja, 2007).

La contaminación del aire es producida por diversos factores relacionados con las actividades diarias de las personas, sector industrial y automotriz, explotaciones mineras, emisiones de gases tóxicos, entre otros; por lo que las enfermedades generadas por esta problemática se presentan en su mayoría afectando las vías respiratorias (MAATE, 2011). En la figura 1, se detalla el total de casos registrados de infecciones respiratorias agudas en el Ecuador.



Figura 1. Infecciones respiratorias agudas en el Ecuador.

Fuente: MAATE, 2011

4.2.2. Índices de la calidad del Aire

Los índices de calidad de aire son un conjunto de información que permite determinar de modo eficaz el estado del aire para notificar a la comunidad sus posibles efectos en la salud. La manera más elemental de evaluar la calidad atmosférica es establecer relaciones entre las directrices propuestas por la Norma Ecuatoriana de la Calidad del Aire (NECA) y la acumulación de polutantes extraídos por los sistemas de monitoreo (MAATE, 2011). En la Tabla 1 se muestra los rangos y niveles de la calidad del aire existentes en la atmósfera.

Tabla 1. Interpretación del índice de la calidad del aire.

Nivel	Bueno (ug/m3)	Alerta (ug/m3)	Alarma (ug/m3)	Emergencia (ug/m3)
Dióxido de carbono	0-14999	15000	30000	40000

CO₂				
Ozono	0-199	200	400	600
O₃				
Dióxido de nitrógeno	0-999	1000	2000	3000
NO₂				
Dióxido de azufre	0-199	200	1000	1800
SO₂				
Material particulado	0-249	250	400	500
PM₁₀				
Material Particulado	0-149	150	250	350
PM_{2.5}				

Obtenido de: MAAE-Acuerdo Ministerial 097-A, 2015

4.2.3. Contaminantes

Según la NECA, existen varios polutantes que inciden en la contaminación atmosférica los cuales son fundamentales para evaluar la calidad del aire (Secretaria del ambiente, 2019). Cabe mencionar que si los valores de los gases se lo encuentran en partículas por millón (PPM); para convertir a microgramos sobre metro cúbico se utiliza lo siguiente: 1ppm = 1000 ug/m³. A continuación, se detallará cada uno de ellos.

Material Particulado (PM). Es una mezcla de partículas líquidas y sólidas presentes en el aire de diferente composición y tamaño; es por esta razón, que se clasifican en: PM2.5 y PM10. Las más dañinas para la salud son las menores a 10 micrómetros de diámetro (PM10) ya que pueden penetrar y alojarse en el aparato respiratorio causando una disminución de las funciones pulmonares en niños y adultos con problemas de asma.

Dióxido de Azufre (SO₂). Es un gas irritante e incoloro generado en su mayoría por la quema de fósiles y procesos industriales de combustión. Las personas más vulnerables pueden sufrir de opresiones en el pecho o dificultades para respirar, incluso sometidos a exposiciones cortas.

Dióxido de Nitrógeno (NO₂). Es un gas sumamente tóxico que resulta de la combinación del oxígeno y del Óxido Nítrico; además del transporte vehicular, las fuentes generadoras de este gas son similares a las del SO₂. La tos y la disminución de las funciones pulmonares se presentan principalmente en niños asmáticos debido a la inhalación constante de NO₂.

Ozono (O₃). Es un contaminante de la atmósfera terrestre producido por la reacción química de los gases emitidos por las fuentes en presencia de la luz solar. El O₃ es un gas muy peligroso por lo que una concentración elevada puede causar problemas al aparato respiratorio, provocando asma y afecciones pulmonares.

Dióxido de Carbono (CO₂). Es un gas denominado asesino silencioso porque, es un gas que no tiene olor, ni color, no produce irritación en ninguna vía respiratoria; es producido por la incompleta oxidación a causa del carbono durante el proceso de combustión. La intoxicación por CO, es la causa más común de muerte ya que está relacionada al envenenamiento por gases o vapores tóxicos para la salud humana. Otra causa que genera dióxido de carbono es la mala ventilación o la colocación de dispositivos eléctricos y motores en espacios cerrados o sellados herméticamente. A continuación, en la tabla 2, se muestra los valores fijados por la NECA de los contaminantes:

Tabla 2. Valores fijados por la NECA de los contaminantes

Contaminante	Tiempo de muestra
Material particulado	15 ug / m ³ de media anual
PM₁₀	50 ug / m ³ de media en 24 h
Material particulado	50 ug / m ³ de media anual
PM_{2.5}	100 ug / m ³ de media 24 h
Dióxido de azufre	125 ug / m ³ de media 24 h

NO₂	500 ug / m ³ de media de 10 min
Dióxido de nitrógeno	40 ug / m ³ de media anual
SO₂	200 ug / m ³ de media en 1 h
Dióxido de carbono	10000 ug / m ³ de media en 8 h
CO₂	30000 ug / m ³ de media en 1 h
Ozono	100ug / m ³ de media de 8 h
O₃	

Obtenido de: Secretaría del ambiente, 2019

4.2.4. Efectos en la Salud

De acuerdo con el artículo de (OYARZÚN G, 2010) titulado "Contaminación aérea y sus efectos en la salud", se han examinado los impactos de la contaminación del aire en la salud a través de modelos experimentales que comparan diferentes segmentos de la población, incluyendo individuos sanos y enfermos. De esta manera, se ha establecido que los efectos están vinculados a la duración de la exposición, la concentración del contaminante (dosis) y la susceptibilidad individual.

La cantidad de contaminante que una persona recibe está relacionada con la tasa de ventilación por minuto, como se ilustra en la Ecuación 1:

$$De = Con * texp * Vmin \quad Ec (1)$$

Donde:

De : Dosis efectiva

Con : Concentración

texp : Tiempo de exposición

Vmin : Ventilación minuto (volumen de gas inhalado por minuto)

Por otro lado, la cantidad de contaminante que recibe una persona se eleva en las siguientes situaciones:

- Aumento de temperatura y humedad
- Aumento de la altitud

- Aumento de la progesterona en el embarazo
- Estados febriles (fiebre)
- Ejercicio físico

Estos factores inducen un incremento en la frecuencia respiratoria, lo que a su vez resulta en una mayor entrada de contaminantes a los pulmones. Esto afecta no solo al sistema respiratorio, sino también a otros sistemas y órganos del organismo.

4.3. Red de Sensores Inalámbricos Para Monitoreo de la Calidad de Aire

Las Redes de Sensores Inalámbricos (WSN por sus siglas en inglés, Wireless Sensor Networks) consisten en una red inalámbrica que puede abarcar desde una escala pequeña hasta una de gran magnitud. Estas redes se pueden organizar de manera flexible y admiten múltiples configuraciones, con una amplia cantidad de nodos que poseen recursos limitados. La comunicación en una WSN se realiza a través de saltos múltiples utilizando canales de comunicación estrechos. Esta tecnología se caracteriza por tener un bajo costo de diseño y ser relativamente novedosa. Además, encuentra aplicaciones en diversos ámbitos, como el medio ambiente, la agricultura, la salud y los edificios inteligentes.

Las WSN se componen de dispositivos autónomos distribuidos en un área de interés con el propósito de monitorear diferentes parámetros físicos o ambientales, como temperatura, sonido, vibraciones, presión, movimiento o presencia de agentes contaminantes.

De acuerdo con (Santamaría Buitrago & Archila Córdoba, 2013), una Red de Sensores Inalámbricos (WSN) es un conjunto de nodos, o sensores, que están interconectados para transmitir señales de eventos que están siendo monitoreados o controlados, junto con información técnica relacionada. Estos nodos comparten uno o más canales para la transmisión de datos, empleando diversas tecnologías y protocolos inalámbricos, cada uno con un propósito específico definido por el usuario.

Los componentes esenciales de una WSN se basan en (Figura 2):

- **Sensores:** Varían en tipos y tecnologías, y tienen la función de recolectar información del entorno y convertirla en señales eléctricas.
- **Nodos de sensor:** Reciben los datos provenientes de los sensores a través de sus interfaces de entrada y transmiten esta información a la estación base.

- **Gateway:** Elementos de interconexión entre la red de sensores y una red TCP/IP, permitiendo la comunicación entre ambos sistemas.
- **Red inalámbrica:** Representa la conexión entre los sensores y el gateway, posibilitando la transmisión de datos de manera inalámbrica.

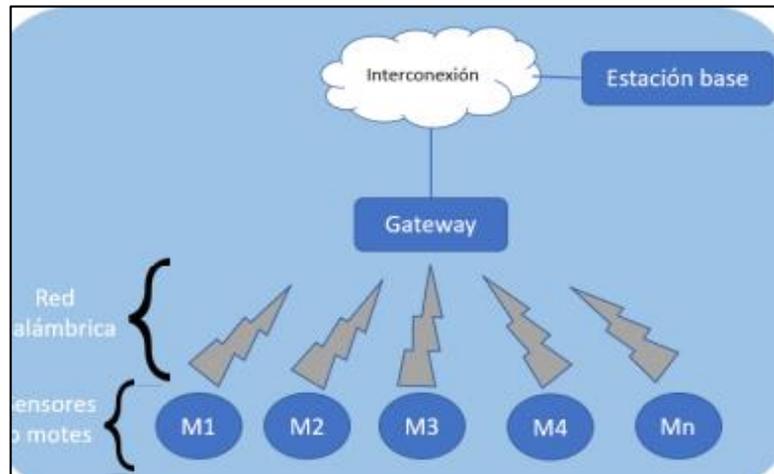


Figura 2. Red de sensores inalámbricos.

Fuente: Santamaría Buitrago & Archila Córdoba, 2013

4.3.1. Características de una WSN

Según (Ballester et al., 2009), en la tabla 3 se muestra las características principales de una WSN:

Tabla 3. Principales características de una WSN

Características	Definición
Tolerancia a fallos	Es posible que ciertos nodos sensores experimenten fallos o queden inoperativos debido al agotamiento de energía, daños físicos o influencias del entorno. Sin embargo, es fundamental que la eventual falla de nodos sensores no ponga en riesgo la operación integral de la red de sensores. Este concepto es conocido como principio de tolerancia a fallos o fiabilidad.
Escalabilidad	Los diseños actuales deben ser capaces de operar con una cantidad considerable de nodos, en el rango de cientos, miles o incluso, en función de la aplicación, millones. Además, es crucial que estos diseños consideren la alta densidad de nodos, ya que, en ciertas situaciones, pueden existir varios centenares de nodos sensores en una misma área geográfica.
Bajo consumo de energía	La red está formada por sensores de bajo consumo que generalmente dota a los dispositivos de una gran autonomía (típicamente de 5 a 10 años).

Costo de producción	Dado que las redes de sensores estarán compuestas por una gran cantidad de nodos, el costo de cada nodo individual se convierte en un factor fundamental para determinar la viabilidad económica de una red inalámbrica en comparación con una red cableada. Si el costo total de implementar la red inalámbrica es más elevado que el despliegue de sensores tradicionales, la justificación económica de la red de sensores puede ser cuestionable.
Funcionamiento desatendido	Esta tecnología funciona de manera autónoma sin una excesiva administración que demande de personal adicional
Conectividad ad-hoc	Generalmente son redes descentralizadas

Obtenido de: (Ballester et al., 2009)

4.3.2. Topologías de Red

Los nodos que componen la red pueden ser conectados de diversas maneras, y esta elección de configuración es un aspecto crucial que determina el desempeño y la capacidad de la red. La manera óptima de organizar estos componentes depende de diversos factores, incluyendo la cantidad de nodos involucrados, si son estáticos o móviles, entre otros (Segovia & Vidal, 2014).

En las WSN, se suelen emplear topologías específicas, algunas de las cuales se representan en la figura 3 que se presenta a continuación:

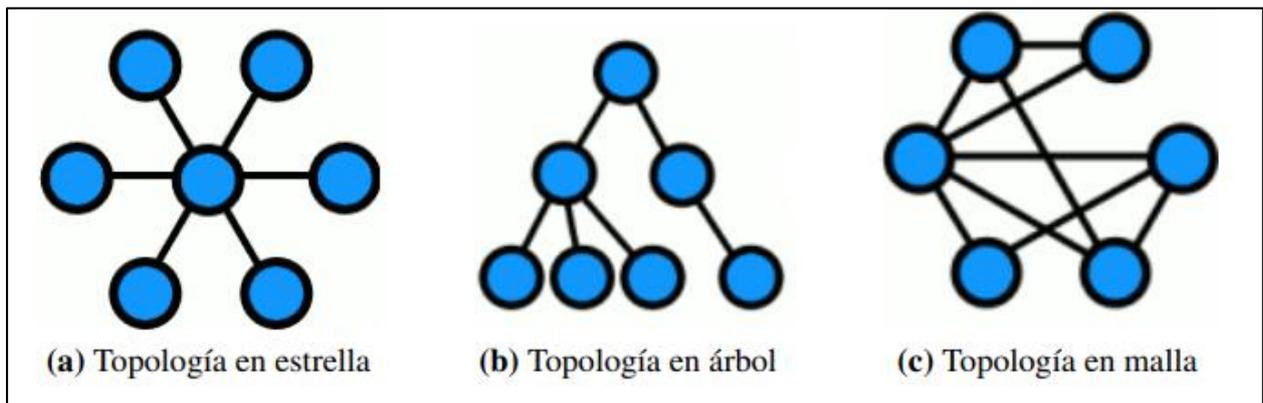


Figura 3: Topologías de Redes de Sensores.

Fuente: Segovia & Vidal, 2014

Nota: Existen varias topologías utilizadas en las Redes de Sensores Inalámbricos (WSN), cada una con sus propias características y aplicaciones específicas.

1. **Topología en estrella:** En esta topología, hay un nodo coordinador central y múltiples nodos terminales. Los nodos terminales solo pueden comunicarse con el coordinador. Si dos nodos terminales desean intercambiar información, deben hacerlo a través del coordinador. Esta topología se usa, por ejemplo, en Body Area Networks (BAN), que son redes para la monitorización de salud en el cuerpo humano.

2. **Topología en árbol:** En esta topología, hay un nodo raíz central que realiza funciones de coordinación. Además, existen routers que extienden el alcance de la red y nodos terminales. Los routers y el nodo raíz tienen nodos hijos, mientras que los nodos terminales son las hojas de este árbol y se comunican solo con sus nodos padres. Si un nodo padre se desconecta, los nodos hijos no pueden comunicarse con otros dispositivos en la red. Este diseño se utiliza, por ejemplo, con el protocolo CTP (Collection Tree Protocol).
3. **Topología en malla:** En esta topología, se encuentran el coordinador, routers y sistemas terminales. Se trata de una red multi-salto, lo que significa que los datos se enrutan a través de múltiples nodos para llegar a su destino. Cada nodo de la red puede comunicarse con cualquier otro nodo, lo que permite una mayor flexibilidad en la comunicación. Esta topología es especialmente útil cuando se desea ampliar el alcance de la red agregando más dispositivos.

Cada una de estas topologías tiene sus ventajas y desventajas, y la elección dependerá de las necesidades específicas de la aplicación y de las restricciones de diseño.

4.3.3. Ventajas y desventajas

Las WSN permiten la comunicación entre nodos sensores, Gateway y estaciones bases, dependiendo de la configuración los nodos podrían ser configurados para que se queden inmóviles si no están en uso, esto permitirá un tiempo de vida más largo, así como también bajo consumo de potencia, logrando mayor cobertura en su comunicación, costos accesibles, facilidad de instalación, tiempo de respuesta inmediata, precisión en sus aplicaciones, seguridad en la transmisión de datos y eficiencia en la red.

Efectivamente, en relación a las desventajas, es importante señalar que las redes inalámbricas necesitan disponer de fuentes de alimentación energética, y cada dispositivo individual dentro de la red requiere un consumo de energía significativo. Es relevante tener en cuenta que este nivel de consumo energético puede variar en función de la configuración específica de la red y las aplicaciones que estén en uso.

4.3.4. Fases de despliegue de las WSN.

Según (Moya, 2016), los sensores inalámbricos pueden ser instalados o dirigidos a diferentes puntos que se desea monitoreo, por ello se presente las siguientes fases:

a. Fase de pre-despliegue y despliegue.

Durante esta etapa, los sensores inalámbricos tienen la capacidad de ser desplegados de acuerdo a las necesidades específicas. Por ejemplo, podrían ser lanzados desde un avión en movimiento, incorporados en misiles o armas de fuego, o posicionados manualmente por individuos en ubicaciones específicas. También pueden ser integrados en objetos en movimiento, como robots.

b. Fase post-despliegue.

Después de la distribución inicial de los sensores, la topología de la Red de Sensores Inalámbricos (WSN) puede experimentar variaciones debido a factores ambientales como el ruido, interferencias, terremotos y otros elementos. Es por esta razón que la topología de la red de sensores puede cambiar inesperadamente, en función de la configuración y las características particulares de los dispositivos utilizados.

c. Fase de re-desplazamiento adicional de nodos.

La necesidad de regresar a la etapa inicial puede surgir en función del entorno físico, la disposición de los componentes, el equipo utilizado, y otros factores. Esto se debe a que los sensores están susceptibles a posibles daños, degradación y alteraciones en sus características. Como consecuencia, es esencial sustituir cualquier sensor que muestre signos de deterioro, problemas en la instalación o en la configuración. Ignorar estas condiciones podría generar dificultades en el funcionamiento de la red.

4.3.5. Protocolos de Comunicación Inalámbrica para una WSN

Los protocolos utilizados en las Redes de Sensores Inalámbricos (WSNs) están representados en la figura 4, la cual combina las siguientes capas: física, enlace, red, transporte y aplicación. Además de esto, se incluyen los planos de administración de la energía, manejo de la movilidad y gestión de las tareas. Cada capa y plano dispone de un conjunto de protocolos que abarcan diversas operaciones. (Segovia & Vidal, 2014).

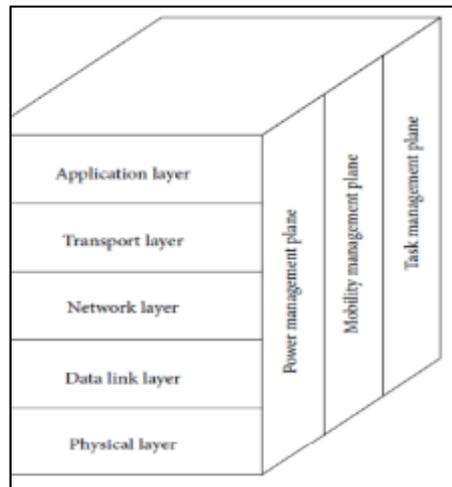


Figura 4: *Protocolos de una WSN.*

Fuente: *Segovia & Vidal, 2014*

Capa física: La capa física desempeña el papel de seleccionar la frecuencia adecuada, detectar señales, aplicar técnicas de modulación y llevar a cabo el cifrado de los datos.

Capa enlace: La capa de enlace tiene la responsabilidad de multiplexar flujos de datos, identificar tramas de datos, gestionar el acceso al medio y manejar errores. También se encarga de garantizar conexiones confiables punto a punto o punto a multipunto. El protocolo de acceso al medio (MAC - Medium Access Protocol) busca reducir al mínimo las colisiones con nodos vecinos, considerando que estas redes operan en entornos ruidosos y los nodos pueden estar en movimiento. Asimismo, se debe considerar la disponibilidad de energía de los dispositivos.

Capa de red: La capa de red se ocupa de determinar rutas para dirigir paquetes de un extremo a otro de la red. Al diseñar esta capa para WSN, se deben tomar en cuenta principios como una gestión eficiente de la energía, direccionamiento basado en atributos, conocimiento de la ubicación y la agregación de datos.

Capa de transporte: La capa de transporte ayuda a mantener el flujo de datos en caso de que la aplicación en la WSN lo requiera.

Capa de aplicación: La capa de aplicación administra y asigna tareas, y también se encarga de transmitir datos como activaciones y desactivaciones de nodos, sincronización de relojes, consultas sobre la configuración de la red de sensores, autenticación, distribución de claves, entre otros.

Administración de energía: La gestión de energía tiene como función principal supervisar el consumo de energía en un nodo. Por ejemplo, un nodo podría entrar en modo de reposo después de recibir un mensaje de su nodo vecino. También puede alertar a sus vecinos cuando tenga poca energía, de modo que no participe en los mensajes de enrutamiento.

Administración de movilidad: Esta función detecta y registra el movimiento de los nodos para que un nodo siempre tenga conocimiento de sus nodos vecinos y pueda mantener una ruta hacia la estación base. Saber quiénes son los nodos vecinos permite a un nodo equilibrar el uso de energía y las tareas a realizar.

Administración de tareas: Esta función equilibra y organiza las tareas de captura de datos en una región específica. Por ejemplo, puede determinar que no es necesario que todos los nodos en una región capturen datos al mismo tiempo. También puede asignar diferentes tareas a nodos según su nivel de energía disponible.

4.3.6. Estándares para WSN

Existen diversos protocolos en el ámbito de las Redes de Sensores Inalámbricos (WSNs), muchos de los cuales han sido estandarizados por diferentes organizaciones compuestas por actores de la industria (Candía & Varela, 2017). Las WSNs pueden emplear diversos protocolos para la transmisión y recepción de datos, y a continuación se presentan algunos de los más relevantes y ampliamente utilizados:

IEEE 802.15.4: Este estándar se originó en el grupo de investigación 802.15 de la IEEE, especializado en el desarrollo de tecnologías inalámbricas de área personal (WPAN). IEEE 802.15.4, cuya primera versión se publicó en 2003, define el nivel físico y el control de acceso al medio para redes inalámbricas de área personal con tasas bajas de transmisión de datos (LR-WPAN). La revisión actual del estándar se aprobó en 2011. Sirve como base para la especificación de ZigBee, 6LoWPAN y otros protocolos que se describen a continuación, y tiene como objetivo brindar una solución integral para este tipo de redes al construir los niveles superiores de la pila de protocolos que el estándar no abarca.

ZigBee: Este estándar define un conjunto de protocolos para la formación de redes inalámbricas de corto a medio alcance y baja velocidad de datos.

6LoWPAN: Desarrollado por la IETF (Internet Engineering Task Force), que es la organización responsable de los estándares de Internet para asegurar su interoperabilidad,

6LoWPAN actúa como una capa de adaptación entre la capa de red (IP) y las capas inferiores. Su objetivo es permitir la comunicación IPv6 con tecnologías inalámbricas de "baja potencia y comunicación con pérdidas" (LLN - Low Power and Lossy Networks según la IETF), utilizando IEEE 802.15.4.

Z-Wave: El protocolo Z-Wave [52] es una tecnología inalámbrica basada en comunicaciones de Radio Frecuencia diseñada específicamente para control, monitoreo y aplicaciones de lectura de estado en ambientes residenciales y comerciales. Se adapta a la automatización mediante redes de topología de malla bidireccional, de alcance limitado.

LoRa: La Lora Alliance estableció una especificación para redes LPWAN (low power wide area network) denominada LoRa (acrónimo de "Long Range"), diseñada para nodos inalámbricos con batería y baja tasa de transmisión de datos. En el modelo OSI, estas especificaciones conciernen a la capa física. La modulación de radio patentada por LoRa busca lograr comunicaciones de largo alcance (estimado entre 2 km y 5 km en entornos urbanos y hasta 15 km en áreas despejadas), con bajo consumo. A cambio, se reducen las velocidades de datos, que oscilan entre Kb/seg (desde 0.3 kbps hasta un máximo de 50 kbps).

SigFox: Fundada en 2009 en Francia, SigFox desarrolló una red inalámbrica de bajo consumo, baja tasa de datos y largo alcance (LPWAN), usando tecnología propietaria para comunicación de radiofrecuencia en bandas no licenciadas de 868 MHz y 915 MHz, modulando mediante tecnología UNB (Ultra Narrow Band).

LTE-M y NB-IOT: Dentro del grupo de LP-WAN, otros dos protocolos asociados a la infraestructura de redes celulares son LTE-M y NB-IOT, introducidos por la asociación 3GPP ("3rd Generation Partnership Project"), una entidad de telecomunicaciones encargada, entre otras cosas, de la evolución de protocolos para telefonía, desde GSM hasta GPRS, 3G y 4G-LTE.

4.3.7. Seguridad WSN

Existe una variedad de algoritmos que brindan seguridad a las redes de sensores inalámbricos, y en sus sistemas operativos se incorporan métodos de protección como el cifrado y la criptografía. Sin embargo, en el entorno de las WSNs no existe un método definitivo para eliminar por completo las vulnerabilidades y prevenir los ataques. Esto se debe a factores como las condiciones físicas de los nodos, que pueden facilitar el acceso no autorizado, y a la

transmisión inalámbrica de información, que la hace susceptible de ser interceptada. Como respuesta a esta problemática, se han introducido servicios de seguridad con el propósito de fortalecer y brindar mayor confiabilidad a las redes WSNs. Un ejemplo concreto es la aplicación de encriptación en tiempo real de los datos captados por los sensores, junto con sistemas de auto vigilancia y protocolos o algoritmos que contribuyan a la seguridad. (Moya, 2016).

4.3.8. Red General de Medición de la Calidad del Aire

Estas redes son implementadas con el propósito de evaluar el nivel de exposición de los residentes, los ecosistemas y las construcciones a los agentes contaminantes presentes en la atmósfera. Estas redes suministran información acerca de las tendencias de dichos contaminantes en una localidad. Es imperativo que la información recolectada sea representativa, lo que implica abarcar una amplia gama de zonas dentro de la localidad, que incluyan áreas industriales, comerciales y residenciales. En este contexto, las ubicaciones de las estaciones deben ser seleccionadas de manera que reflejen adecuadamente estas distintas zonas.

Cada estación debe estar conectada de manera online con un centro de control, transmitiendo los datos de manera automatizada a intervalos regulares. El centro de control se encarga de analizar los datos enviados y, si detecta algún problema, notifica al personal de mantenimiento para intervenir en la estación correspondiente. Además, el centro de soporte revisa los datos en su estado crudo que llegan desde las estaciones y, si es necesario, identifica datos sospechosos, marcándolos para un análisis posterior.

Las estaciones de nivel de fondo, colocadas en las periferias urbanas, tienen como objetivo estudiar el comportamiento de los contaminantes. Esto se logra al medir sus concentraciones y observar sus trayectorias, las cuales son influenciadas por las condiciones del viento en términos de velocidad y dirección. Los datos obtenidos de estas estaciones se manejan de manera similar a los de las estaciones que monitorean la calidad del aire ambiente en general.

4.4. Redes Low Power Wide Area Networking (LPWAN)

La tecnología LPWAN se define como una red de amplio alcance y baja potencia, funcionando como un protocolo inalámbrico de transporte de datos que en la actualidad desempeña un papel fundamental en la implementación de IoT (Borja, 2021). Dentro del contexto de Internet de las Cosas (IoT), las LPWANs están emergiendo como un punto focal y están atrayendo considerable atención en la comunidad tecnológica. Estas redes se caracterizan por proporcionar conectividad inalámbrica de largo alcance, bajo consumo de energía y costos

reducidos, especialmente aptas para dispositivos alimentados por baterías. No obstante, las LPWAN también presentan restricciones notables en términos de ancho de banda, frecuencia de transmisión de mensajes y tamaño de paquetes de datos. Las tecnologías LPWAN tienen el potencial de brindar conectividad a un gran número de dispositivos, con un menor gasto energético, mayor alcance y menor costo en comparación con las redes móviles. Se anticipa que las LPWAN tendrán un impacto significativo en las comunicaciones de la IoT.

4.4.1. Características

- **Cobertura extensa:** La distancia de comunicación entre dispositivos abarca desde distancias cortas en áreas urbanas hasta un poco más de 10 km en zonas rurales.
- **Eficiencia energética:** Las redes están optimizadas para reducir el consumo de energía. Los módulos de comunicación LPWAN, conocidos como transceptores, pueden funcionar durante varios años con baterías pequeñas y económicas.
- **Coste reducido:** Gracias a protocolos de bajo consumo y amplio alcance, junto con topologías en estrella o malla, se disminuye la complejidad en el diseño de hardware y la infraestructura de red. Esto se traduce en menores costos de implementación.
- **Tasa de transmisión y ciclo de operación:** El entorno de implementación de las redes LPWAN se caracteriza por cambios graduales en las mediciones a lo largo del tiempo. Esto permite reducir la tasa de transmisión de la red y el ciclo de operación de los nodos sensores a aproximadamente el 1%.
- **Escalabilidad:** Las redes LPWAN deben ser adaptables al crecimiento. Por lo general, estas redes experimentan un aumento significativo de dispositivos en un corto período. Como resultado, todos los componentes de la red, especialmente el gateway y el servidor, deben poder gestionar el aumento en el flujo de información o permitir la incorporación de nuevos dispositivos para mantener la escalabilidad

4.4.2. Aplicaciones de las LPWAN

Ofreciendo requisitos energéticos más reducidos, mayores alcances y costos inferiores en comparación con las redes móviles convencionales, las LPWAN brindan una variedad de posibilidades para aplicaciones de comunicación entre máquinas (M2M) e Internet de las Cosas (IoT), muchas de las cuales previamente se veían limitadas por restricciones presupuestarias y de energía (Sharon Shea, 2017). La elección de una tecnología LPWAN depende de la aplicación específica, considerando aspectos como la velocidad requerida, la cantidad de datos y el área de cobertura. Las LPWAN son especialmente apropiadas para aplicaciones que

demandan una entrega infrecuente de mensajes de enlace ascendente con datos de menor tamaño. La mayoría de las tecnologías LPWAN también presentan capacidades de enlace descendente. Estas tecnologías se emplean ampliamente en diversos campos, tales como la medición inteligente, la iluminación eficiente, el rastreo de activos, la gestión de ciudades inteligentes, la agricultura de precisión, el seguimiento de ganado, la administración energética, la manufactura y las implementaciones industriales de IoT.

4.4.3. Requisitos de la Aplicación LPWAN

La amplia gama de las aplicaciones de LPWAN da como resultado una amplia variación de las características como: cobertura, capacidad, costo, baja potencia, entre otras. Los dispositivos pueden abarcar una cobertura que va desde distancias muy cortas a distancias remotas, desde posiciones estacionarias a móviles, desde conexiones basadas en batería de baja potencia hasta conexiones comerciales basadas en energía, y una gama de entornos amigables a hostiles. El rango de cobertura para algunas aplicaciones está altamente localizado, por ejemplo, implementaciones estacionarias en interiores. Las aplicaciones que implican la movilidad de dispositivos, como el seguimiento de activos, requieren una cobertura de servicio global (Chaudhari et al., 2020).

4.4.4. Requisitos de tráfico

El mecanismo de comunicación intrínseco de las redes LPWAN involucra el flujo de datos originado por sensores distribuidos. Además de la posible existencia de tráfico generado por teléfonos inteligentes u otros dispositivos, el propio tráfico LPWAN puede variar en una amplia gama de características, como el número de mensajes, el tamaño de los mensajes y los niveles de confiabilidad requeridos.

En ciertas situaciones, podría ser necesario establecer una priorización en la programación de mensajes para transmisiones que se activan en respuesta a eventos específicos. Dado el elevado número de dispositivos en funcionamiento, existe la posibilidad de que los requisitos de los Acuerdos de Nivel de Servicio (SLA) de cada aplicación no puedan ser cumplidos en su totalidad. Por tanto, resulta esencial contar con mecanismos que permitan la coexistencia de distintos tipos de tráfico, garantizando la calidad de servicio (QoS) necesaria y la conformidad con los SLA.

En el contexto de aplicaciones LPWAN, podría ser imperativo implementar medidas que permitan gestionar múltiples categorías de dispositivos finales en función de sus respectivas demandas de comunicación en el enlace ascendente o descendente.

4.4.5. Capacidad y Densificación

Uno de los requisitos fundamentales de las tecnologías LPWAN consiste en su capacidad para gestionar de manera simultánea una extensa cantidad de dispositivos conectados, operando a velocidades de transmisión de datos reducidas. Numerosas aplicaciones demandan la capacidad de ser compatibles con más de 100,000 dispositivos, de manera que esta compatibilidad pueda expandirse de manera gradual y efectiva. Esta cualidad de escalabilidad hace referencia a la habilidad de expandirse sin dificultades, transitando desde una red que inicialmente incorpora un número limitado de dispositivos diversos hasta alcanzar un punto donde la red englobe una amplia variedad de dispositivos, así como también nuevos dispositivos, aplicaciones y funciones, sin comprometer en ningún momento la calidad y la disponibilidad de los servicios que ya están en funcionamiento (Mekki et al., 2019).

Dado que los dispositivos finales de las redes LPWAN cuentan con capacidades computacionales y de energía reducidas, los componentes de red, tales como las puertas de enlace y las estaciones de acceso, también desempeñan una función esencial en la mejora de la escalabilidad. La utilización de múltiples canales y diversas antenas, basadas en una gama variada de técnicas de diversidad, igualmente puede tener un impacto significativo en la escalabilidad de estas redes LPWAN. No obstante, es de suma importancia asegurar que estas características no tengan repercusiones negativas en otras métricas de rendimiento. El manejo de una gran cantidad de dispositivos también conlleva la densificación de la red. En este sentido, la red debe hacer frente a los desafíos derivados de la congestión en el acceso al medio y de las interferencias, evitando así cualquier degradación en el rendimiento de la red.

4.4.6. Operaciones energéticamente eficientes y fuentes de baja potencia

En diversos escenarios de aplicaciones de las redes LPWAN, las condiciones y restricciones del entorno no permiten el proceso de recarga de las baterías. Se prevé que la duración de la batería exceda los diez años sin necesidad de carga. Por lo general, se emplean baterías de tipo AA o de formato de botón. En situaciones en las que la batería se descargue, podría no ser factible su reemplazo en lapsos cortos de tiempo. Es fundamental que el costo de las baterías sea asequible. Las redes LPWAN deben operar bajo un estricto y extremadamente bajo ciclo de trabajo para maximizar la vida útil de los nodos. Por consiguiente, la eficiencia operativa en términos de un consumo energético sumamente reducido se convierte en un requisito esencial para los dispositivos IoT/M2M que funcionan con alimentación por batería.

4.4.7. Cobertura

La gama de operaciones exige capacidades de comunicación tanto a distancias largas como cortas. Por lo general, las tecnologías LPWAN necesitan brindar comunicación de larga distancia, abarcando desde 10 a 40 km en entornos rurales o desérticos, y de 1 a 5 km en áreas urbanas, superando en +20 dB a las redes celulares heredadas (Akpakwu et al., 2017). Para aplicaciones orientadas al monitoreo y la recolección de datos, podría ser necesario ofrecer cobertura en lugares interiores de difícil acceso, como ubicaciones subterráneas o sótanos. Esta cobertura debe mantenerse congruente con expectativas de tasas de datos adaptables y tasas de error de datos controladas. La utilización de la banda de frecuencia sub-GHz facilita a la mayoría de las tecnologías LPWAN establecer comunicaciones sólidas y confiables, manteniendo un menor consumo energético, ya que las frecuencias más bajas en la banda sub-GHz presentan propiedades de propagación más favorables en comparación con las frecuencias de 2.4 GHz y superiores. Además, las tasas de modulación reducidas empleadas en LPWAN permiten una asignación de mayor energía por bit, extendiendo así la cobertura. Una tasa de modulación más baja también favorece la correcta demodulación de la señal por parte de los receptores.

4.4.8. Identificación de ubicación

La determinación de la ubicación de los dispositivos emerge como un requisito esencial. La precisión en la localización desempeña un rol crucial en aplicaciones como la logística y el seguimiento del ganado. El espectro de exactitud puede variar, abarcando desde unos cuantos centímetros hasta varios metros. Es indispensable garantizar el soporte para tareas de supervisión y seguridad, de manera que se puedan detectar incidentes anómalos como cambios en la posición de los dispositivos, a la vez que se facilite el nivel apropiado de autenticación. La asignación de ubicaciones se puede lograr empleando sistemas de posicionamiento global (GPS), sistemas análogos al GPS o aplicando algoritmos inteligentes con el apoyo de la infraestructura de la red.

4.4.9. Seguridad y privacidad

Los requisitos de seguridad para los dispositivos LPWAN son particularmente rigurosos debido a la considerable cantidad de dispositivos, sus vulnerabilidades inherentes y su diseño simplificado. Se hace imperativo brindar respaldo a los aspectos primordiales de autorización, autenticación, establecimiento de confianza, confidencialidad, salvaguarda de los datos y prevención del rechazo. La provisión de seguridad debe tener la capacidad de enfrentar ataques

de código malicioso (tales como gusanos), abordar las amenazas de pirateo en dispositivos y sistemas LPWAN, y contrarrestar acciones de espionaje, seguimiento y ataques de denegación de servicio (DoS) (Mekki et al., 2019). También reviste importancia proteger la identidad del dispositivo y la confidencialidad de su ubicación frente al público. Además, es fundamental que esta seguridad también englobe la transmisión de datos en ambas direcciones, según lo exijan diversas aplicaciones.

4.4.10. Rentabilidad

Las aplicaciones de tecnologías LPWAN presentan una notable sensibilidad tanto hacia los aspectos vinculados al dispositivo como a los costos operativos (conforme se presenta en la Tabla 4). Además de los requisitos convencionales de lograr bajos costos operativos e implementación en la red, la extensa cantidad de dispositivos implicados impone considerables restricciones en lo que respecta a los gastos, el funcionamiento y una necesidad primordial de minimizar el consumo de energía (tal como se ilustra en la Figura 5). La capacidad de actualizar el software sin modificar el hardware emerge como una característica esencial que debe ser respaldada. Adicionalmente, es esencial proporcionar compatibilidad con la escalabilidad, la instalación sencilla y un mantenimiento económico.

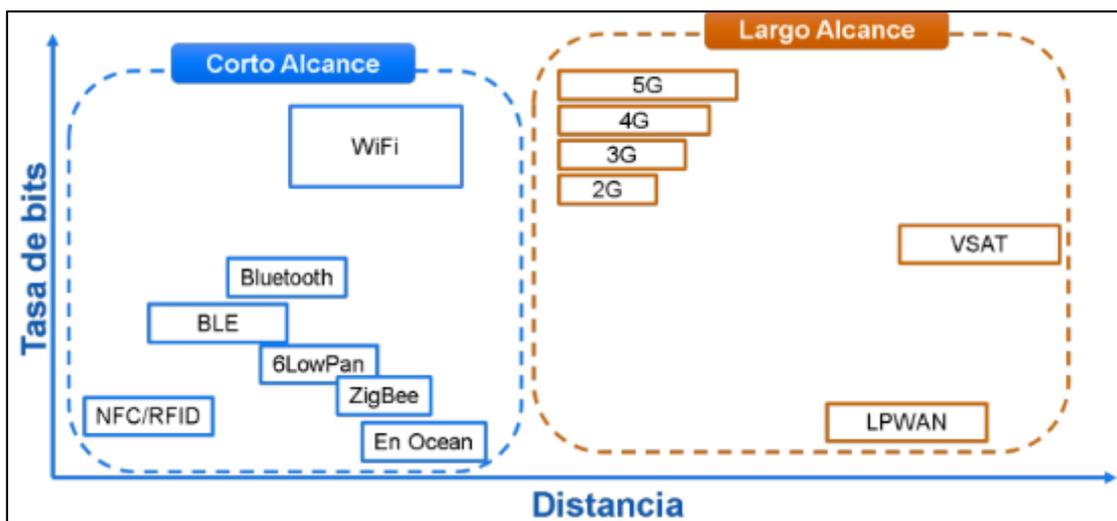


Figura 5. Comparativa de Tecnologías Inalámbricas.

Fuente: Porras Calderón & Salah García

Tabla 4. Comparativa de las Tecnologías LPWAN

Tecnologías	LoRa	Sigfox	LTE-M	NB-IoT
Modulación	CSS	BPSK	QPSK	QPSK
Frecuencia	Bandas ISM sin licencia	Bandas ISM sin licencia (868)	Bandas de frecuencia LTE con licencia	Bandas de frecuencia LTE con licencia y 2G

	(868 MHz en Europa, 915 MHz en América del Norte y 433 MHz en Asia)	MHz en Europa, 915 MHz en América del Norte y 433 MHz en Asia)		200
Velocidad de transmisión	50 Kbps	10 - 100 bps	1 Mbps	100 Kbps
Distancia de transmisión	5 Km (urbano) 20 Km (rural)	10 Km (urbano) 40 Km (rural)	1 Km (urbano) 5Km (rural)	1 Km (urbano) 10 Km (rural)
Inmunidad a interferencias	Muy alto	Muy alto	Muy alto	Bajo
Ancho de banda	500 KHz	100 KHz	200 KHz	150 Kbps
Disponibilidad	Today	Today	2016 standard	2016 standard
Duración de la batería	15 años	20 años	10 años	10 años
Costo	Medio	Medio	Alto	Alto
Longitud máxima de carga útil	243 bytes	12 bytes (UL), 8 bytes (DL)	27.2 (DL), 62.5 (UL)	1600 bytes
Autenticación y Encriptación	Si (AES-128)	No soportado	Si (LTE)	Si (LTE)
Tasa Adaptativa	Si	No	No	No
Comunicación en tiempo real	No	No	Si	No
Localización	Si (TDOA)	Si (RSSI)	Si	No (bajo especificación)
Permite red privada	Si	No	No	No
Estandarización	LoRa – Alliance	La compañía Sigfox está colaborando con ETSI en la estandarización de la red basada en Sigfox	3GPP	3GPP

Obtenido de: Hernández, 2020

4.4.11. Elección de la tecnología LPWAN

LoRa ha emergido como una de las tecnologías de mayor prominencia en el ámbito de las LPWAN. Esto se atribuye a sus capacidades de comunicación bidireccional, a la facilidad de interoperabilidad entre sensores sin la necesidad de configuraciones complejas, así como a la libertad que brinda a usuarios, desarrolladores y empresas para llevar a cabo la expansión del Internet de las Cosas (IoT) (como se presenta en la Figura 6) (Porrás Calderon & Salah Garcia, 2019).

Al considerar la elección de la tecnología de comunicación más adecuada para implementar en el contexto del IoT, es fundamental sopesar dos características principales: el coste y la velocidad, y lograr un equilibrio entre ambas. Además, se deben tener en cuenta parámetros

tales como la calidad del servicio, la vida útil de la batería, la latencia, la escalabilidad, la longitud de la carga útil, la cobertura, el alcance, la implementación y el coste, como se detalla en la Tabla 5.

Tabla 5. Comparativa de las Tecnologías LPWAN en Términos IoT

	LoRa	Sigfox	LTE-M	NB-IOT
QoS	Media	Media	Alta	Muy Alta
Batería	20 años	15 años	10 años	10 años
Latencia	Alta	Alta clase A y B Baja en clase C	Baja	Baja
Escalabilidad	Baja	Baja	Alta	Alta
Rango	Alto	Media	Bajo	Bajo
Implementación	Medio	Alto	Baja	Baja
Coste	Bajo	Bajo	Alta	Alto

Obtenido de: Hernández, 2020

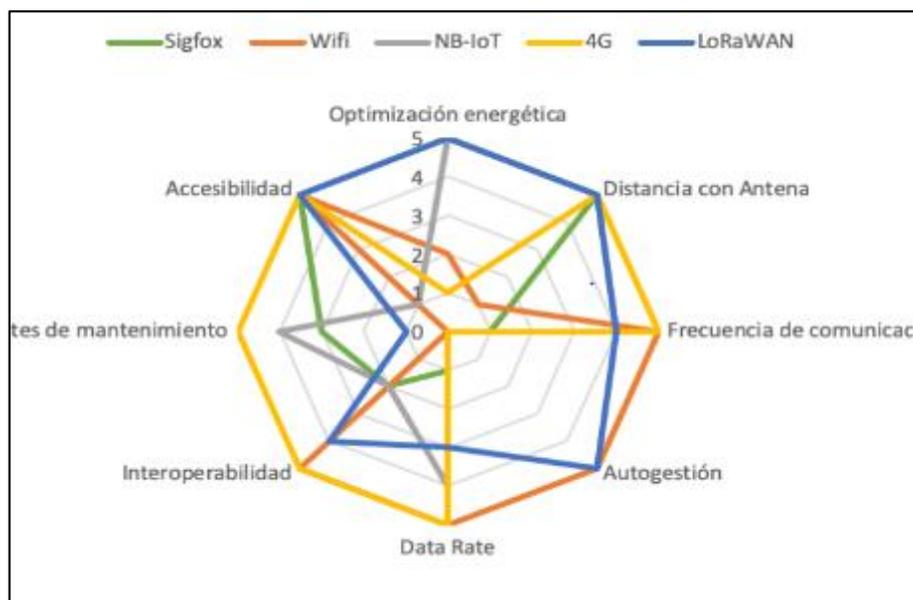


Figura 6. Gráfica Comparativa.

Fuente: Campos, 2020

Basándonos en lo expuesto previamente, podemos deducir que LoRa exhibe beneficios en relación a la duración de la batería, el proceso de implementación y el costo. En contraste, Sigfox presenta ventajas en cuanto a cobertura, duración de la batería y costos. Por otro lado, NB-IoT y LTE-M proveen ventajas en aspectos como la Calidad de Servicio (QoS), la latencia y la escalabilidad.

4.5. LoRa y LoRaWAN

4.5.1. LoRa

La tecnología LoRa se caracteriza por emplear una forma de modulación en la capa física basada en el concepto Chirp Spread Spectrum (CSS), el cual consiste en utilizar una señal de tipo chirp. Este enfoque implica aumentar (mediante "upchirp") o disminuir (mediante "downchirp") la frecuencia de una señal sinusoidal durante un intervalo de tiempo, abarcando la totalidad del ancho de banda asignado al canal. Esto conlleva un incremento en la resistencia a las señales de interferencia y a su vez disminuye la atenuación de la señal debido a los efectos de propagación multicamino. Al tratarse de un estándar LPWAN, es factible reducir la tasa de transferencia de datos, siempre y cuando se pueda lograr una cobertura más extensa en la comunicación. La característica de bajo consumo energético y su alcance significativo contribuyen a que LoRa se convierta en uno de los protocolos más ampliamente utilizados en el ámbito del desarrollo de aplicaciones de Internet de las Cosas (IoT). Esto permite que los dispositivos empleados en estos contextos industriales puedan extender su vida útil de manera considerable.

4.5.2. LoRaWAN

LoRaWAN es una arquitectura de red inalámbrica de largo alcance, destinada a abarcar áreas extensas. Los protocolos desarrollados para este tipo de red son de reciente creación y se aplican en diversos contextos. Esta estructura está especialmente concebida para dispositivos de bajo consumo energético, que encuentran su aplicación principal en entornos relacionados con el Internet de las Cosas (IoT) y el Internet Industrial de las Cosas (IIoT). En una red LoRaWAN, se implementa una topología denominada "Estrella de Estrellas" (Star of Stars), compuesta por cuatro componentes principales: dispositivos finales (nodos), gateways, un servidor de red y un servidor de aplicación. Los dispositivos finales, comúnmente conformados por sensores o actuadores, comparten información adquirida con el gateway utilizando la capa física LoRa a través de nodos o módulos. Por su parte, el gateway recibe esta información y la comunica al servidor por medio de un protocolo IP, como redes WiFi o redes móviles celulares. Este protocolo posee características que permiten recolectar datos de múltiples nodos ubicados a diferentes distancias en un solo gateway, incluso a lo largo de varios kilómetros, a través de una comunicación unidireccional (Heredia & Lucero, 2021).

4.5.2.1. Ventaja de LoRaWAN

- Uno de los beneficios primordiales reside en su aptitud para ser instalada en zonas donde no exista cobertura de otras redes o donde la misma sea insuficiente. Al ser una red interna, se evita la necesidad de pagar arrendamientos mensuales, aunque el mantenimiento deberá ser gestionado por el usuario.
- El valor más considerable que motiva la elección de LoRaWAN radica en su considerable rango de cobertura y su eficiencia en el consumo de energía.
- Otro aspecto favorable se encuentra en la ya incorporada encriptación de seguridad en su estructura.

4.5.2.2. Desventajas de LoRaWAN

- Puede resultar limitado en su utilidad si se necesita un ancho de banda superior al que puede manejar, lo que impide la transmisión de contenidos multimedia a través de esta red.
- Otro aspecto desfavorable podría ser el gasto asociado a su implantación, dado que es importante recordar que todo el equipo necesario para establecer esta red deberá ser adquirido, instalado y configurado por parte del usuario.

4.5.2.3. Clases de LoRaWAN

De acuerdo con (Bankov et al., 2017), se establecen tres categorías para los nodos LoRaWAN, cada una de ellas con distintos atributos que posibilitan al usuario llevar a cabo distintos tipos de comunicación basados en el consumo energético. A continuación, se detallan las características de cada una de las clases definidas:

Clase A: Esta es la categoría más común y es compatible con la mayoría de los módulos o nodos. Facilita la comunicación bidireccional, donde la transmisión Downlink ocurre en intervalos específicos, lo que resulta en un menor consumo de energía. Esta clase se recomienda especialmente para dispositivos alimentados por batería.

Clase B: Similar a la Clase A, la Clase B también permite la comunicación bidireccional. La distinción radica en que los intervalos de recepción están programados según un horario determinado por el gateway. Esto incrementa el consumo de energía del nodo en comparación con la Clase A.

Clase C: En esta categoría, el nodo permanece en constante escucha, siempre que no esté transmitiendo. Esto conlleva a un mayor consumo de energía, no obstante, ofrece la latencia más baja en cuanto a Downlink comparada con las clases previas.

4.5.2.4. Arquitectura.

La arquitectura LoRaWAN se sustenta en un modelo Star of Stars, el cual asegura una comunicación eficiente entre dispositivos, además de promover un bajo consumo energético en los dispositivos finales. Esta topología está compuesta por cuatro elementos principales: dispositivos finales, también conocidos como nodos o motas, gateways, un servidor de red y un servidor de aplicación. Los nodos desempeñan la tarea de recolectar datos provenientes de diversos tipos de sensores utilizados en estas aplicaciones y posteriormente transmiten la información a los gateways, siguiendo las pautas establecidas por el protocolo LoRaWAN. La función de los gateways consiste en transmitir la información sin procesar, obtenida de cada nodo, hacia un servidor de red, a través de una interfaz de back-haul con alto rendimiento (throughput $\geq 100\text{Mbps}$, delay $\leq 200\text{ ms}$) como Ethernet, 3G/4G, satélite o WiFi. Finalmente, el servidor de red recibe los paquetes de datos enviados por los nodos, los descifra y aplica protocolos de seguridad, permitiendo que las distintas aplicaciones accedan a los datos desde el servidor de red y los utilicen según sus necesidades (Casals et al., 2017). La Figura 7 ilustra cada uno de los componentes que conforman esta arquitectura mencionada.

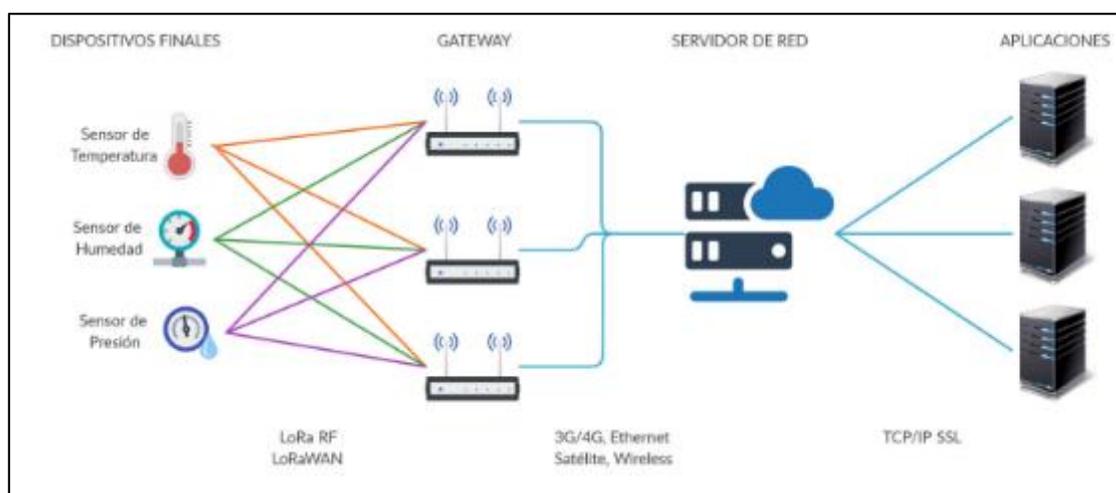


Figura 7. *Arquitectura LoRaWAN.*

Fuente: *Cheong et al., 2017*

4.6. Plataforma IoT

Una plataforma de Internet de las Cosas (IoT) puede abarcar desde soluciones sencillas que almacenan datos y proporcionan interfaces estándar al usuario, hasta sistemas más completos que posibilitan el empleo de herramientas para pronósticos, análisis o la creación de interfaces más sofisticadas.

En esencia, una plataforma de IoT debe ser capaz de recopilar los datos enviados desde diversos dispositivos interconectados. Además, debería simplificar la creación de aplicaciones, ya sean para dispositivos móviles u otros sistemas, con el propósito de representar de manera coherente los datos provenientes de los dispositivos IoT conectados a la plataforma. Esto incluye tanto la información cruda de los dispositivos como los datos que han sido procesados (Belmonte, 2019).

4.6.1. Ubidots

Ubidots es una plataforma de Internet de las Cosas (IoT) que presenta una infraestructura basada en nubes y abarca dispositivos de alcance global. Se destaca por ser una opción asequible y confiable, integrable en el conjunto de ecosistemas de plataformas IoT. Ubidots se especializa en soluciones combinadas de hardware y software para la monitorización, control y automatización de procesos a distancia en campos como salud, energía, industria, manufactura, servicios públicos y transporte. Como adición, recientemente ha introducido Ubidots for Education, una plataforma destinada a estudiantes que les permite crear, desarrollar, experimentar, aprender y explorar las perspectivas futuras de aplicaciones y soluciones conectadas a Internet (Figura 8).

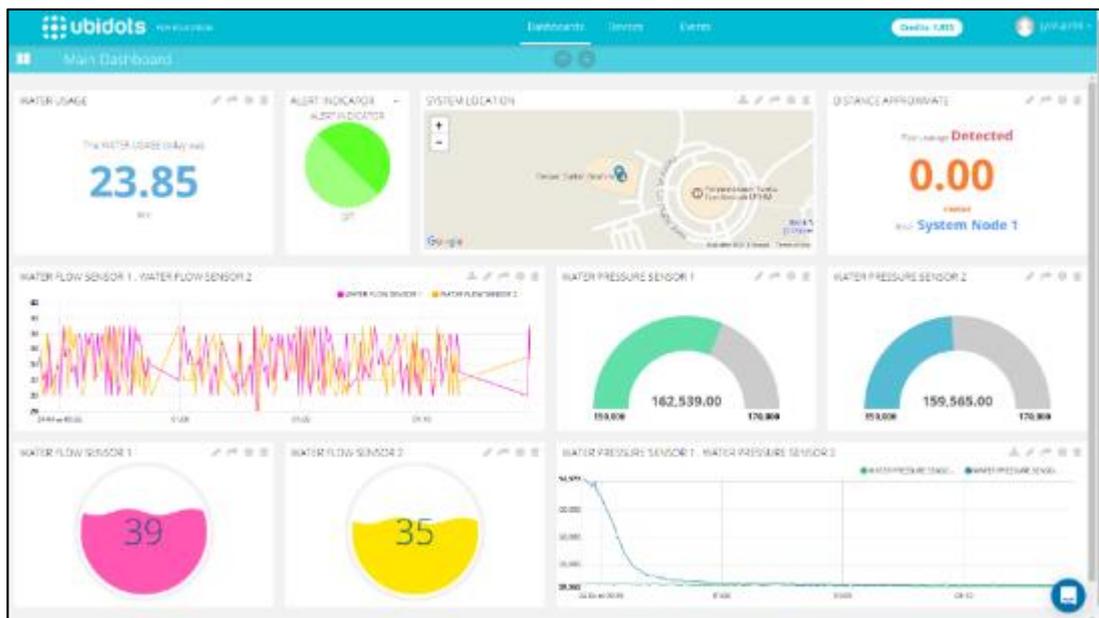


Figura 8. Ubidots.

Fuente: Ubidots, 2021

4.6.1.1. Características

- Establecimiento de una conexión entre el hardware y la nube mediante una variedad de bibliotecas, kits de desarrollo de software (SDK) y guías instructivas.

- Automatización de la configuración de variables, propiedades y aspecto de los dispositivos para facilitar la replicación del proceso en dispositivos nuevos.
- Adaptación personalizada de la interfaz de programación de aplicaciones (API).
- Ampliación de la capacidad de supervisión y análisis de los datos de las aplicaciones mediante la integración de APIs.
- Transformación de datos intrínsecos en información significativa a través de la creación de variables sintéticas.
- Creación de paneles de control en tiempo real para el análisis de datos y la gestión de dispositivos.
- Simplificación del proceso de compartir datos mediante enlaces públicos o la incorporación de paneles de control o widgets en aplicaciones web privadas y móviles.
- Activación de comandos de "interrupción" o "reinicio" cuando el hardware permanece inactivo durante un período prolongado.
- Asignación de permisos y restricciones a todos los usuarios que interactúan con los paneles de control, dispositivos y/o eventos.

4.6.2. *ThingSpeak*

ThingSpeak representa una plataforma de software que proporciona un servicio de análisis fundamentado en la Internet de las Cosas (IoT), facilitando la recopilación, evaluación y representación directa de datos desde flujos en la nube. Este sistema posibilita la visualización inmediata de la información emanada por los dispositivos. Además, es viable ejecutar códigos MATLAB para el análisis y procesamiento en tiempo real de los datos al ser transmitidos. ThingSpeak es comúnmente empleado en la formulación de maquetas y pruebas conceptuales para sistemas de IoT que requieren un análisis de volúmenes de datos extensos (Ver Figura 9).

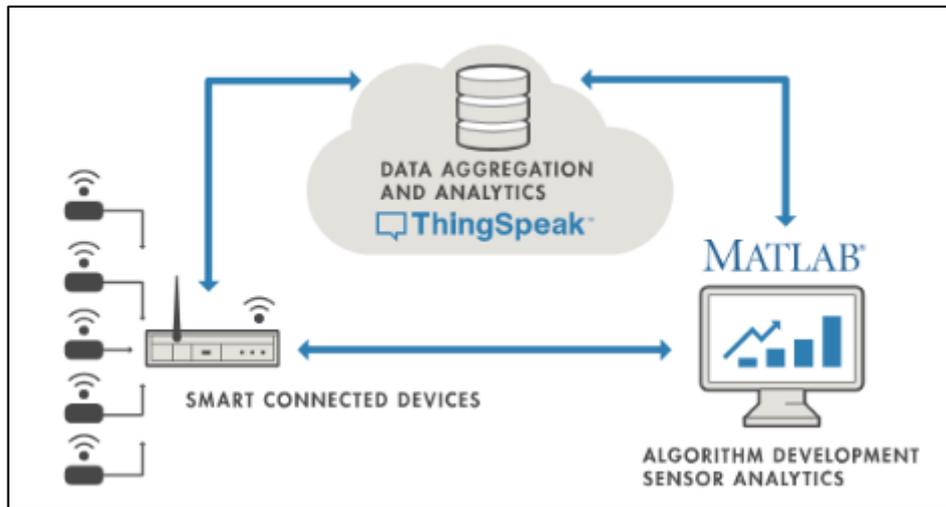


Figura 9. ThinkSpeak.

Fuente: ThinkSpeak, 2019

4.6.2.1. Características

- Varias de las características principales de ThingSpeak incluyen:
- Simple ajuste de los dispositivos que transmiten información a ThingSpeak.
- Observación en tiempo real de los datos recopilados de diversas sondas.
- Aprovechar la capacidad de MATLAB para el procesamiento de los datos.
- Elaboración de prototipos y creación de sistemas IoT sin requerir servidores ni elaboración de programas de software.
- Ejecución automática de acciones en función de los datos, posibilitando además la interacción con servicios como Twilio o Twitter.

4.6.3. *My Devices Cayenne*

Cayenne se destaca como una plataforma singular al ser el pionero en la construcción de proyectos de IO mediante la técnica de arrastrar y soltar (drag and drop), lo cual otorga a los usuarios la capacidad de crear prototipos de forma ágil y compartir sus proyectos de dispositivos interconectados. La propuesta de Cayenne radica en apoyar a los usuarios en la creación de prototipos fundamentados en la IoT y en llevarlos posteriormente a su producción. Adicionalmente, las aplicaciones móviles de Cayenne permiten la supervisión y el control remoto de los proyectos de IO a través de las aplicaciones de teléfonos inteligentes Android o iOS, así como en navegadores populares. Asimismo, mediante widgets personalizables, simplifica la visualización de datos, la configuración de reglas y la programación de eventos a través de su Tablero en Línea Cayenne (Ver Figura 10).

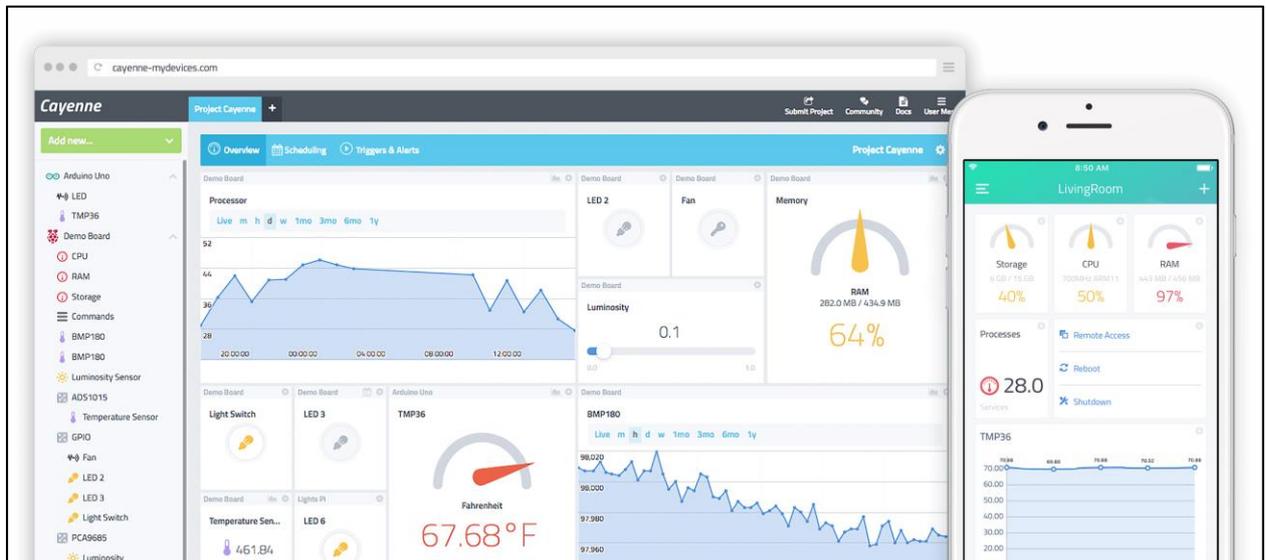


Figura 10. My Devices Cayenne.

Fuente: My Devices Cayenne, 2020

Características

- Proporciona un tablero personalizable que se puede adaptar con widgets de arrastrar y soltar, lo que brinda un control total sobre el proyecto.
- Presenta información de datos, estado y acciones de los dispositivos a través de los widgets.
- Habilita la configuración de mensajes de texto SMS y notificaciones por correo electrónico según los eventos que se desencadenen.
- Posibilita la administración remota y la creación de nuevos proyectos mediante las aplicaciones móviles.
- Automatiza tareas en pocos y sencillos pasos mediante sentencias If/Then poderosas basadas en datos y acciones en tiempo real.
- Programa eventos para computadoras, microcontroladores, sensores y actuadores interconectados.
- Acceso y observación en tiempo real, así como acceso a datos históricos de dispositivos y sensores.
- Rastreo basado en la ubicación de cualquier dispositivo conectado para obtener información sobre la posición, el estado y el historial de ubicaciones.
- Permite una conexión sencilla de dispositivos habilitados con LoRa para recopilar, transferir y mostrar información inteligente procesable.

4.6.4. Comparativa de plataformas Web

En la tabla 6, se describe y realiza una comparativa entre las diferentes plataformas web:

Tabla 6. Comparativa de Plataformas Web

Plataforma	SDK/ Lenguajes soportados	Protocolos soportados	Ventajas	Desventajas
Ubidots	Python, Java, C, PHP, Node.js, Ruby	MQTT, HTTP	Dispone de numerosas librerías de placas de desarrollo específicas y lenguajes de programación. Potente herramienta de representación gráfica de datos.	No permite la gestión de dispositivos por grupos. Carece de ofertas flexibles por uso.
ThinkSpeak	Matlab	MQTT, HTTP	La plataforma es de código abierto. Ofrece gran número de integraciones de redes sociales. Creación de prototipos de sistemas IoT sin crear servidores o desarrolladores softwares.	Es necesario utilizar un servicio de terceros para alertas que no sean de Twitter. Difícil configuración y sistemas poco intuitivo. Documentación limitada a HW.
My Devices Cayenne	Arduino, Raspberry Pi	MQTT	Plataforma capaz de gestionar directamente los conectores de las placas de desarrollo, evitando programar la interacción del sistema con los sensores. Documentación, muy extensa y con muchos ejemplos.	Limitado soporte para placas de desarrollo y sensores, ya que tienen que ser compatibles con el software proporcionado por la plataforma. Uso excesivo de su aplicación de su aplicación móvil para la configuración y gestión de los dispositivos IoT.

Obtenido de: Elaborado por el autor, 2023

La plataforma web seleccionada para la visualización y almacenamiento de los datos del presente proyecto es Ubidots, debido a sus diversas características que nos presenta de tal manera que nos permite obtener un correcto funcionamiento del envío y presentación de los datos y su facilidad de usar.

5. Metodología

La metodología a utilizar para llevar a cabo el desarrollo del presente Trabajo de Integración Curricular es el método experimental, por lo que, se procederá a la configuración de los dispositivos y la plataforma web, en base a las normativas nacionales e internacionales establecidas para cada una de las variables objeto de este trabajo

5.1. Investigación y análisis de los parámetros más importantes de la calidad del aire

La investigación se la llevó a cabo, mediante la revisión bibliográfica de fuentes confiables a nivel nacional e internacional con el fin de identificar cuáles eran los gases más contaminantes presentes en el aire.

Las directrices de la OMS sobre la calidad del aire ofrecen orientaciones a escala mundial sobre los valores límites de los contaminantes atmosféricos clave que entrañan riesgos para la salud. Estas directrices son de aplicación en todo el mundo se basan en una evaluación efectuada por expertos de las pruebas científicas actualmente disponibles sobre los siguientes gases:

- Material Particulado (PM)
- Ozono (O₃)
- Dióxido de Nitrógeno (NO₂)
- Dióxido de Azufre (SO₂)
- Dióxido de carbono (CO₂)

Los gases mencionados anteriormente tienen gran impacto en la salud de los seres humanos provocando una gran tasa de morbilidad, mortalidad y diferentes enfermedades (asma, síntomas bronquiales, insuficiencia respiratoria, cáncer, entre otras.), atribuida a la contaminación del aire ambiente.

5.2. Evaluar y caracterizar el por qué el uso de la tecnología LPWAN, LoRA para el desarrollo del proyecto, referente a su, aplicación, alcance, funcionamiento, arquitectura y protocolos.

En este apartado se realizó una investigación bibliográfica sobre las tecnologías LPWAN la cual nos brinda un sinnúmero de ventajas, partiendo de ello la tecnología LPWAN se desprenden tecnologías de comunicación inalámbrica que pueden ser utilizadas para la comunicación de los diferentes dispositivos. Para elegir la tecnología a utilizar se describió y comparo diferentes

características, destacándose lo siguiente: el alcance, consumo energético, ancho de banda, velocidad de transmisión, seguridad entre la más importantes, con el fin de elegir la apropiada para el desarrollo del proyecto, siendo esta la tecnología LoRA/LoRaWAN.

5.3. Diseño de una red de sensores basada en la tecnología LoRa

Se describirá de manera detallada el diseño, desarrollo e implementación de red de sensores IoT mediante LoRaWAN para el monitoreo de parámetros ambientales de Co, O3, PM, No3 y SO2.

5.3.1. Diseño de la red LoRaWAN

La red LoRaWAN utilizara una topología estrella, la cual está compuesta por dos nodos sensores y un Gateway, a continuación, se muestra el diseño gráfico de la red a implementarse (Figura 11).

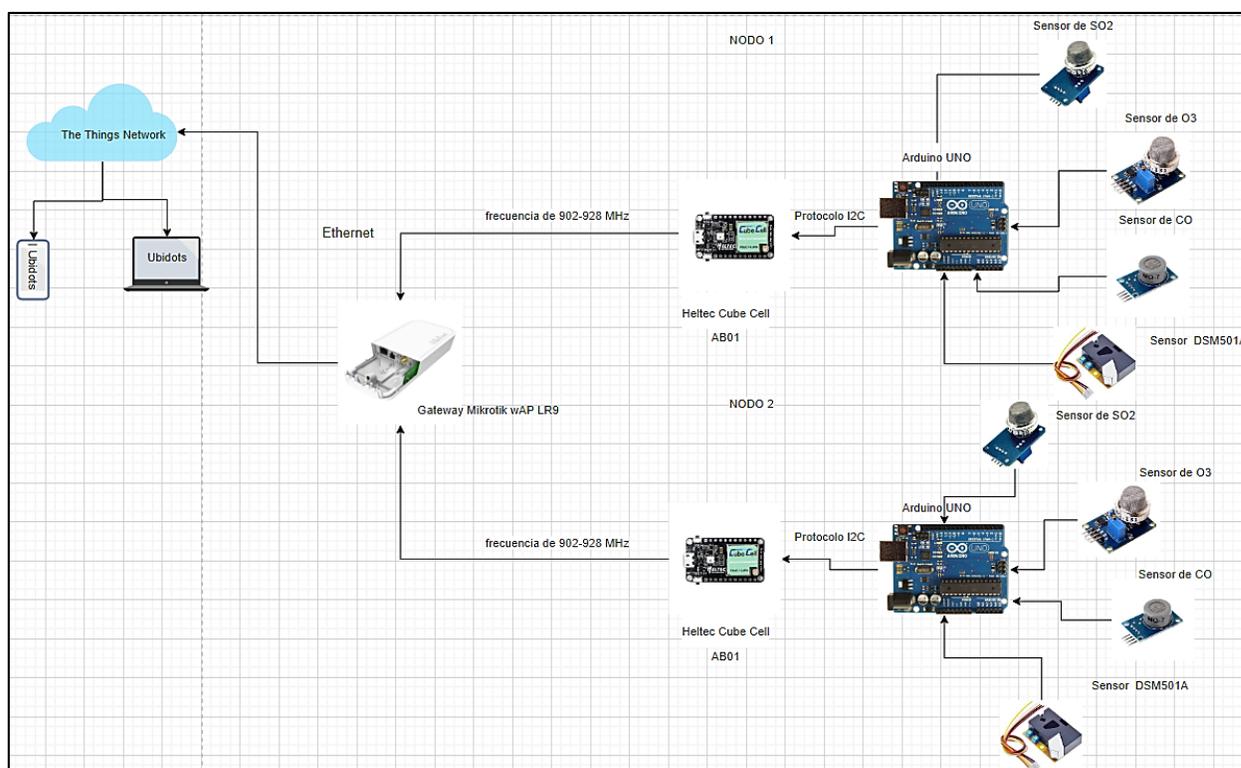


Figura 11. Red LoRaWAN para el Monitoreo de Gases Ambientales.

Fuente: Elaborado por el autor, 2023

5.3.2. Despliegue de la red LoRaWAN

La red IoT está compuesta por un Gateway Mikrotik wAP LR9, un arduino UNO, un módulo heltec cubecell y los sensores de la serie MQ para la medición de los parámetros CO, O₃, PM, NO₃ y SO₂.

5.3.3. Nodo Gateway

El nodo Gateway Mikrotik wAP LR9, también conocido como RBwAPR-2nD&R11e-LR9 es una solución lista para usar la tecnología LoRa. Este kit contiene un reenviador de paquetes UDP preinstalado a cualquier servidor LoRa público o privado y un punto de acceso inalámbrico impermeable al aire libre con interfaz WLAN de 2,4 GHz y puerto Ethernet que podría utilizarse como backend.

Los dispositivos hardware que componen al Gateway son: una tarjeta concentradora R11e-LoRa9, un adaptador de corriente de 24V/0.8A, un inyector PoE, cable adaptador automotriz de 0,35m y 4 pines, una antena interna de 2dBi (Figura 12).

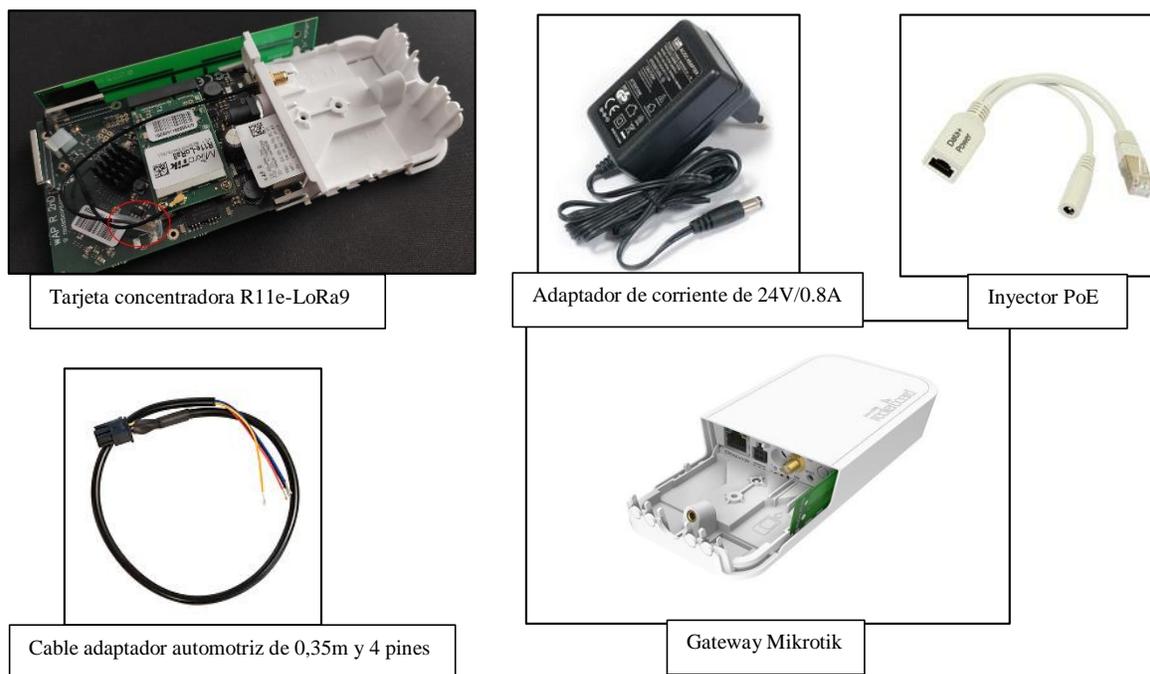


Figura 12. Materiales que Componen el nodo Gateway.

Fuente: Elaborado por el autor, 2023

Para poder configurar el Gateway, este por defecto crea un punto de acceso Wi-Fi (AP) con el nombre de *Mikro Tik-0115C4C*, a continuación, se detalla los pasos de su configuración:

1. Conectarse a la red, entrar al navegador y digitar <http://192.168.88.1>. El usuario siempre será *admin* y sin contraseña por defecto (Figura 13).

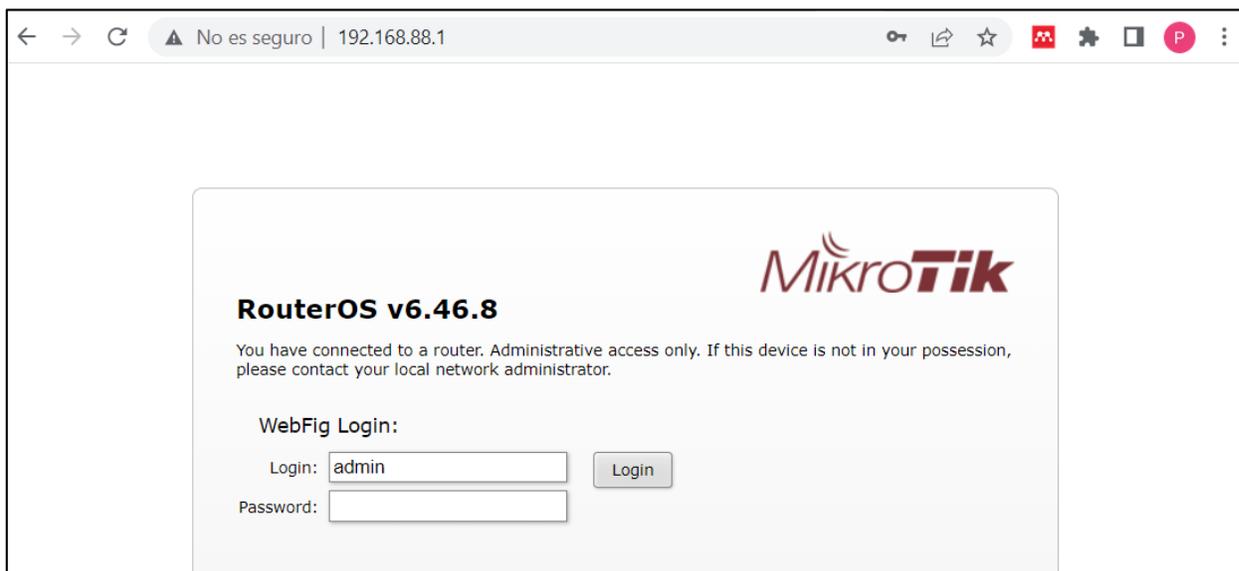


Figura 13. Inicio de sesión para ingresar al Gateway

2. A continuación, como se muestra en la figura 14, se procede a configurar los parámetros básicos de la red a través de *Quick set*.

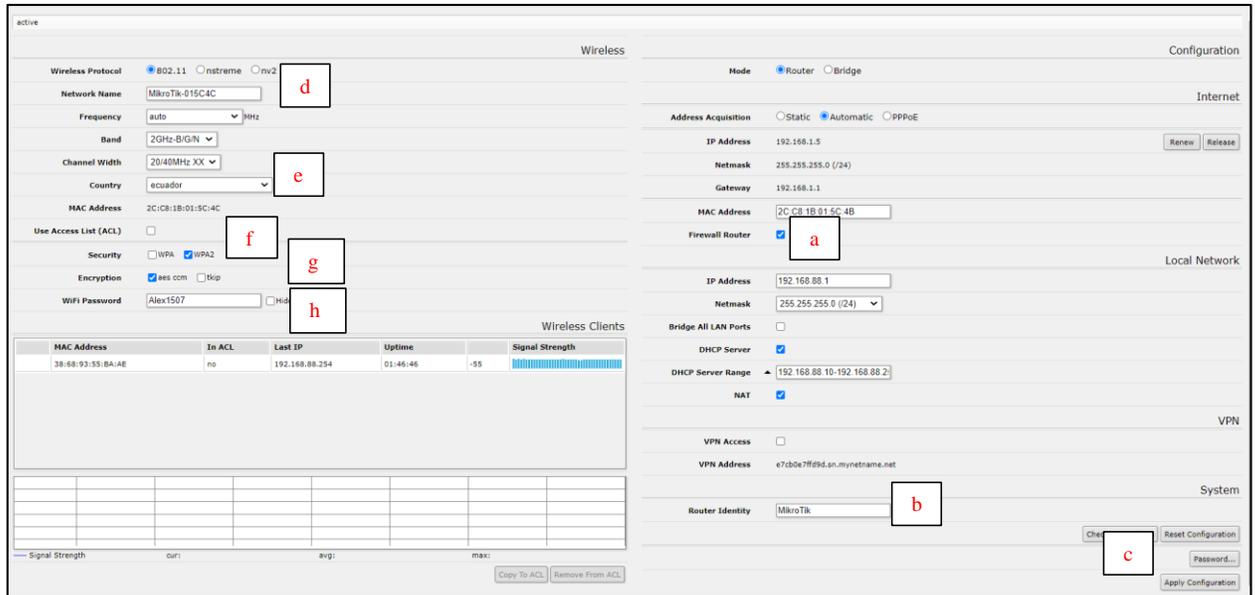


Figura 14. Configuración inicial "Quick Set"

- a) Firewall: la casilla debe estar desactivada para poder acceder al Gateway desde la LAN
- b) El nombre del Gateway se puede elegir libremente
- c) Cambiar la contraseña para el acceso administrativo
- d) SSID del punto de acceso Wi-Fi
- e) Configuración de país para la conexión WLAN
- f) Cifrado WPA2 (WPA ya no se considera seguro)
- g) AES como algoritmo utilizado
- h) Contraseña para la WLAN

5.3.4. Configuración para The Things Network

3. Se procede a configurar LoRa, para acceder a ella dar click sobre *WebFig* y a continuación sobre *LoRa* (Figura 15).

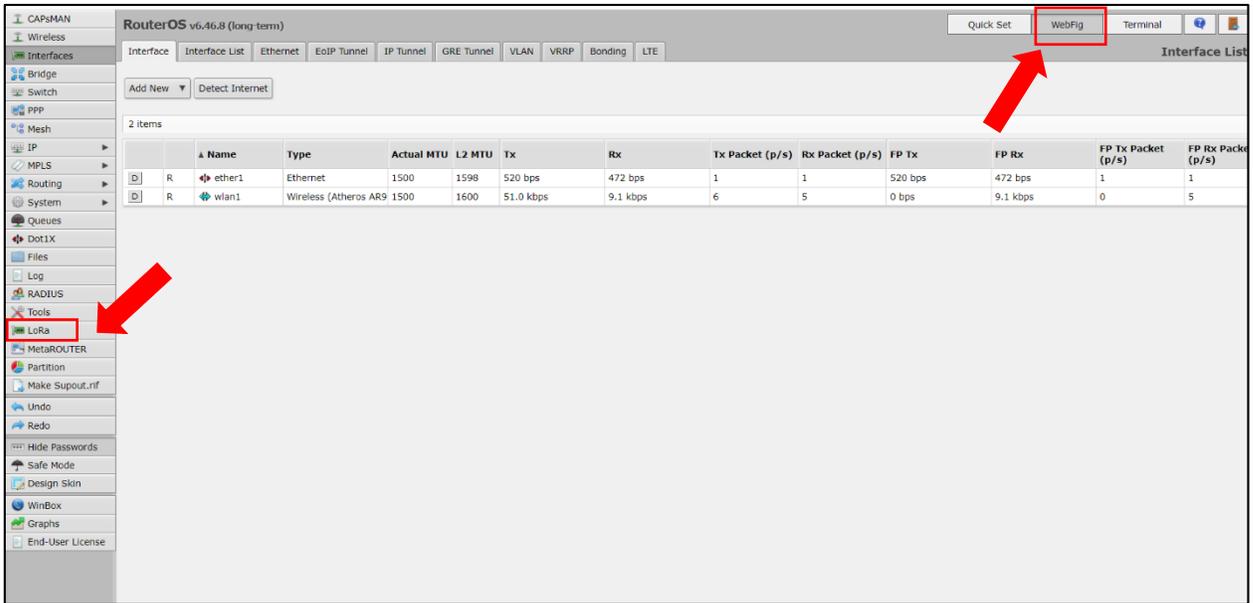


Figura 15. Configuración LoRa.

4. Agregamos un servidor de red y, para ello dar clic sobre *server*, y clic sobre *add new*, se rellena los campos y clic sobre *apply* (Figura 16).

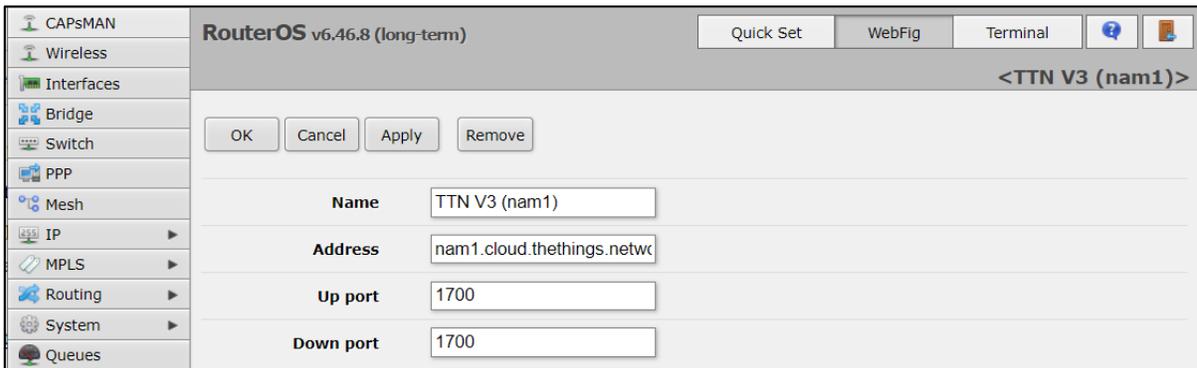


Figura 16. Configuración Server

5. Servidor de red LoRaWAN añadido (Figura 17).

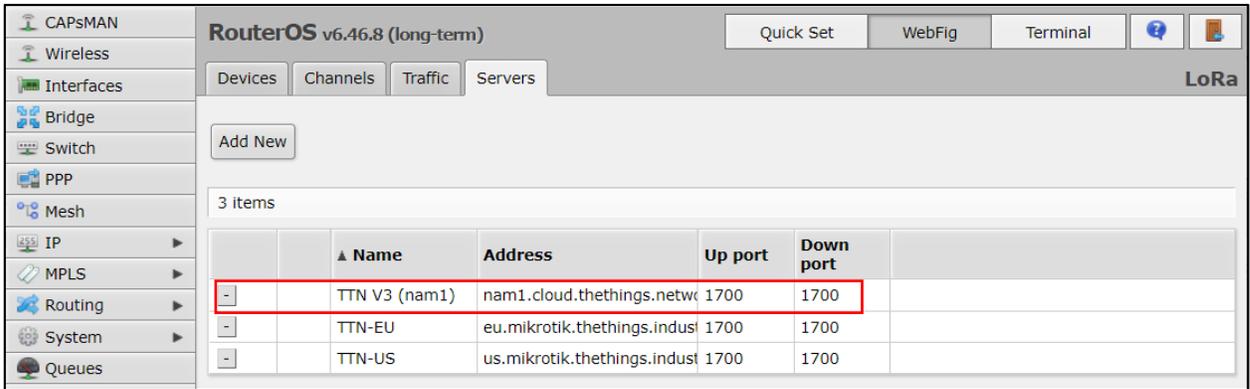


Figura 17. *Servidor de red LoRaWAN (LNS) añadido.*

- Una vez los servidores definidos los LNS, haz clic sobre la pestaña *Devices* y haz clic sobre el único dispositivo que aparece en la lista (Figura 18).

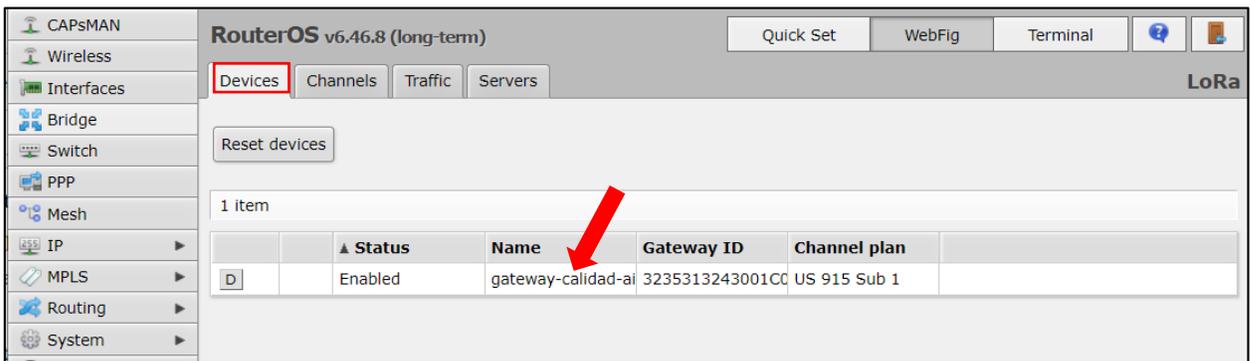


Figura 18. *Dispositivo LoRa (Concentrador)*

- A continuación, como se observa en la figura 19, aparecen las opciones que se pueden configurar.

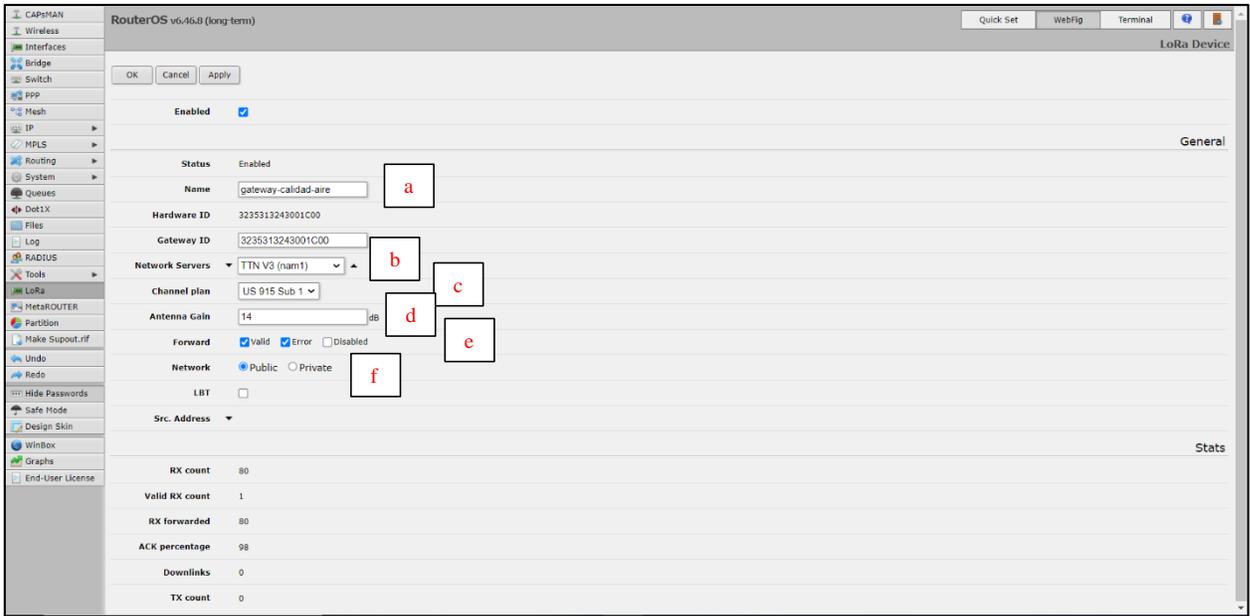


Figura 19. *Agrega o elimina distintos LNS a los que reenviar los paquetes LoRa*

- a) Define el nombre del concentrador LoRa
 - b) Agrega o Elimina (distintos LNS a los que reenviar los paquetes RF LoRa. Selecciona TTN V3 nam1 para utilizar The Things Network.
 - c) Se elige el plan de canales dependiendo de la región en la que te encuentres.
 - d) Configurar la ganancia de la antena (dB) a la que está conectado el Gateway. Con una antena potente, la potencia de transmisión debe regularse para *no superar los 14 dBm* de potencia de transmisión.
 - e) Selecciona los tipos de mensajes a reenviar
 - f) Declara la red como Pública o Privada.
8. Una vez configuradas las opciones, marcar la casilla de verificación *Enabled* para habilitar la tarjeta concentradora LoRa R11e-LR9 (Figura 20).



Figura 20. *Habilitar el Gateway*

9. En la pestaña *Traffic* se observa los mensajes entrantes (Figura 21).

Type	Time	Gateway ID	Message Type	Dev Addr	Freq (MHz)	Modulation	Bandwidth	Datarate	Coderate	CRC Status	RSSI (dB)	SNR (dB)
Rx	Dec/03/2022 18:56:01	3235313243001C00	Rejoin-request		902.300	LoRa	125 kHz	SF 7	4/6	Error	-96.00	-11.25
Rx	Dec/03/2022 18:55:01	3235313243001C00	Confirmed Data Up	BA 82 08 4E	902.500	LoRa	125 kHz	SF 7	4/5	Error	-101.00	-11.75
Rx	Dec/03/2022 18:52:41	3235313243001C00	Rejoin-request		903.100	LoRa	125 kHz	SF 7	4/6	Error	-103.00	-12.25
Rx	Dec/03/2022 18:51:21	3235313243001C00	Rejoin-request		903.100	LoRa	125 kHz	SF 7	4/8	Error	-102.00	-11.50
Rx	Dec/03/2022 18:47:21	3235313243001C00	Unconfirmed Data Up	99 D7 4E 48	903.000	LoRa	500 kHz	SF 8	4/7	Error	-96.00	-14.00
Rx	Dec/03/2022 18:45:21	3235313243001C00	Rejoin-request		903.300	LoRa	125 kHz	SF 7	4/5	Error	-103.00	-11.00
Rx	Dec/03/2022 18:43:51	3235313243001C00	Unconfirmed Data Down	12 2E DB 6B	902.500	LoRa	125 kHz	SF 8	4/7	Error	-97.00	-13.25
Rx	Dec/03/2022 18:43:21	3235313243001C00	Join-request		903.500	LoRa	125 kHz	SF 7	4/5	Error	-103.00	-11.75
Rx	Dec/03/2022 18:43:11	3235313243001C00	Unconfirmed Data Down	D0 23 70 16	902.700	LoRa	125 kHz	SF 7	-	Error	-99.00	-10.25
Rx	Dec/03/2022 18:42:41	3235313243001C00	Join-accept	E8 18 E3 A9	903.300	LoRa	125 kHz	SF 7	-	Error	-100.00	-12.00
Rx	Dec/03/2022 18:40:41	3235313243001C00	Join-request		903.500	LoRa	125 kHz	SF 7	-	Error	-103.00	-11.75
Rx	Dec/03/2022 18:40:41	3235313243001C00	Join-accept	80 7E B9 C9	903.500	LoRa	125 kHz	SF 7	-	Error	-103.00	-11.50
Rx	Dec/03/2022 18:40:41	3235313243001C00	Proprietary		903.100	LoRa	125 kHz	SF 7	4/7	Error	-103.00	-11.25
Rx	Dec/03/2022 18:40:41	3235313243001C00	Unconfirmed Data Down	2E 32 42 0D	903.700	LoRa	125 kHz	SF 7	-	Error	-103.00	-11.50
Rx	Dec/03/2022 18:35:41	3235313243001C00	Unconfirmed Data Down	88 C0 D4 EB	903.700	LoRa	125 kHz	SF 7	-	Error	-101.00	-11.75
Rx	Dec/03/2022 18:34:41	3235313243001C00	Rejoin-request		903.100	LoRa	125 kHz	SF 7	-	Error	-102.00	-11.50
Rx	Dec/03/2022 18:33:31	3235313243001C00	Confirmed Data Down	FB D4 F6 31	903.300	LoRa	125 kHz	SF 7	4/5	Error	-103.00	-10.75
Rx	Dec/03/2022 18:32:31	3235313243001C00	Unconfirmed Data Up	32 BC BB 2F	903.500	LoRa	125 kHz	SF 7	4/7	Error	-104.00	-12.00
Rx	Dec/03/2022 18:31:31	3235313243001C00	Rejoin-request		903.500	LoRa	125 kHz	SF 7	4/6	Error	-104.00	-11.25
Rx	Dec/03/2022 18:27:31	3235313243001C00	Proprietary		902.700	LoRa	125 kHz	SF 7	4/5	Error	-101.00	-11.75
Rx	Dec/03/2022 18:24:21	3235313243001C00	Join-accept	04 12 D6 BA	903.100	LoRa	125 kHz	SF 7	4/5	Ok	-103.00	-12.00
Rx	Dec/03/2022 18:19:21	3235313243001C00	Rejoin-request		903.700	LoRa	125 kHz	SF 7	-	Error	-103.00	-11.50
Rx	Dec/03/2022 18:18:11	3235313243001C00	Join-accept	F3 E3 5F 84	902.500	LoRa	125 kHz	SF 7	4/5	Error	-96.00	-12.00
Rx	Dec/03/2022 18:15:41	3235313243001C00	Join-accept	6F 9F FD FD	903.100	LoRa	125 kHz	SF 7	-	Error	-103.00	-11.75
Rx	Dec/03/2022 18:14:01	3235313243001C00	Confirmed Data Up	ED F1 72 84	902.900	LoRa	125 kHz	SF 7	4/8	Error	-102.00	-12.00
Rx	Dec/03/2022 18:10:21	3235313243001C00	Unconfirmed Data Down	C2 64 7C 20	902.900	LoRa	125 kHz	SF 7	-	Error	-101.00	-11.25
Rx	Dec/03/2022 18:07:41	3235313243001C00	Join-request		903.500	LoRa	125 kHz	SF 7	-	Error	-101.00	-11.25
Rx	Dec/03/2022 18:05:01	3235313243001C00	Confirmed Data Down	27 05 44 36	902.700	LoRa	125 kHz	SF 7	-	Error	-101.00	-11.00

Figura 21. Tráfico de mensajes del gateway

10. Para culminar con la integración de TTN se crea una cuenta en su página, para ello se debe conocer a que servidor se selecciona, existen tres opciones, Norte América, Europa y Australia, para Ecuador utilizamos la de Norte América (Figura 22).

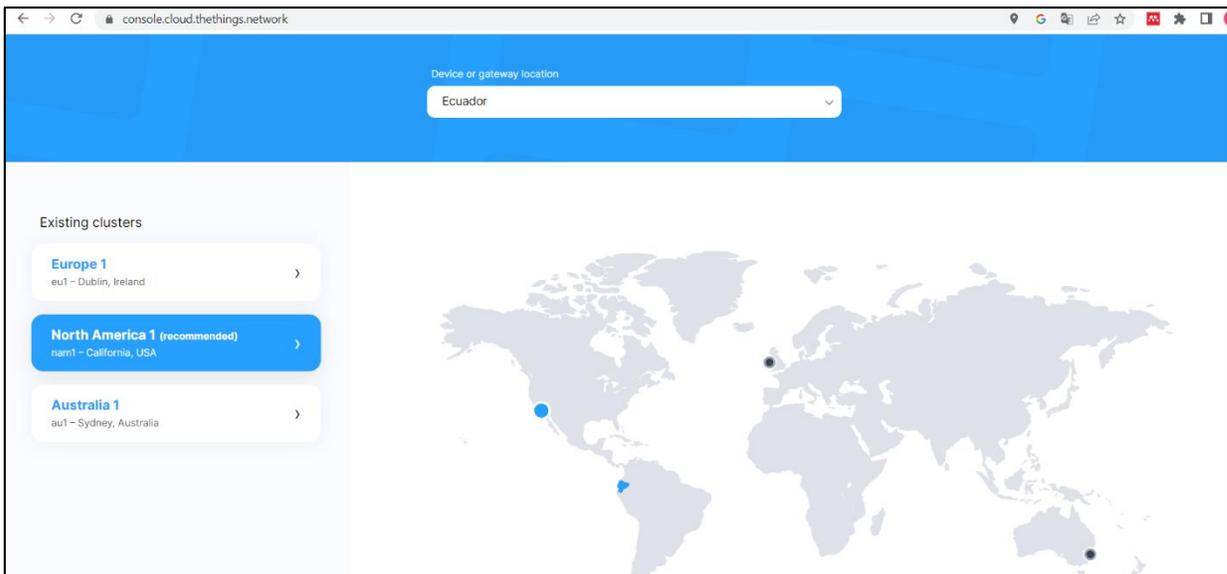


Figura 22. Selección de clúster para integración de gateway y nodos sensores a la TTN.

11. En la siguiente pestaña (Figura 23) seleccionamos el Gateway, dispositivos y aplicaciones.

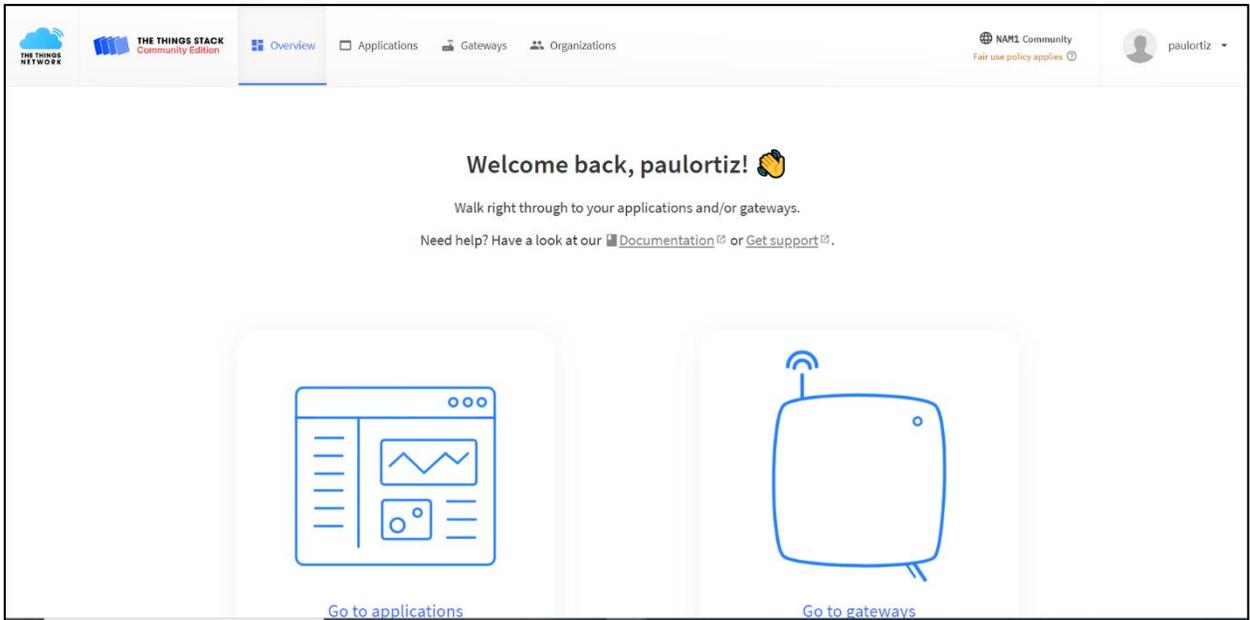


Figura 23. Pestaña de ingreso y selección de Dispositivos, nodos y gateway.

12. En la siguiente ventana (Figura 24) se registra los Nodos y el Gateway con la TTN.

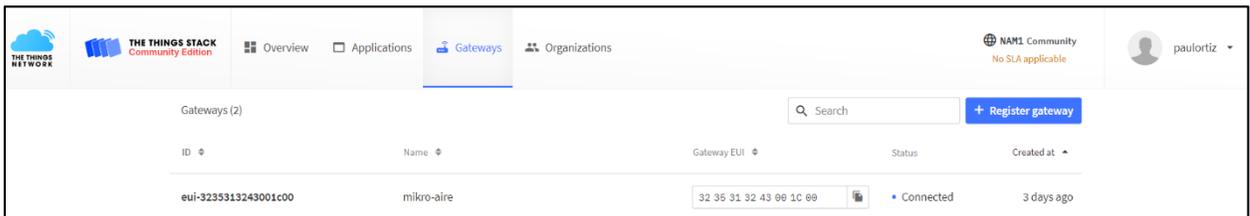


Figura 24. Ventana para agregar el Gateway. Fuente: TTN

13. Ingresar los datos para la identificación del Gateway como: la frecuencia, la región y como identificamos el Gateway (Figura 25).

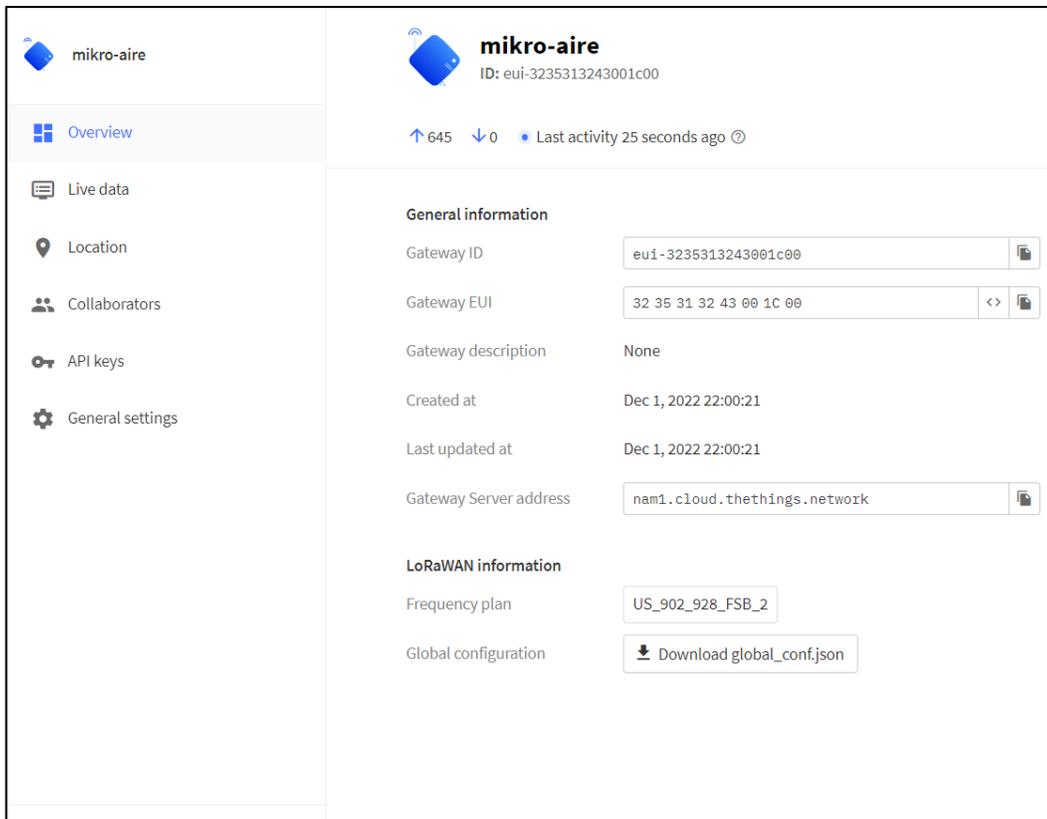


Figura 25. Configuración de parámetros del Gateway.

14. Administración del Gateway en la Consola de TTN (Figura 26).

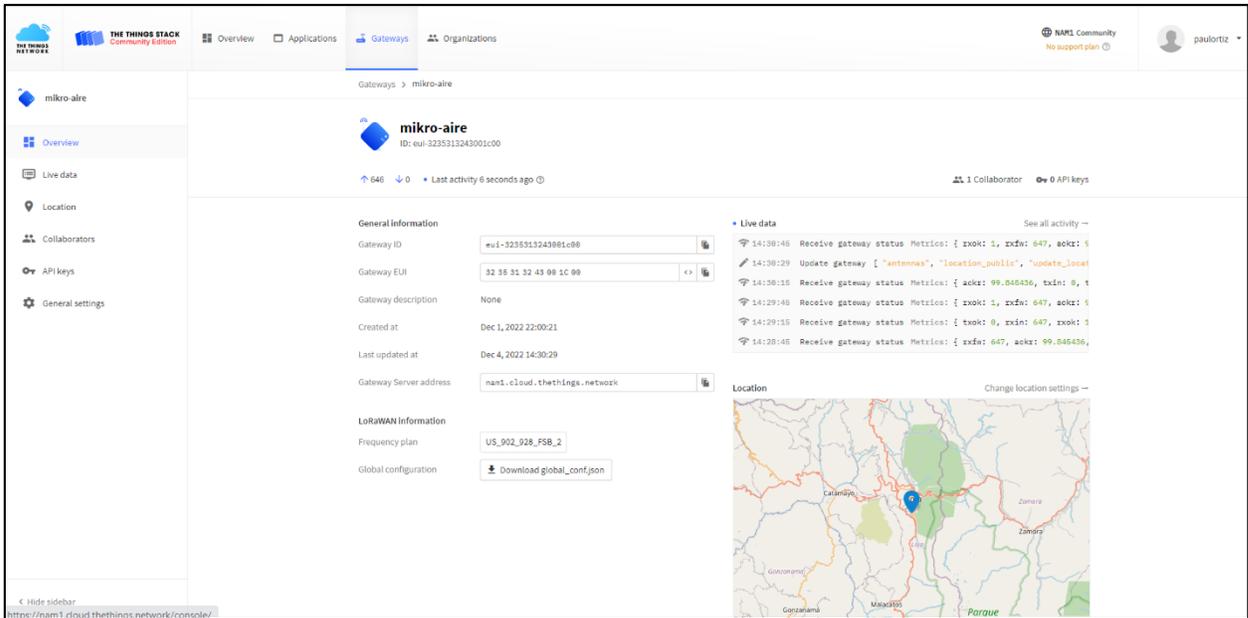


Figura 26. Configuración y registro del Gateway.

5.3.5. Nodos Sensores

Los nodos sensores lo conforman: los sensores, Arduino UNO y el módulo heltec cubecell; los datos son enviados desde los sensores al Arduino, mediante comunicación analógica y digital, dichos datos son almacenados y enviados al módulo cubecell utilizando la comunicación I2C; finalmente estos datos son transmitidos por el protocolo LoRaWAN al Gateway.

El módulo hace uso de un chip ASR6501 (Figura 27), que integra una MCU PSoC de la serie 4000 (ARM® Cortex® M0+ Core), y el chip LoRA SX1272. La principal ventaja es que son completamente compatibles con Arduino, tienen capacidad para ser alimentados directamente por batería o un pequeño panel solar (desde 5.5 a 7v), y un consumo realmente bajo: 10 mA en modo recepción LoRa, 70 mA emitiendo a 10 dB, y apenas 3.5uA en modo Deep Sleep, lo que los hacen muy adecuados para entornos de muy bajo consumo energético.

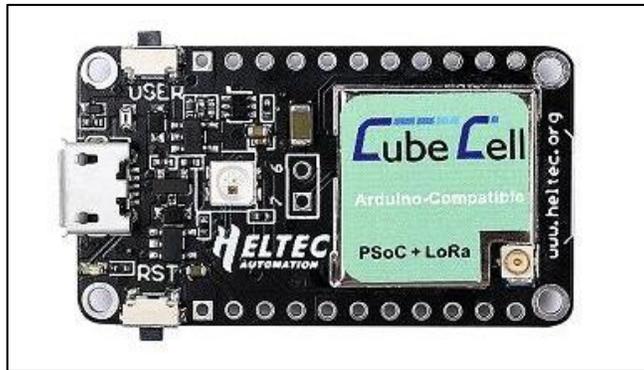


Figura 27. *Módulo Heltec Cube Cell*

Características:

- Chip ASR605X (ASR6501, ASR6502), estos chips integran el MCU PSoC 4000 series (ARM Cortex M0+ core) y el SX1262
- Soporta interfaz LoRaWAN 1.0.3
- Diseñado para bajo consumo de potencia
- Con interfaz para batería SH1.25-2, integra sistema de control para batería de litio
- Con interfaz micro USB
- Viene con chip CP2102 USB a puerto serial, útil para programación y debugging
- Con sistema de control para energía solar lo que la hace compatible con paneles solares de 5.5-7V
- Alimentación vía USB: 5V
- Alimentación por batería de Litio: 3.7V (4.2 máximo)
- Bandas de trabajo LoRa: EU_433, CN_470_510, EU_863_870, US_902_928
- Con interfaz UART, SPI, I2C
- Canal ADC 12-bit
- 6 pines GPIO
- Salida máxima LoRa 22dBm

- Con conector IPEX para antena LoRa
- Se puede utilizar para la conexión en red o la comunicación entre pares
- Dimensiones: 41.5 x 24.1 x 7.2mm

Los sensores que integran el nodo sensor son: MQ-7, MQ-131, MQ-136 y DSM501A. Destacan que son sensores de fiables y de precisión. En la tabla 7 se describen algunas características importantes:

Tabla 7. Características de los sensores del Nodo Sensor

MQ-7 (CO2)	MQ-131 (O3 y NO2)	MQ-136 (SO2)	DSM501A (PM10 y M2.5)
			
<ul style="list-style-type: none"> • Voltaje de Operación: 5V DC • Voltaje de Calentamiento: 5V (alto) y 1.4V (bajo) • Resistencia de carga: regulable • α: 21% • Rango de concentración: 20 a 2000 ppm 	<ul style="list-style-type: none"> • Voltaje de operación: 5V • Interface: salida analógica y digital TTL • Rango de detección de concentración: 2ppm • Dimensiones: 3.2cm x 2.2cm x 2.7cm 	<ul style="list-style-type: none"> • Voltaje de operación: 5V • Interface: salida analógica y digital TTL • Sensor MQ-136 • Rango de detección de concentración: 1 ~ 100ppm • Dimensiones: 3.2cm x 2.2cm x 3cm 	<ul style="list-style-type: none"> • Puede detectar 1 micrón o más de partículas pequeñas. • Circuito de entrada de 5V para procesamiento de señales. • El mantenimiento es simple, puede mantener las características del sensor por un largo tiempo. • Rango de tensión de funcionamiento DC 5V \pm 0.5V

Obtenido de: *Elaborado por el autor, 2023*

5.3.6. Diseño de la conexión de los Nodos Sensores

En el siguiente diseño muestra la conexión de los diferentes dispositivos para poder recoger y enviar los datos, primeramente, conectamos los sensores al Arduino UNO para poder recibir los datos de manera analógica, para poder transmitir estos valores al módulo Cubecell se

implementó la comunicación I2C de esta forma se envía desde el Arduino esclavo al Módulo maestro; el cual está conectado al Gateway mediante el protocolo Lorawan para la visualización de los datos (Figura 28).

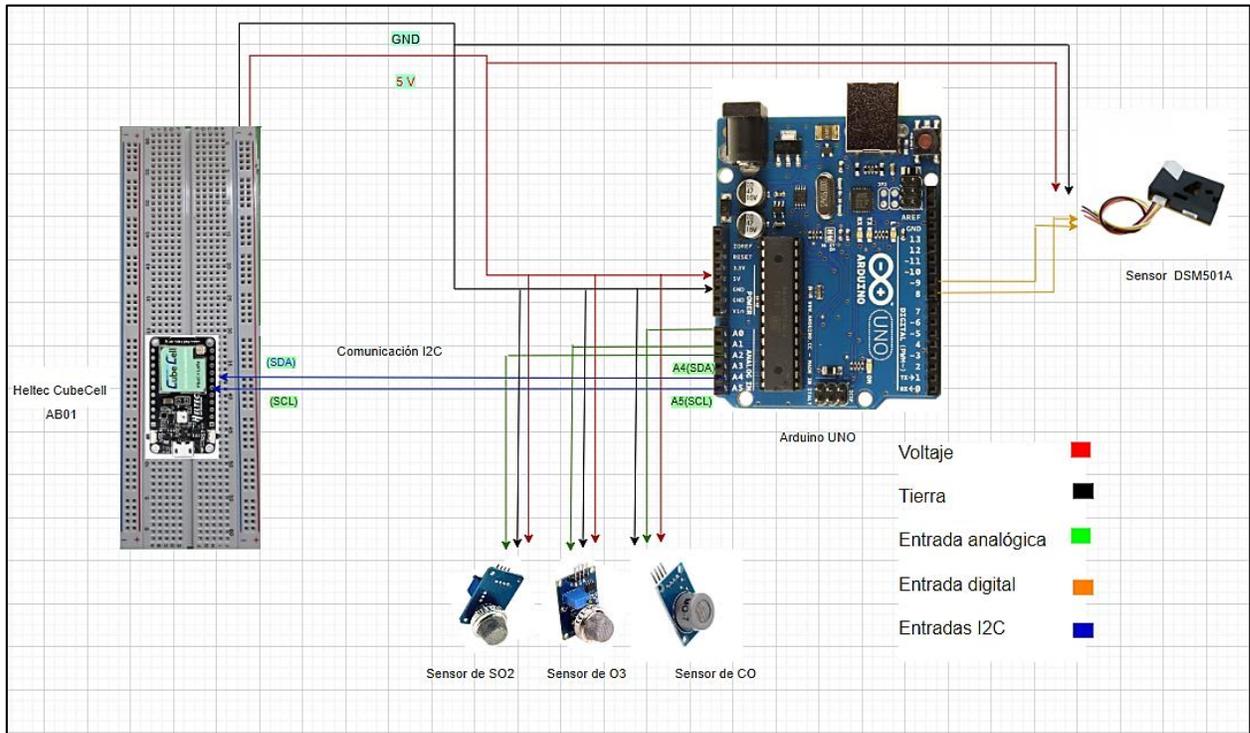


Figura 28. Conexión de los nodos sensores.

Fuente: Elaborado por el autor, 2023

5.3.7. Configuración de los Módulos Heltec Cubecell

1. En el IDE de arduino nos dirigimos en archivos después a preferencias (figura 29), y copiamos el link. json: [https://github.com/HelTecAutomation/CubeCell-Arduino/releases/download /V1.4.0/package_CubeCell_index.json](https://github.com/HelTecAutomation/CubeCell-Arduino/releases/download/V1.4.0/package_CubeCell_index.json) para descargar la librería (Figura 30).

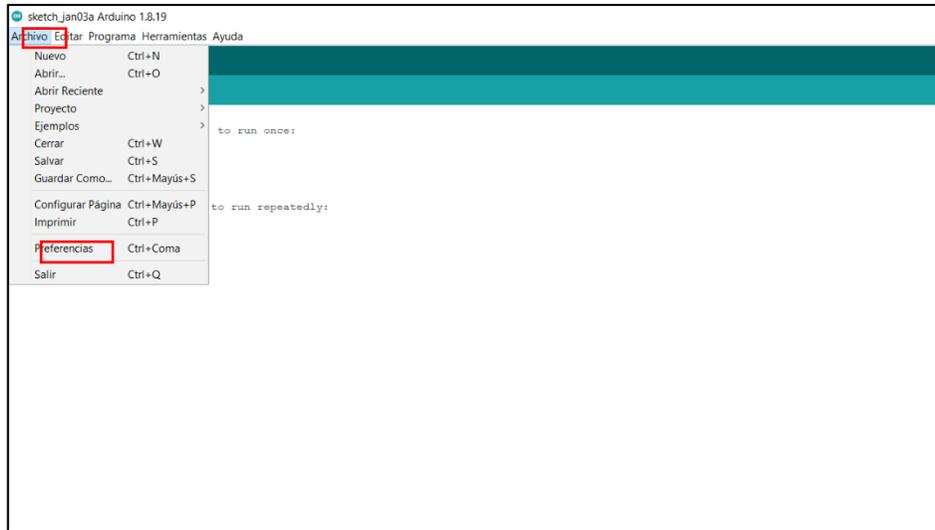


Figura 29. IDE de Arduino

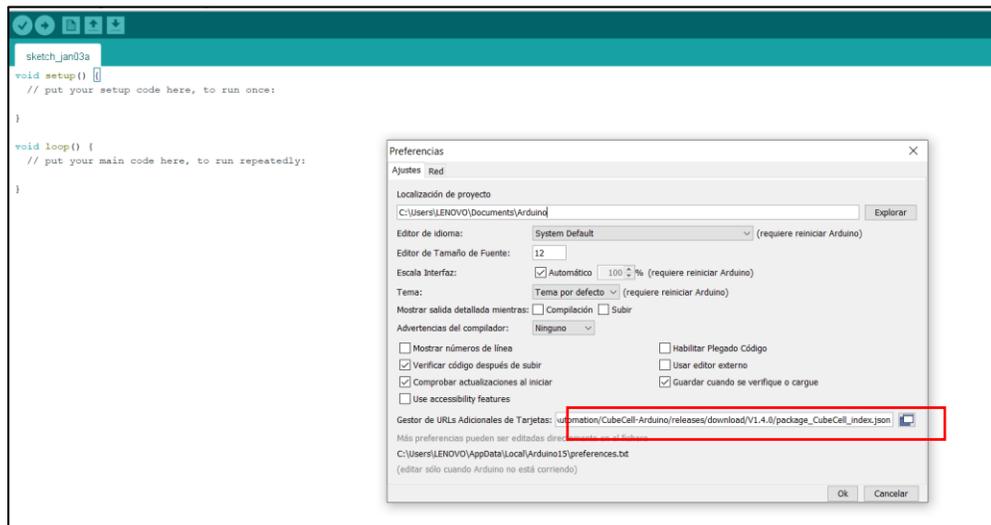


Figura 30. Link de Descarga. json.

2. Posteriormente damos clic en herramientas y vamos a gestor de placas e instalamos cubecell, (figura, 31 y 32).

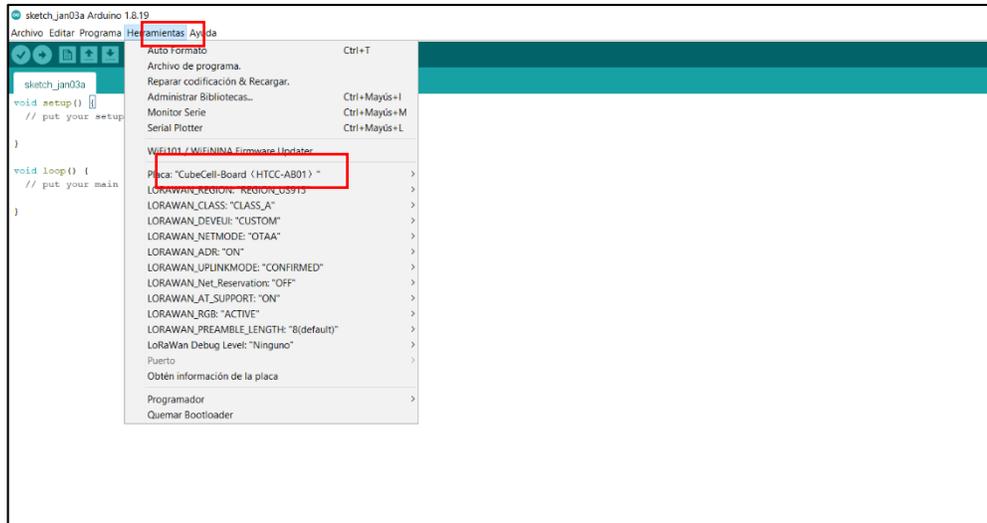


Figura 31. *Gestor de placas*

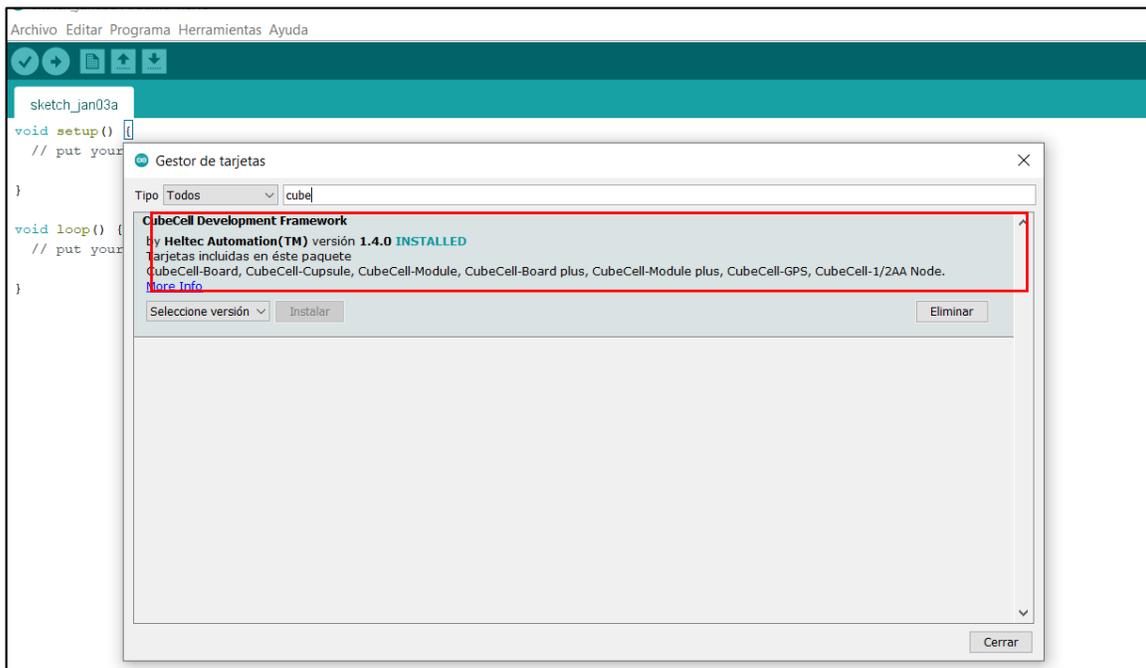


Figura 32. *Instalación de cubecell en el IDE de arduino*

3. Instalado Cubecell se procede a elegir el módulo y el modelo a utilizar, como se muestra en la figura 33.

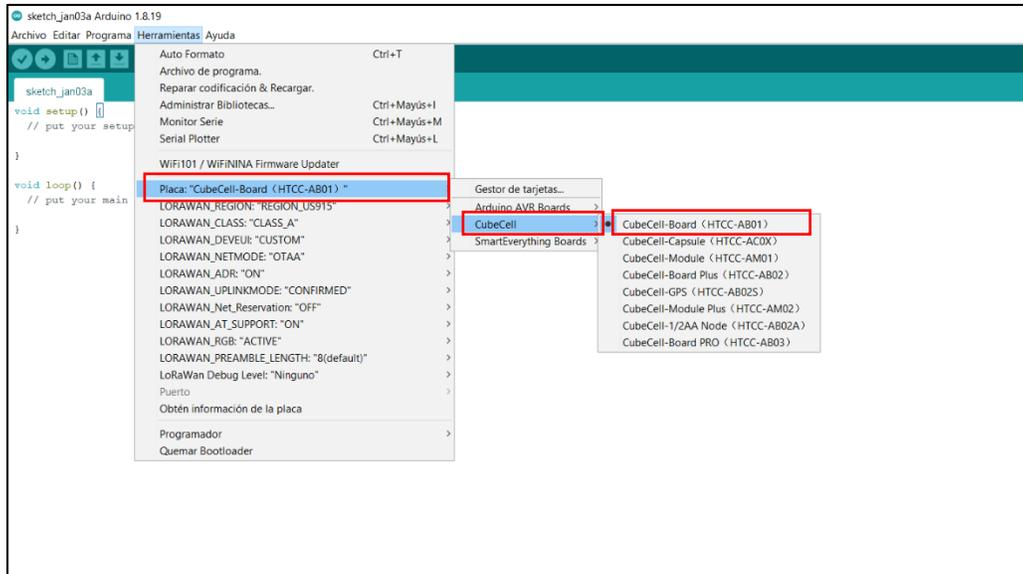


Figura 33. Elección del módulo y modelo a utilizar

4. Se configura los parámetros como la frecuencia, clase, modo de operación, Deveui entre otras.

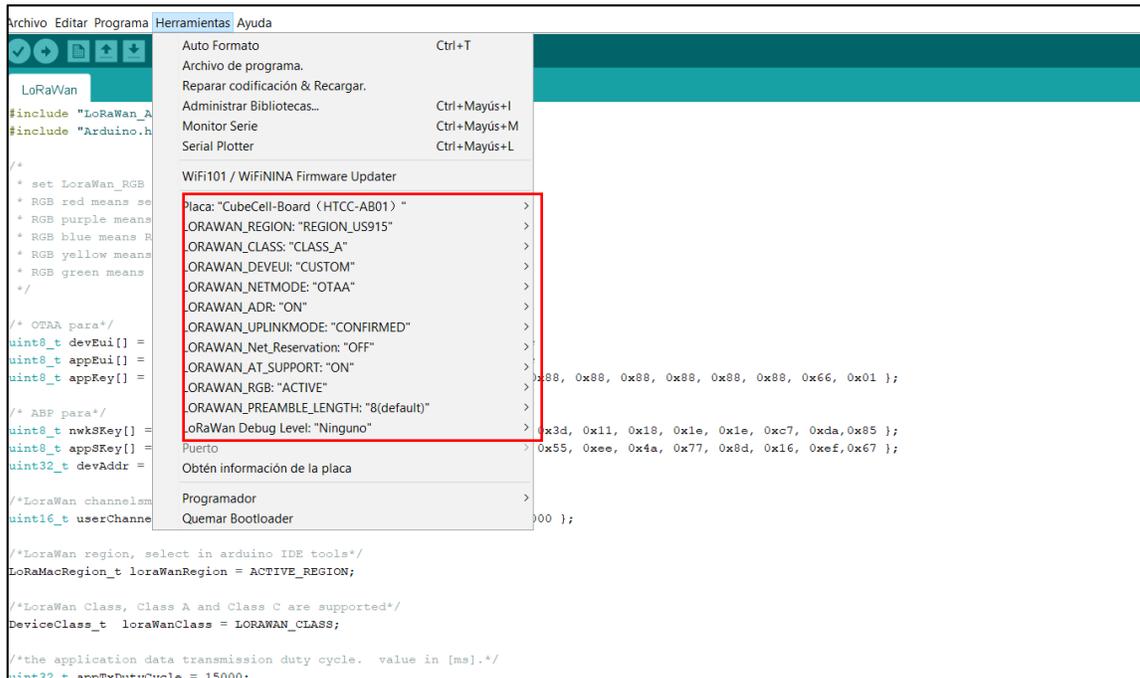


Figura 34. Configuración de los parámetros de los módulos

5. Para poder vincular los módulos a la TTN se debe modificar el código que encontramos en el IDE de Arduino.

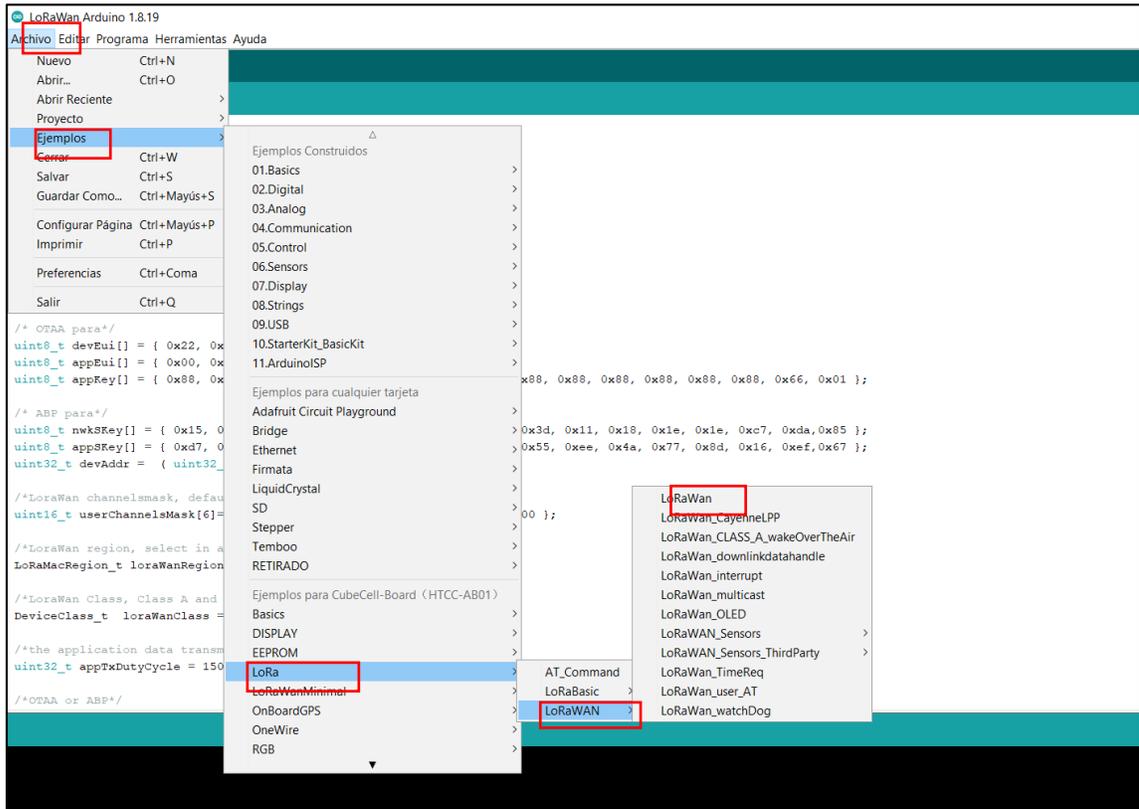


Figura 35. Proceso de selección del código

```
Archivo Editar Programa Herramientas Ayuda
Subir
LoRaWan
#include "LoRaWan_APP.h"
#include "Arduino.h"

/*
 * set LoRaWan_RGB to Active, the RGB active in loraWan
 * RGB red means sending;
 * RGB purple means joined done;
 * RGB blue means RxWindow1;
 * RGB yellow means RxWindow2;
 * RGB green means received done;
 */

/* OTAA para*/
uint8_t devEui[] = { 0x22, 0x32, 0x33, 0x00, 0x00, 0x88, 0x88, 0x02 };
uint8_t appEui[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
uint8_t appKey[] = { 0x88, 0x66, 0x01 };

/* ABP para*/
uint8_t nwkSKey[] = { 0x15, 0xb1, 0xd0, 0xef, 0xa4, 0x63, 0xdf, 0xbe, 0x3d, 0x11, 0x18, 0x1e, 0x1e, 0xc7, 0xda, 0x85 };
uint8_t appSKey[] = { 0xd7, 0x2c, 0x78, 0x75, 0x8c, 0xdc, 0xca, 0xbf, 0x55, 0xee, 0x4a, 0x77, 0x8d, 0x16, 0xef, 0x67 };
uint32_t devAddr = ( uint32_t )0x007e6ae1;

/*LoraWan channelsmask, default channels 0-7*/
uint16_t userChannelsMask[6]={ 0x00FF,0x0000,0x0000,0x0000,0x0000,0x0000 };

/*LoraWan region, select in arduino IDE tools*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;

/*LoraWan Class, Class A and Class C are supported*/
DeviceClass_t loraWanClass = LORAWAN_CLASS;

/*the application data transmission duty cycle. value in [ms].*/
uint32_t appTxDutyCycle = 15000;

/*OTAA or ABP*/
```

Figura 36. Código para trabajar con los módulos Cubecell mediante OTAA en TTN.

6. Creamos una aplicación para conectar los módulos a la TTN.

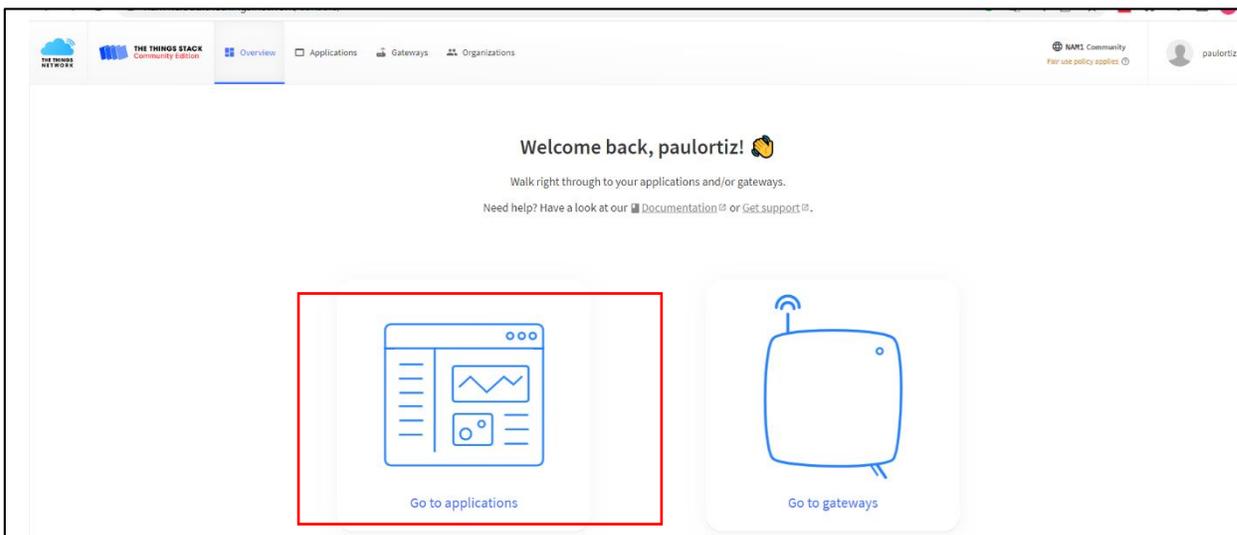


Figura 37. Página TTN.

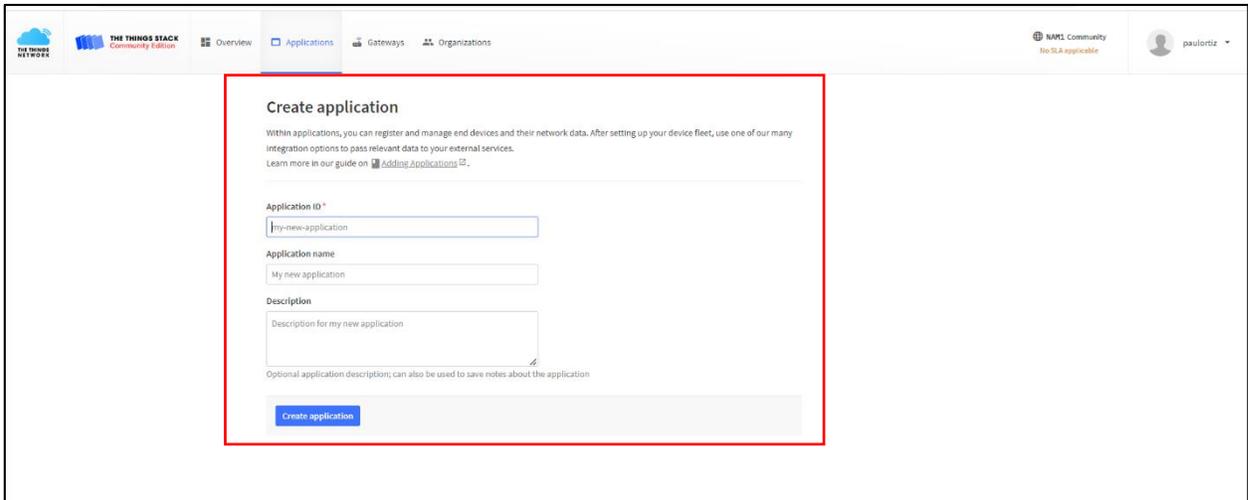


Figura 38. *Completamos los parámetros y creamos la aplicación*

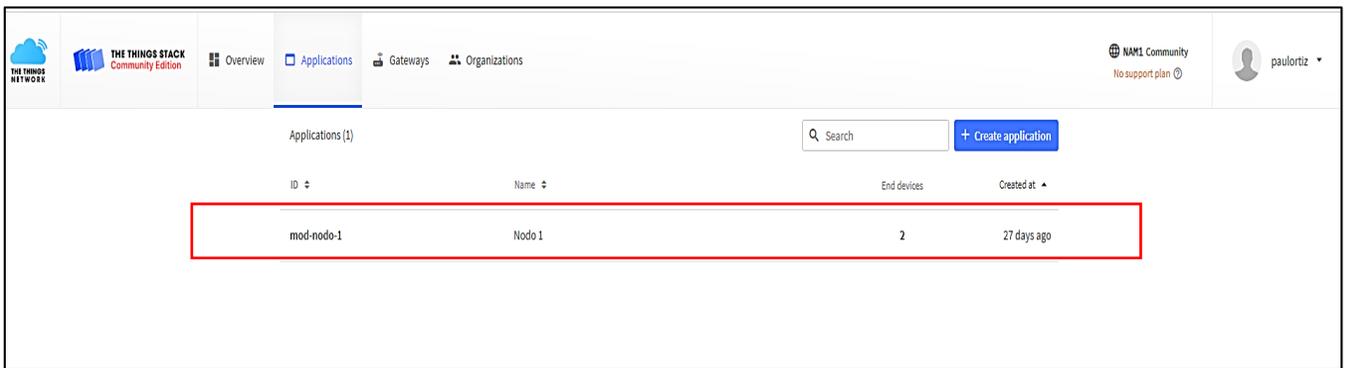


Figura 39. *Aplicación creada para operar con los módulos.*

7. Registramos los módulos en la aplicación

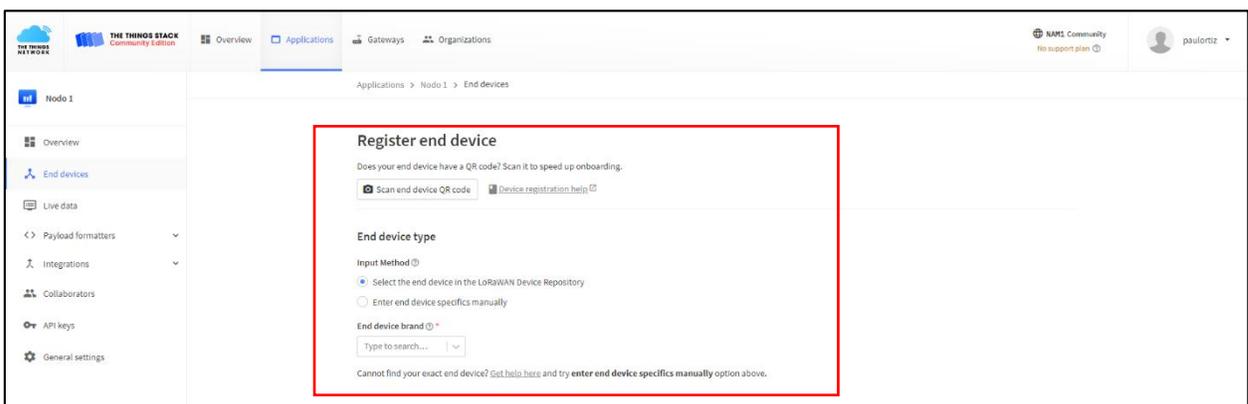


Figura 40. *Registro de los módulos*

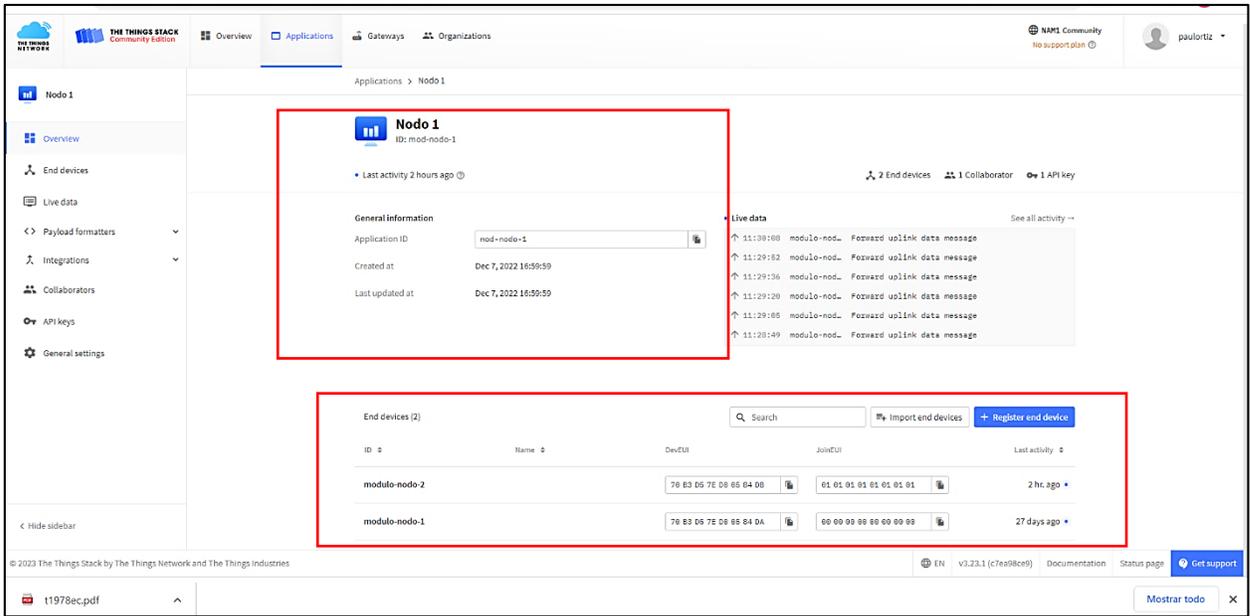


Figura 41. Aplicación creada con los módulos registrados.

8. Se realiza la configuración de los módulos, agregando los parámetros de la TTN como son: el *devEui*, *appEui* y *appKey* en el código de Arduino para la compilación.

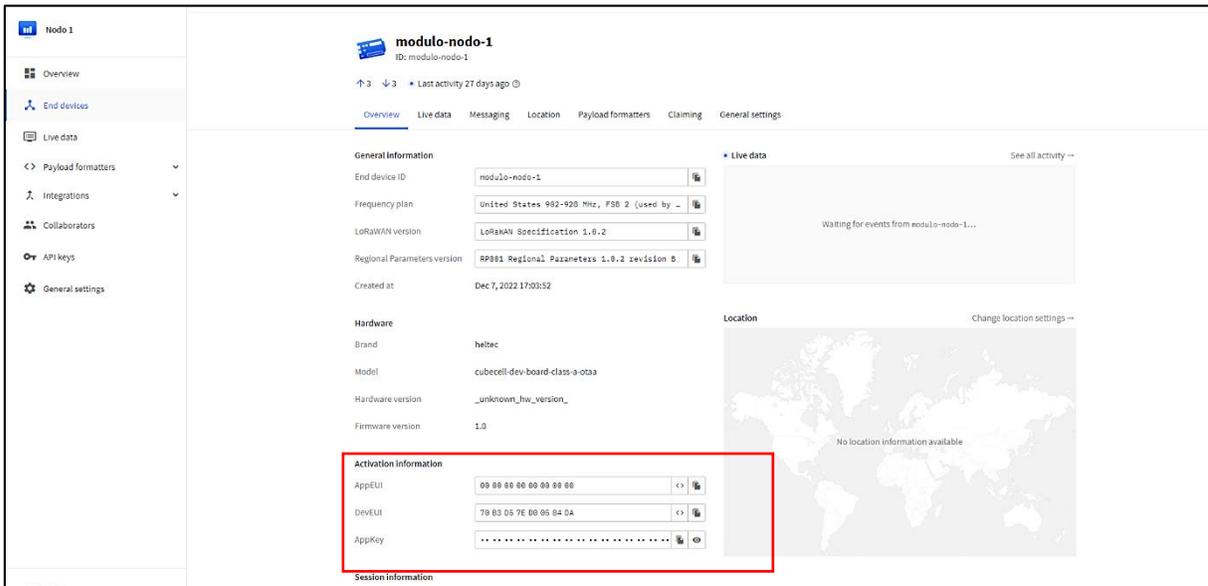


Figura 42. Parámetros del módulo 1 en la TTN.

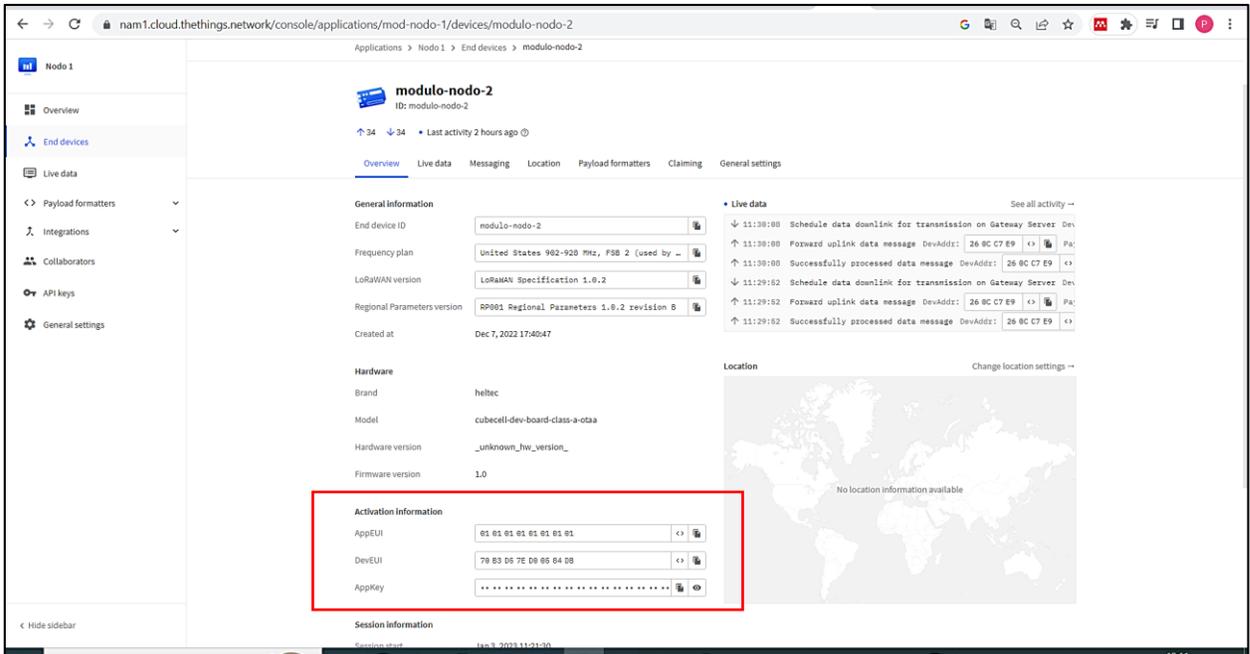


Figura 43. Parámetros del módulo 2 en la TTN.

9. Copiamos el *devEui*, *appEui* y *appKey* en el código de arduino para el primer módulo.

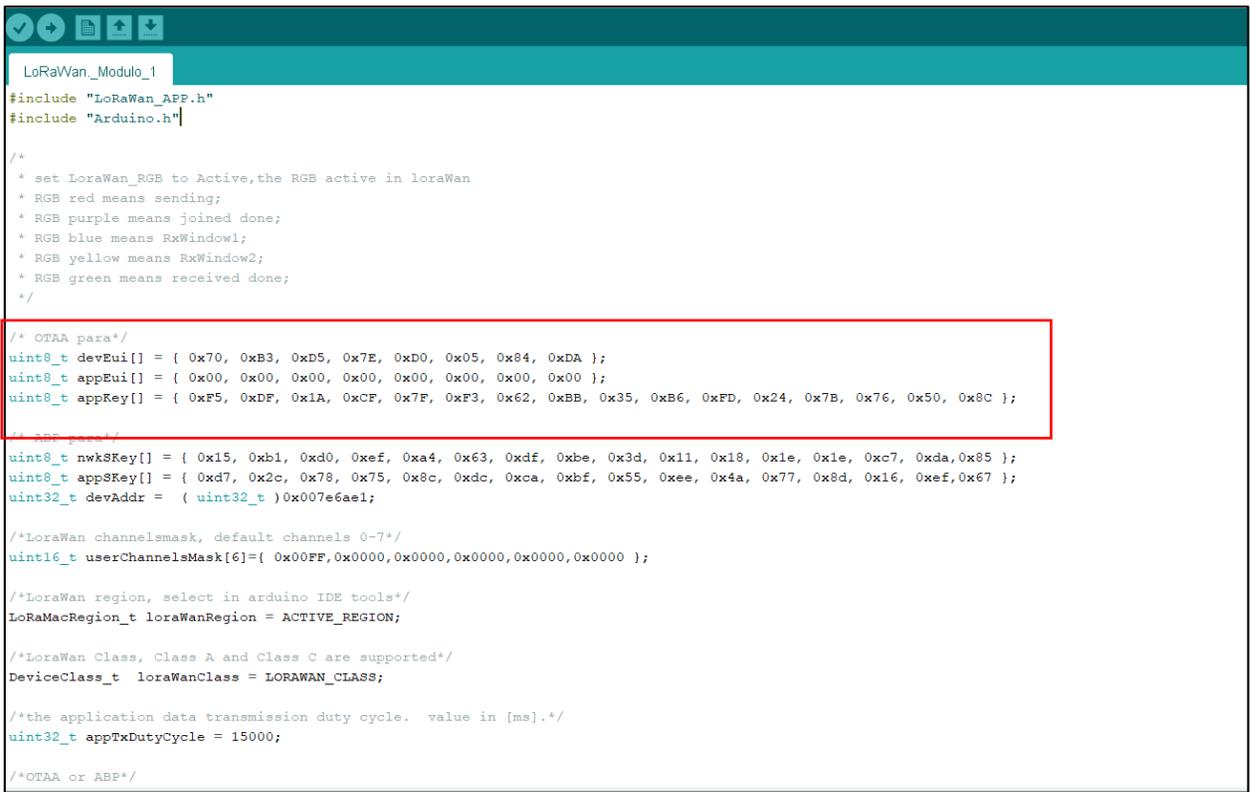
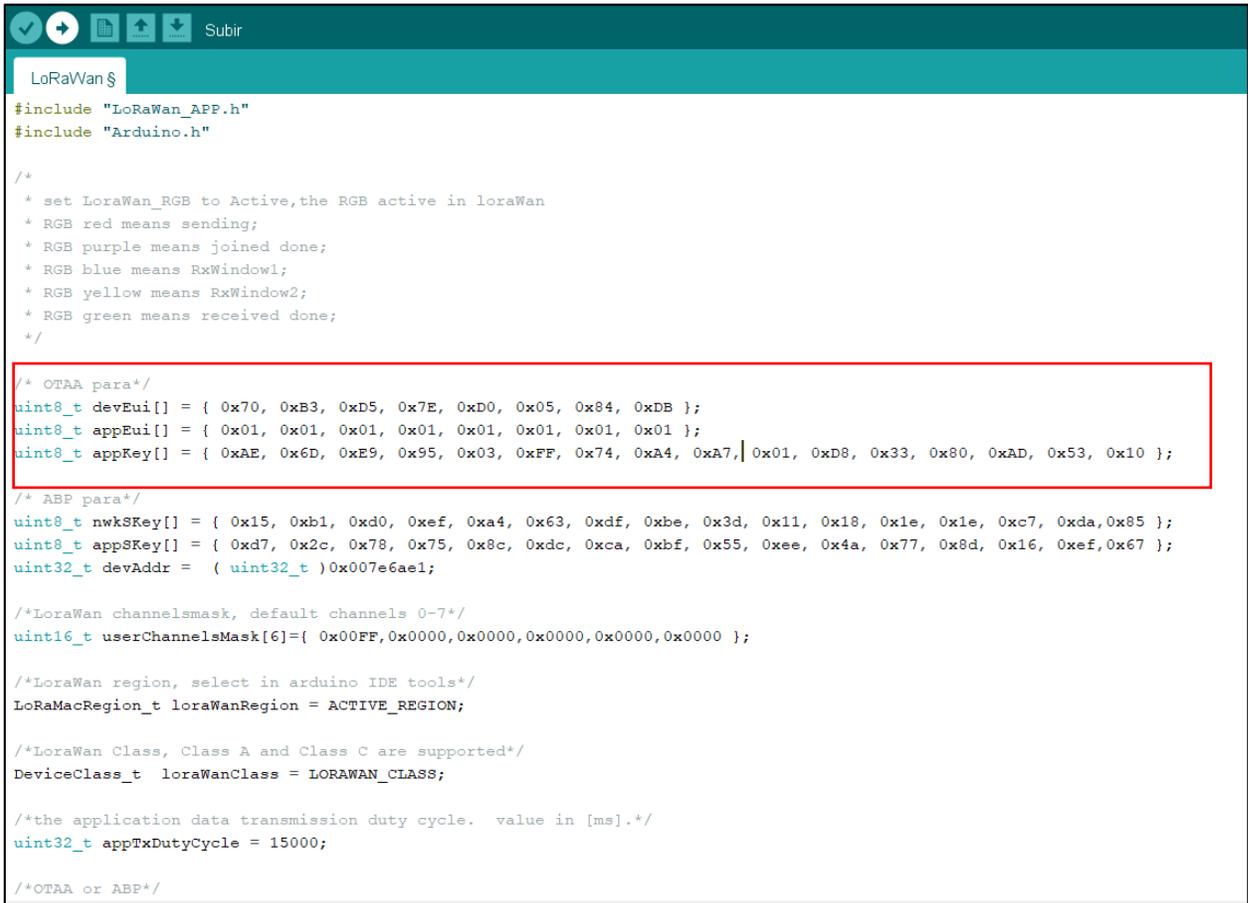


Figura 44. Código Configurado para el módulo 1 en la TTN.

10. Copiamos el *devEui*, *appEui* y *appKey* en el código de arduino para el segundo módulo.



```
LoRaWAN §
#include "LoRaWAN_APP.h"
#include "Arduino.h"

/*
 * set LoraWan_RGB to Active,the RGB active in loraWan
 * RGB red means sending;
 * RGB purple means joined done;
 * RGB blue means RxWindow1;
 * RGB yellow means RxWindow2;
 * RGB green means received done;
 */

/* OTAA para*/
uint8_t devEui[] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0x84, 0xDB };
uint8_t appEui[] = { 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01 };
uint8_t appKey[] = { 0xAE, 0x6D, 0xE9, 0x95, 0x03, 0xFF, 0x74, 0xA4, 0xA7, 0x01, 0xD8, 0x33, 0x80, 0xAD, 0x53, 0x10 };

/* ABP para*/
uint8_t nwkSKey[] = { 0x15, 0xb1, 0xd0, 0xef, 0xa4, 0x63, 0xdf, 0xbe, 0x3d, 0x11, 0x18, 0x1e, 0x1e, 0xc7, 0xda,0x85 };
uint8_t appSKey[] = { 0xd7, 0x2c, 0x78, 0x75, 0x8c, 0xdc, 0xca, 0xbf, 0x55, 0xee, 0x4a, 0x77, 0x8d, 0x16, 0xef,0x67 };
uint32_t devAddr = ( uint32_t )0x007e6ae1;

/*LoraWan channelsmask, default channels 0-7*/
uint16_t userChannelsMask[6]={ 0x00FF,0x0000,0x0000,0x0000,0x0000,0x0000 };

/*LoraWan region, select in arduino IDE tools*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;

/*LoraWan Class, Class A and Class C are supported*/
DeviceClass_t loraWanClass = LORAWAN_CLASS;

/*the application data transmission duty cycle. value in [ms].*/
uint32_t appTxDutyCycle = 15000;

/*OTAA or ABP*/
```

Figura 45. Código Configurado para el módulo 2 en la TTN

11. Como observamos el *live data* envió y recepción de los datos del primer módulo.

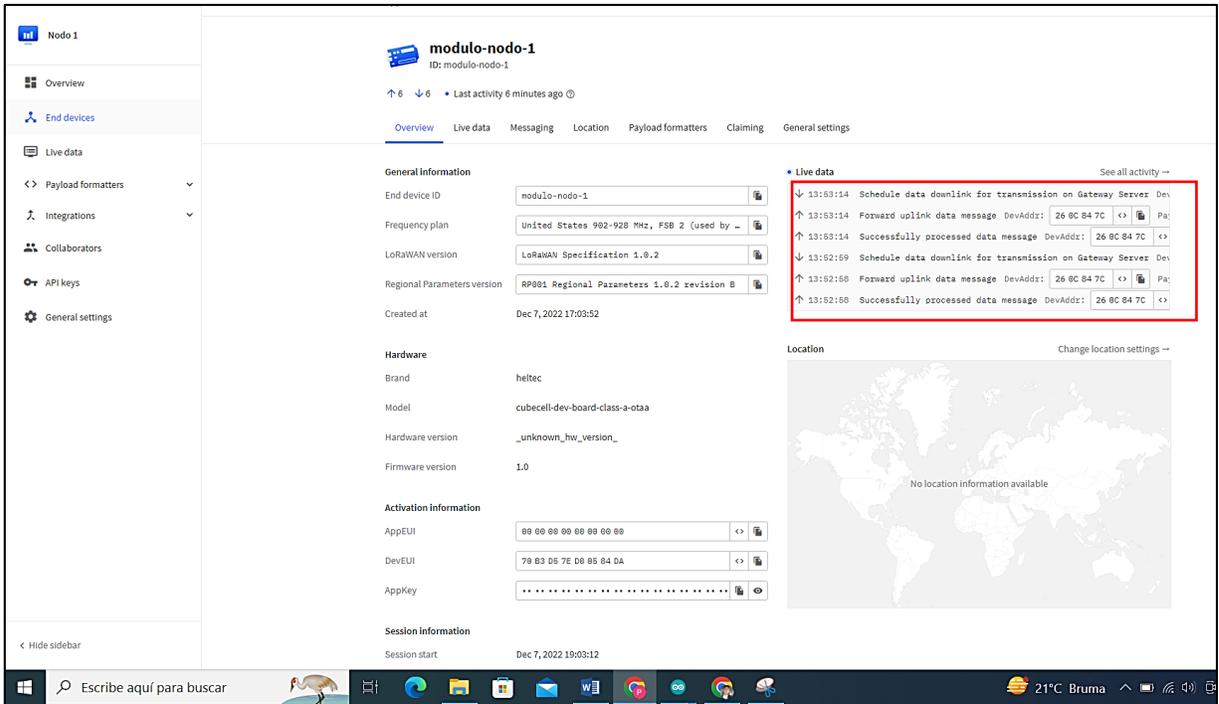


Figura 46. Recepción y envío de datos del módulo 1.

10. Envío y recepción de los datos del segundo módulo.

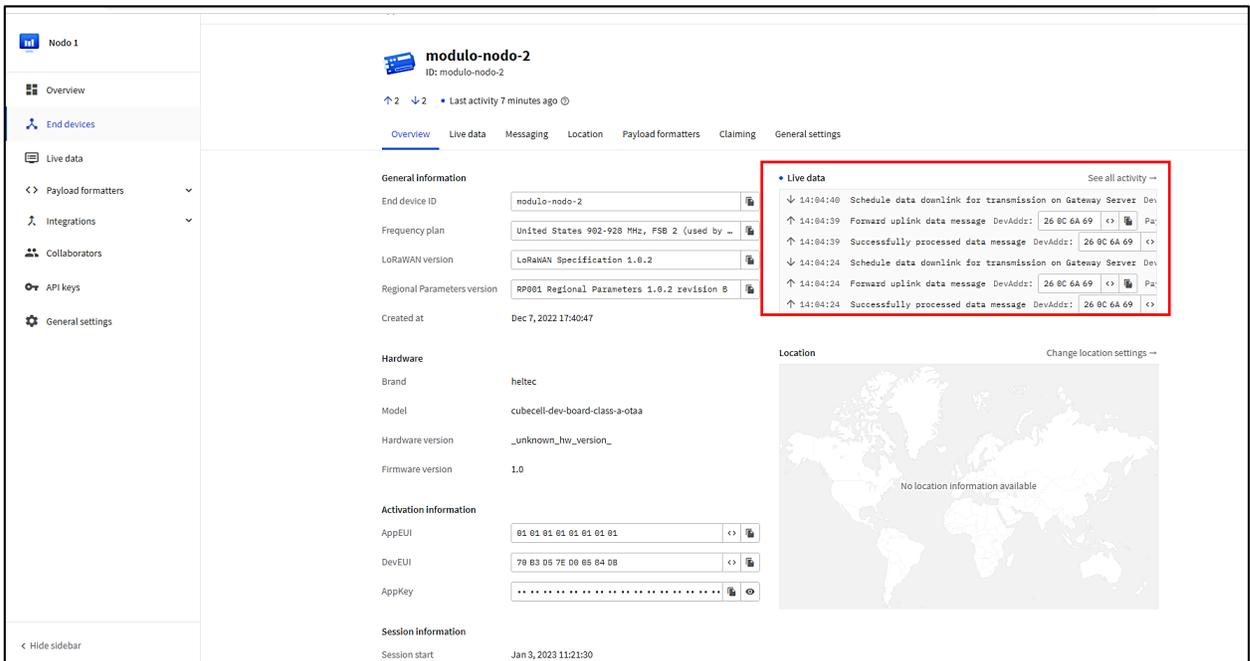


Figura 47. Envío y recepción de datos del módulo 2.

Completado el registro del Gateway y los nodos, se pasa a la fase de monitoreo de la calidad del aire, los datos son encriptados durante su transmisión y recepción desde TTN debido

a que los datos son enviados en forma de tramas, por tal motivo se debe interpretarlos para poder apreciarlos, esto se puede hacer por la opción *Payload formatters*, en la cual se debe escribir el siguiente código dependiendo de las variables que se vayan a utilizar como se muestra en la figura 48.

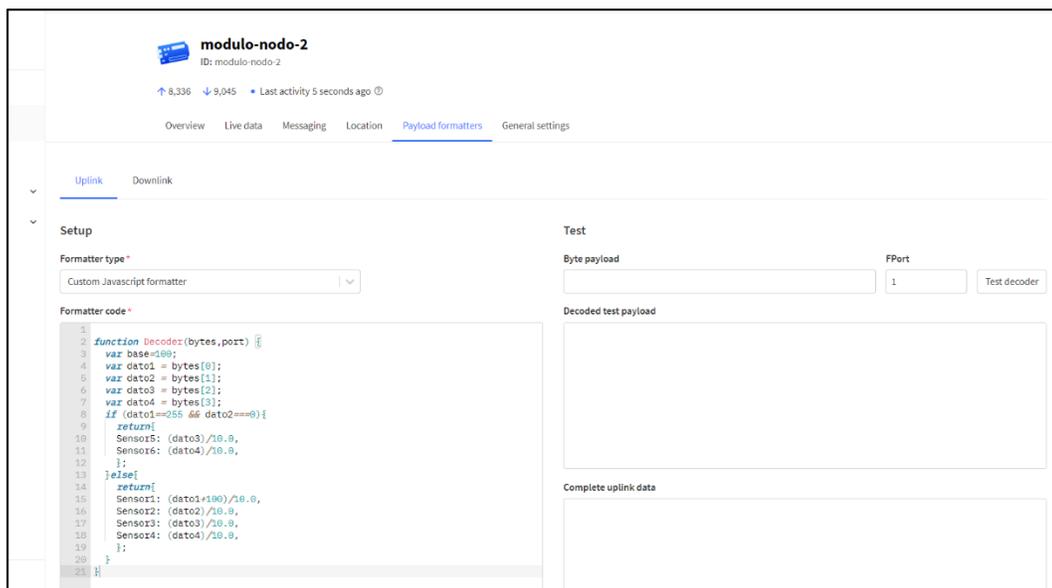


Figura 48. Configuración del *payload formatters*. Fuente: TTN

Las variables son las que se observan en *live data* al momento que se reciban datos del medidor de agua inteligente como se muestra en la figura 49.

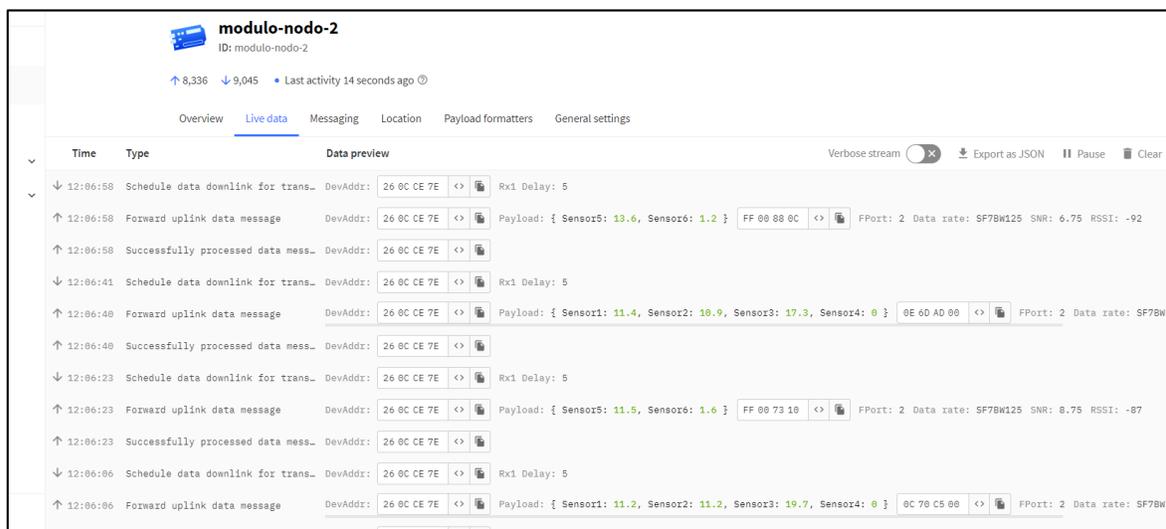


Figura 49. Datos mostrados en *Live Data*. Fuente TTN

Una vez definidas las variables se procedió a la integración entre TTN y Ubidots, en donde las variables creadas en TTN serán clonadas en Ubidots junto a sus valores, para poder mostrarlos gráficamente.

Para poder ver gráficamente las variables en Ubidots primeramente se creó una cuenta, y una vez creada la cuenta se agregó un nuevo plugin de datos seleccionando el complemento de The Thinkgs Stack como se muestra en la figura 50.

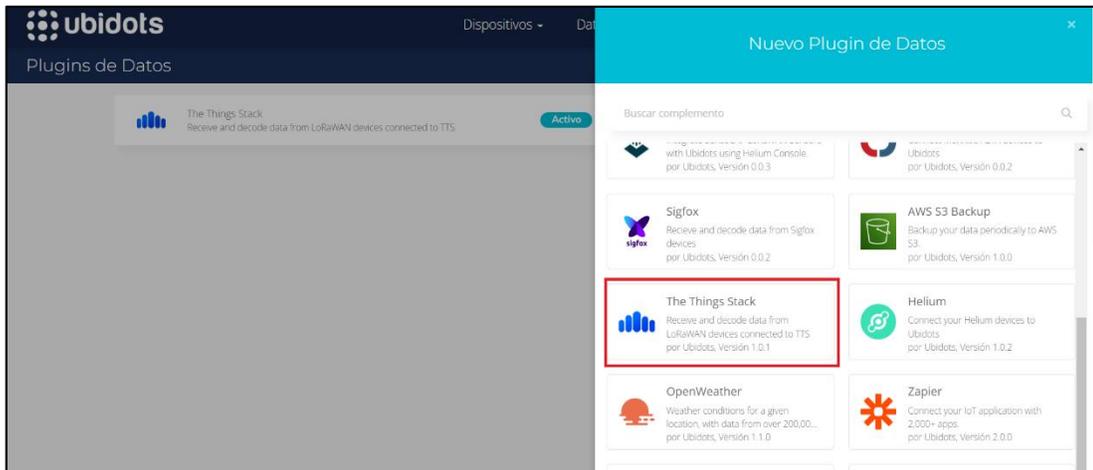


Figura 50. Creación del Plugin de datos.

Al seleccionar el complemento se despliegan los pasos a seguir para la correcta integración, seguidos estos pasos el plugin de datos se activa correctamente como se observa en la figura 51.

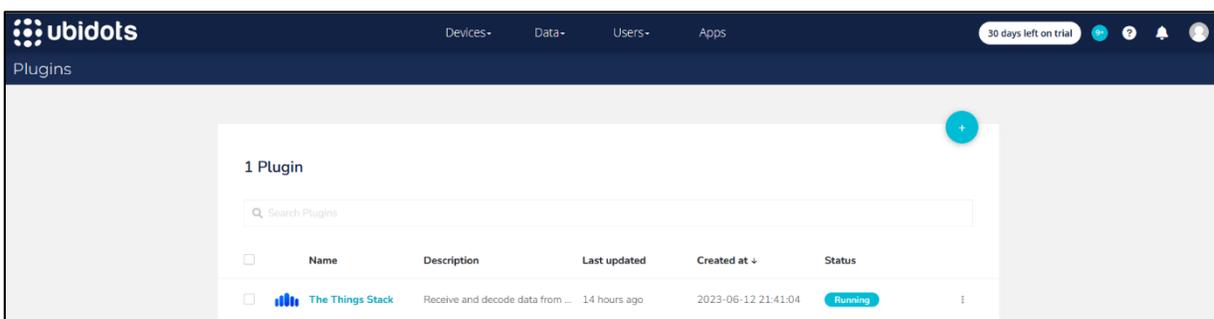


Figura 51. Plugin de datos activo.

Después de haber creado el plugin de datos se procedió a realizar la adaptación del código de ejemplo para una correcta vinculación de TTN con Ubidots, todo esto entrando al plugin creado, dentro del cual se debe ir a *Decoder* y deslizar hasta observar el código de

ejemplo en donde es importante leer los comentarios para poder saber que líneas se puede eliminar y comentar. El código una vez modificado queda de la siguiente manera:

```
function format_payload(args){
  var Ubidots_payload = {};
  // Log received data for debugging purposes:
  // console.log(JSON.stringify(args));
  // Get RSSI and SNR variables using gateways data:
  /*var gateways = args['uplink_message']['rx_metadata'];
  for (const i in gateways) {
    // Get gateway EUI and name
    var gw = gateways[i];
    var gw_eui = gw['gateway_ids']['eui'];
    var gw_id = gw['gateway_ids']['gateway_id'];
    // Build RSSI and SNR variables
    Ubidots_payload['rssi-' + gw_id] = {
      "value": gw['rssi'],
      "context": {
        "channel_index": gw['channel_index'],
        "channel_rssi": gw['channel_rssi'],
        "gw_eui": gw_eui,
        "gw_id": gw_id,
        "uplink_token": gw['uplink_token']
      }
    }
  }
  Ubidots_payload['snr-' + gw_id] = gw['snr'];

  */

  // Get Fcnt and Port variables:
  // Ubidots_payload['f_cnt'] = args['uplink_message']['f_cnt'];
  // Ubidots_payload['f_port'] = args['uplink_message']['f_port'];
```

```

    // Get uplink's timestamp
    // Ubidots_payload['timestamp'] = new Date(args['uplink_message']['received_a
t']).getTime());

    // If you're already decoding in TTS using payload formatters,
    // then uncomment the following line to use "uplink_message.decoded_payload"
    .
    // PROTIP: Make sure the incoming decoded payload is an Ubidots-
compatible JSON (See https://Ubidots.com/docs/hw/#sending-data)
    var decoded_payload = args['uplink_message']['decoded_payload'];

    // By default, this plugin uses "uplink_message.frm_payload" and sends it to
the decoding function "decodeUplink".
    // For more vendor-
specific decoders, check out https://github.com/TheThingsNetwork/lorawan-
devices/tree/master/vendor

    // Merge decoded payload into Ubidots payload
    Object.assign (Ubidots_payload, decoded_payload);
    return Ubidots_payload
}
module.exports = { format_payload };

```

Creado el plugin de datos y modificado el código de ejemplo el paso a seguir fue crear un Webhooks de Ubidots en TTN. Para poder realizarlo se ingresó a TTN en la parte izquierda en la opción denominada *integrations* que se muestra en la figura 52, en donde se siguió los pasos desplegados en la guía que se mostró en la creación del plugin de datos de Ubidots. Al momento de realizar este paso se debe tomar en cuenta el Token requerido para hacer la integración, que se lo puede encontrar en Ubidots en la parte superior derecha entrando al perfil de usuario en la parte de credenciales API (ver figura 53).



Figura 52. Integración de TTN.

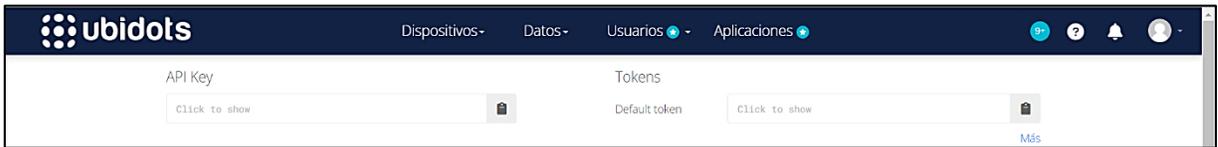


Figura 53. Token Ubidots.

Fuente: Elaborado por el autor

Finalmente, realizado todos los pasos se pueden observar las variables de TTN en Ubidots en la parte de dispositivos como se muestra en la figura 54.

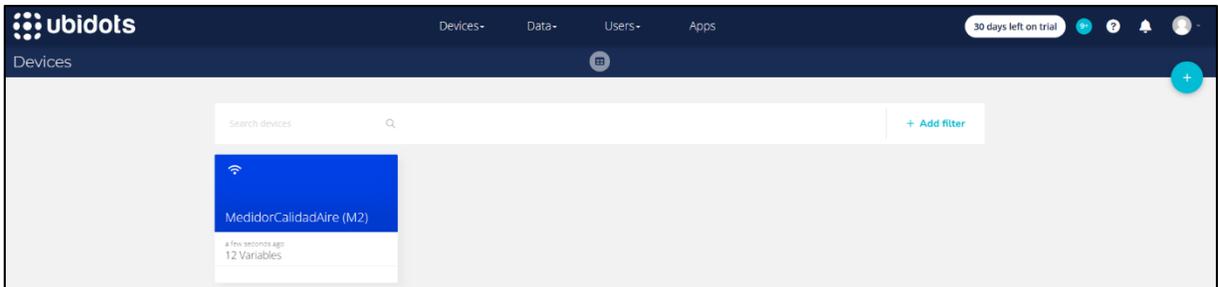


Figura 54. Variables enviadas de TTN a Ubidots.

Fuente: Elaborado por el autor

Para la parte de control de los valores máximos de los gases, se debe crear una variable en Ubidots y trabajar con su modo de eventos, que se encuentra en el apartado de datos como se observa en la figura 55.

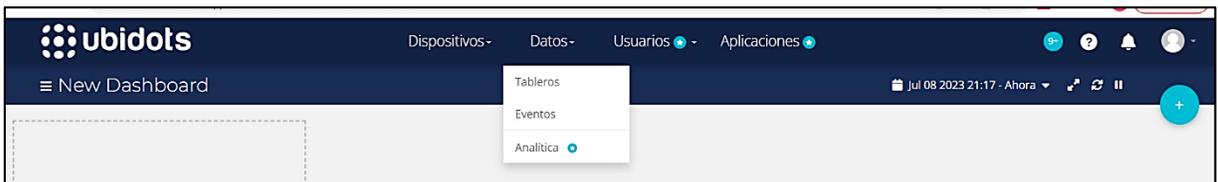
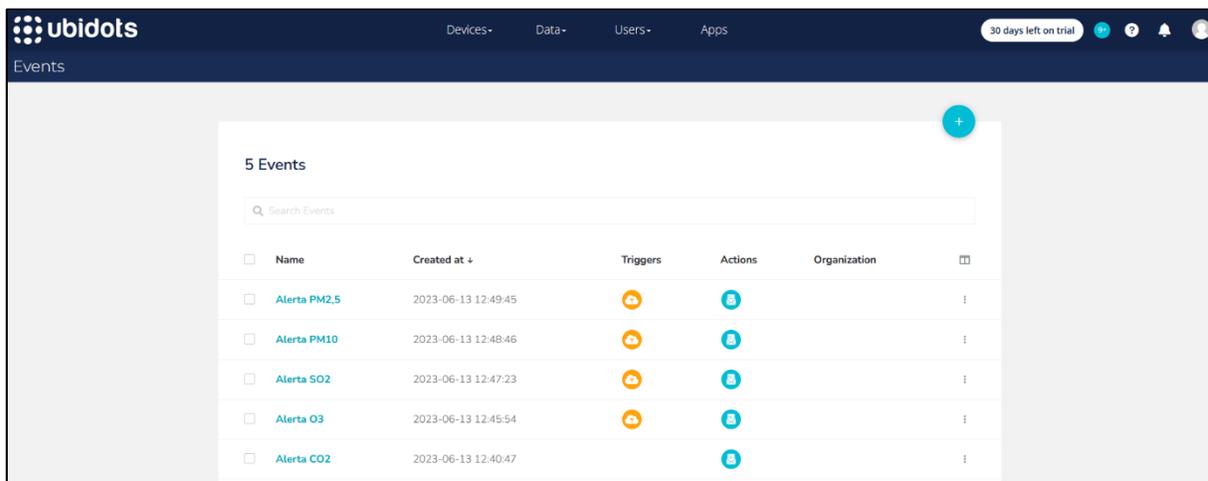


Figura 55. Ventana para la creación de eventos

Fuente: Elaborado por el autor.

Una vez ingresado al apartado de eventos se procedió a crear un evento para cada uno de los gases, con los niveles máximos de contaminación permitidos, la plataforma Ubidots notificará mediante email, SMS y Telegram los valores que excedan su límite, de manera que notifica el valor del gas, la hora y la fecha que ha ocurrido el evento, se muestra en la figura 56.



<input type="checkbox"/>	Name	Created at	Triggers	Actions	Organization	
<input type="checkbox"/>	Alerta PM2.5	2023-06-13 12:49:45				⋮
<input type="checkbox"/>	Alerta PM10	2023-06-13 12:48:46				⋮
<input type="checkbox"/>	Alerta SO2	2023-06-13 12:47:23				⋮
<input type="checkbox"/>	Alerta O3	2023-06-13 12:45:54				⋮
<input type="checkbox"/>	Alerta CO2	2023-06-13 12:40:47				⋮

Figura 56. *Eventos creados para cada gas contaminante.*

Para realizar la configuración del envío de alertas por Telegram, que a diferencia de SMS y correo electrónico solo se agrega la dirección de correo y el número telefónico, en Telegram se deben seguir los siguientes pasos:

Primero debemos crear un bot en Telegram, donde debe abrir la aplicación de Telegram y en el icono de búsqueda escribir “BotFather” para iniciar un chat, figura 57.



Figura 57. Busque en Telegram de BotFather.

Fuente: Elaborado por el autor

Como se observa en la figura 58 en el chat con BotFather, escribe "/newbot" y envía el mensaje. BotFather le pedirá que elija un nombre para su bot, elija uno y envíelo como un mensaje.

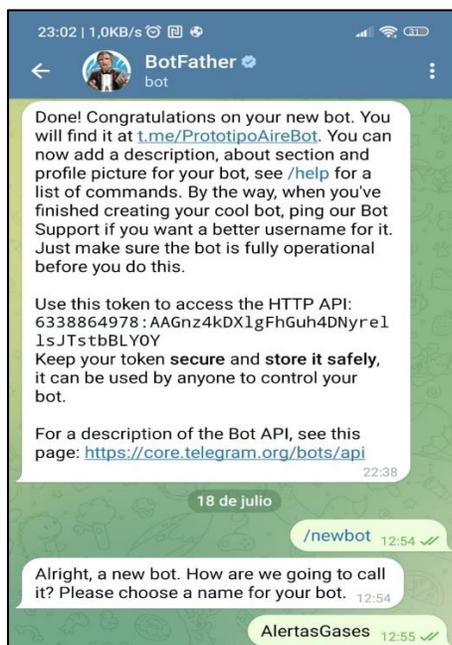


Figura 58. Chat en de Telegram, elección del nombre del bot

Fuente: Elaborado por el autor

Este nombre de usuario debe terminar con la palabra "bot" (p. ej., "myexamplebot"). Envía el nombre de usuario como un mensaje. Si el nombre de usuario elegido está disponible, BotFather creará el bot y le proporcionará un token. Este token se usará como entrada para este complemento, figura 59.



Figura 59. *Token de Telegram*

Fuente: *Elaborado por el autor*

La figura 60 muestra la configuración del complemento Vaya a “Dispositivos” y después ha “Complementos”. Haz clic en el botón “+” en la esquina superior derecha, elige la opción de Telegram y sigue estos pasos: Ingrese el token de su bot, Asigne un nombre a su complemento y Completa una descripción (opcional).

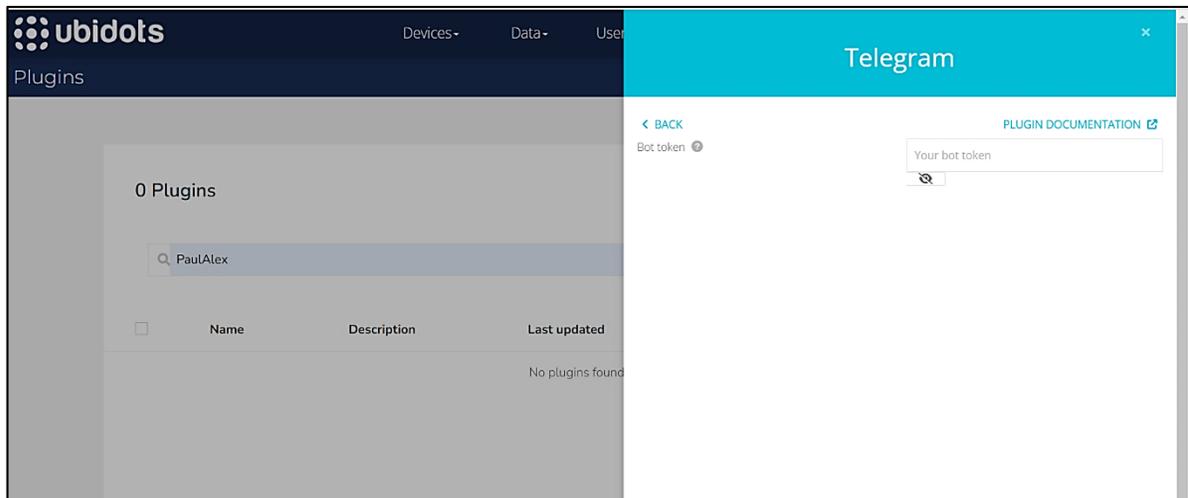


Figura 60. Configuración del complemento de Telegram en Ubidots

Fuente: Elaborado por el autor

Coloque la URL del punto final HTTPS de su complemento en el campo "URL" durante la configuración de "acciones". Puede ubicar esta URL ingresando a su complemento y luego dirigiéndose a su sección "decodificador", figura 61.

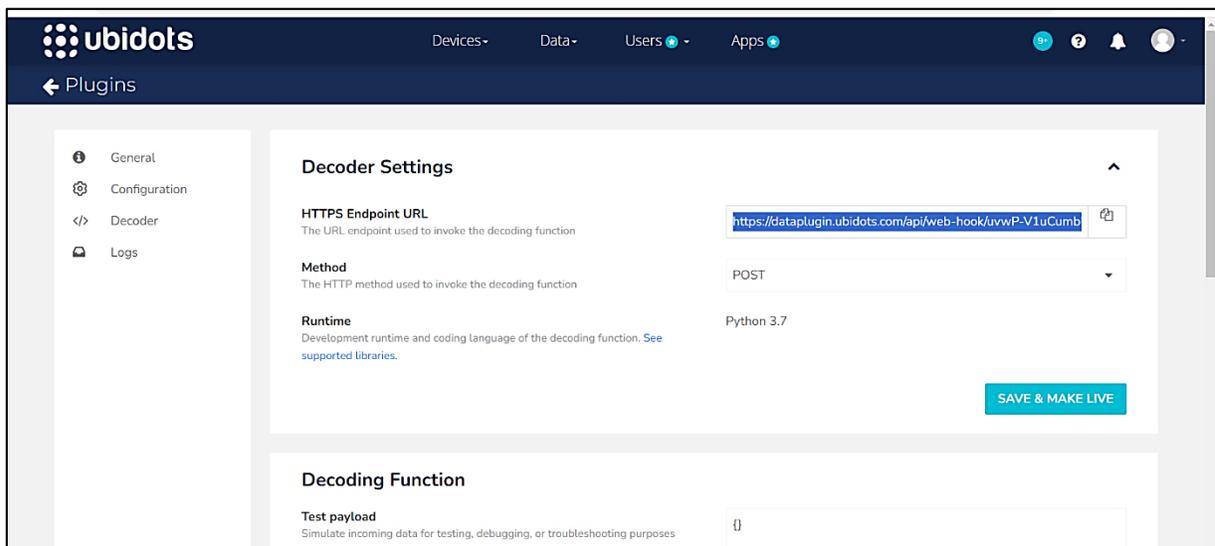


Figura 61. URL de Telegram para crear las alertas.

Fuente: Elaborado por el autor

Cambie la carga útil predeterminada con este JSON, figura 62:

```
{"nombre del grupo": "<NOMBRE DEL GRUPO>", "mensaje": "<MENSAJE>"}
```

Dónde:

<GROUPNAME>es el nombre del grupo objetivo de Telegram. Esta entrada distingue entre mayúsculas y minúsculas, así que asegúrese de escribirla/copiarla exactamente como aparece en Telegram.

<MESSAGE>es el contenido del mensaje a enviar.

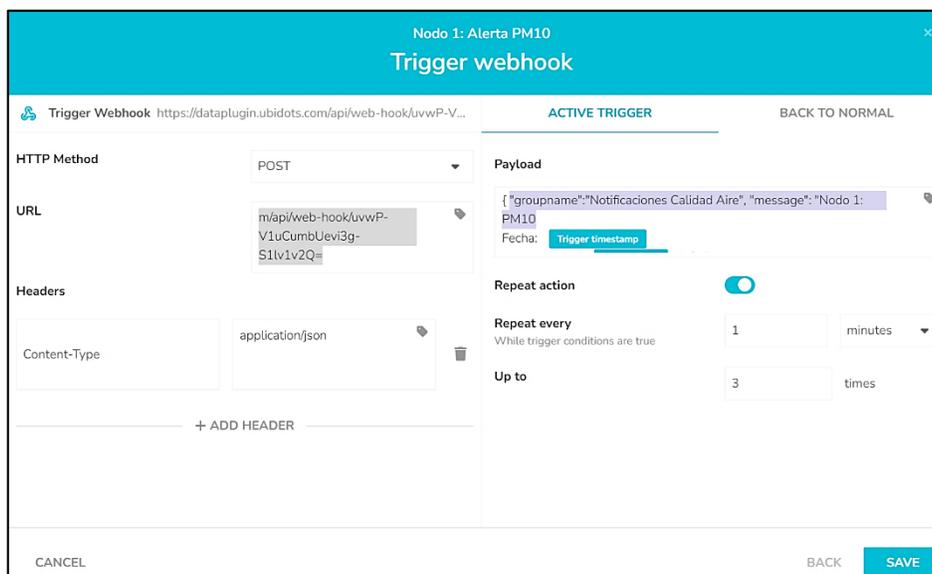


Figura 62. Configuración de las alertas para Telegram en Ubidots.

Fuente: Elaborado por el autor

Finalmente, después de haber clonado las variables y haber creado los eventos se procedió a crear el tablero donde se muestra los valores enviados de los medidores de calidad de aire, seleccionando entre los varios widgets disponibles que se muestran en la figura 63 para que posteriormente quede la aplicación creada en Ubidots como se muestra en la figura 64.

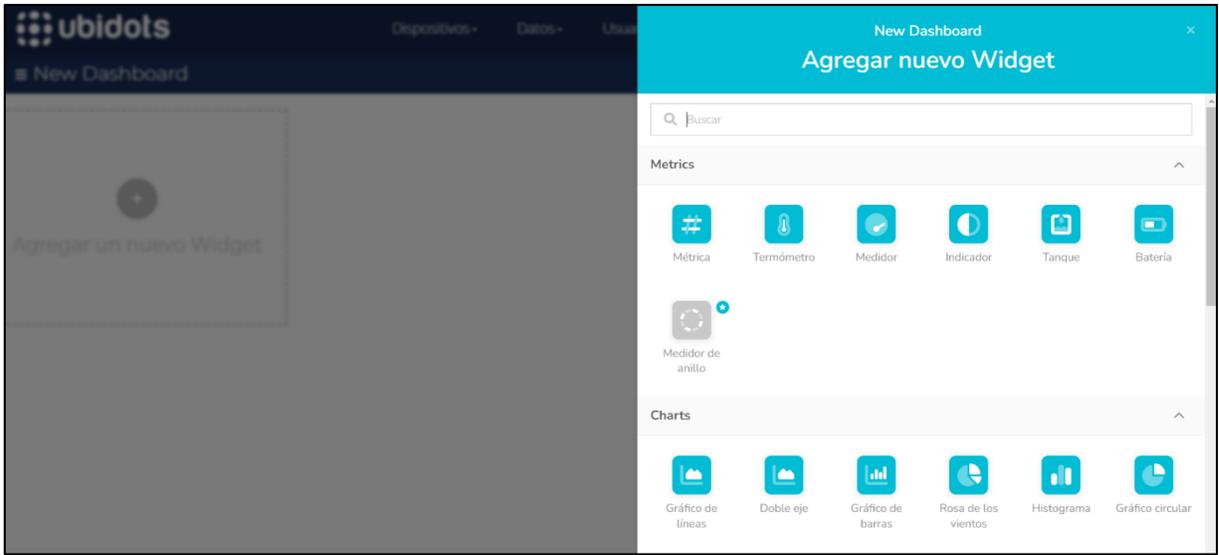


Figura 63. Widget disponible en Ubidots.

Fuente: Elaborado por el autor

Nota: Se seleccionaron los widgets metrics: tank y widget charts: line chart para cada uno de los gases.



Figura 64. Tablero Web que presenta los valores de los medidores de calidad de aire.

Fuente: Elaborado por el autor

5.3.8. *Calibración de sensores*

Para poder utilizar los sensores primeramente deben pasar por una fase de calentamiento, lo recomendable conectarlos de 24 a 48 horas a la corriente, para eliminar las impurezas de fábrica. Esto se lo realiza una sola vez cuando los vamos a utilizar por primera vez, las demás veces a utilizar con 20 segundos previos de calentamiento ya están aptos para ser utilizados, figura 65.

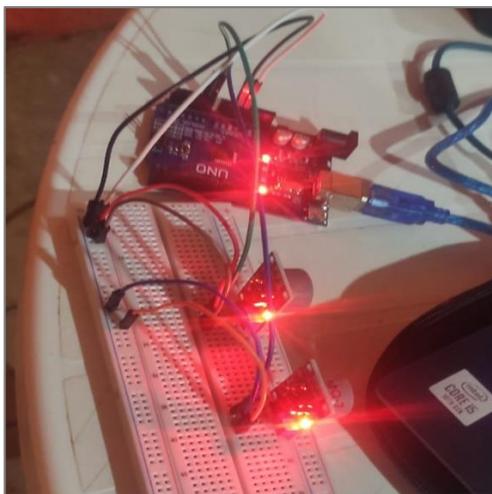


Figura 65. *Calentamiento de los Sensores.*

Fuente: *Elaborado por el autor*

Una vez culminado el proceso de calentamiento de los sensores, se procede a la calibración, la cual se la realizo mediante código en la plataforma digital de arduino IDE. La conexión de los sensores al arduino UNO, como es el voltaje, tierra; se utilizó la entrada analógica del sensor para poder calibrar los sensores.

Terminado de configurar, conectar los dispositivos y montarlos en sus respectivas cajas, se obtiene como resultado las siguientes figuras 66 y 67 donde se muestran los nodos listos para ser ubicados y realizar las mediciones.

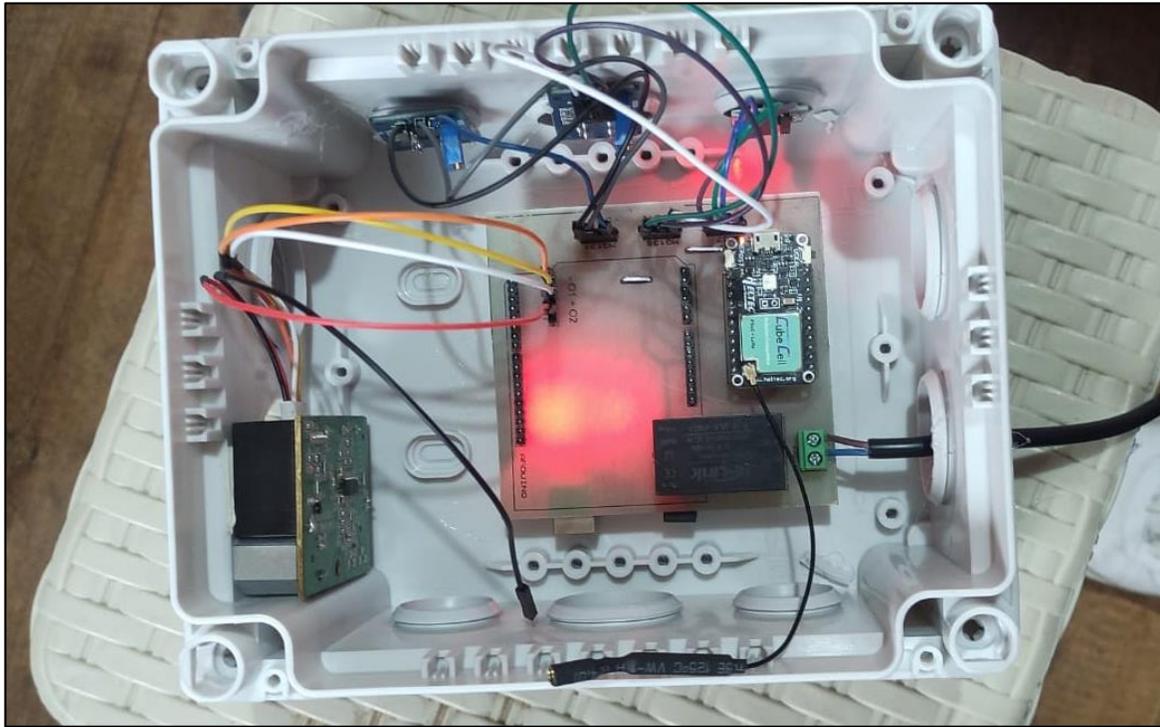


Figura 66. Estructura interna del Nodo Sensor

Fuente: Elaborado por el autor



Figura 67. Estructura externa del Nodo Sensor

Fuente: *Elaborado por el autor*

5.3.9. Materiales de software

Los materiales a utilizar para el desarrollo del siguiente proyecto, se los describe a continuación, clasificándolos por software y hardware.

En la tabla 8, se describen los materiales de software utilizados

Tabla 8. *Materiales de Software*

Descripción	Cantidad
Arduino IDE	1
The Things Network (TTN)	1
Ubidots (plataforma IoT)	1

5.3.10. Materiales de hardware

En la tabla 9, se describen los materiales de hardware utilizados

Tabla 9. *Materiales de Hardware*

Descripción	Cantidad
Gateway Mikrotik wAP LR9	1
Módulo heltec cube cell	2
Sensor MQ-7	2
Sensor MQ-131	2
Sensor MQ-136	2
Sensor DSM501A	2
Arduino UNO	2
Computadora	1

6. Resultados

Los resultados que serán presentados fueron obtenidos mediante la recolección de información de dos nodos sensores que fueron ubicados en el terminal terrestre de Loja y en el mercado mayorista, en la figura 68 y 69 se observa cada nodo sensor ubicado en su respectivo lugar con su ubicación.

El nodo sensor 1 se encuentra ubicado en la avenida gran Colombia e Ibarra en un local de celulares, sus coordenadas son: longitud: $3^{\circ} 59' 17''$ S y latitud: $79^{\circ} 12' 23''$ O.

a)



b)

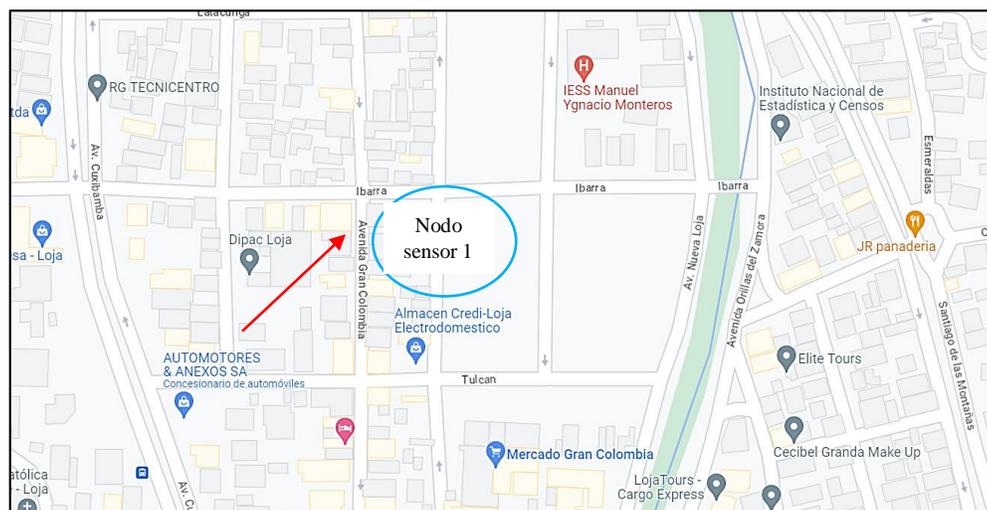


Figura 68. a) Ubicación del nodo sensor 1 mercado mayorista b) Ubicación en google maps

Fuente: *Elaborado por el autor*

El nodo sensor 2 se encuentra ubicado en la avenida 8 de diciembre en un local comercial, cuyas coordenadas son: longitud: 3° 58' 42" S y latitud: 79° 12' 22" O.



Figura 69. a). Ubicación del nodo sensor 2 Terminal Terrestre Loja b). Ubicación en google maps

Fuente: *Elaborado por el autor*

Para procesar los valores arrojados por los nodos sensores y almacenarlos, se utilizó la plataforma Ubidots que esta ayuda a enviar datos a la nube desde cualquier dispositivo

conectado a internet, permitiendo visualizar y procesar esos datos en tiempo real, aparte de visualizar la información también permite configurar acciones y/o alertas basadas en análisis de datos.

La figura 70, presenta los estándares de como presentar los datos siendo estos: en bruto, promedio, mínimo, máximo, suma y conteo.

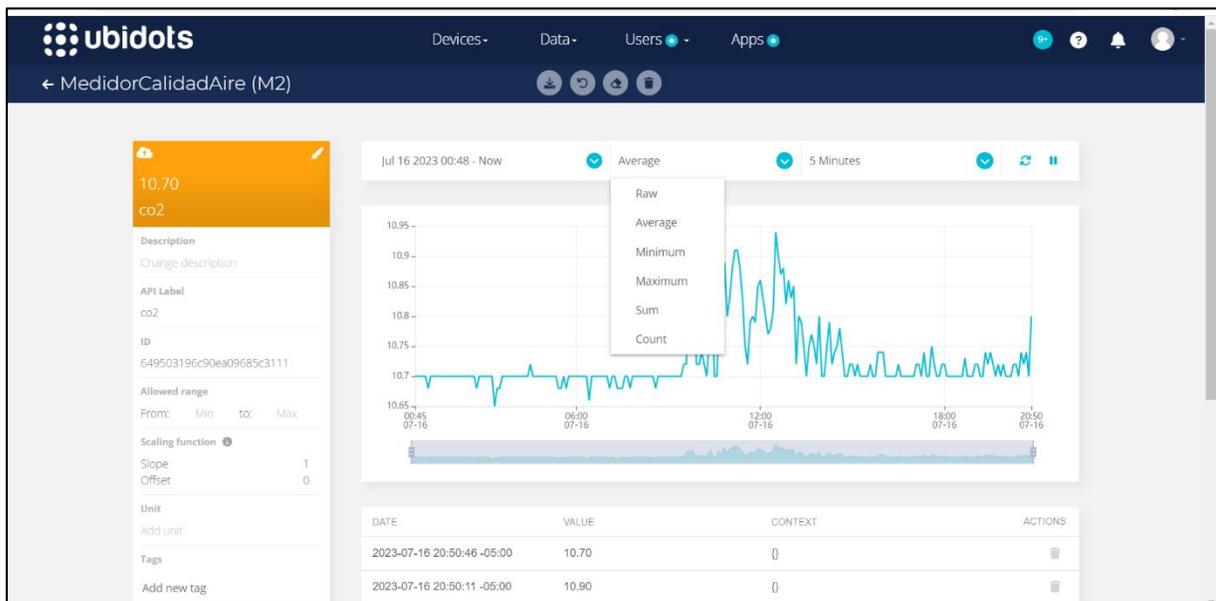


Figura 70. Dashboard de Ubidots de los estándares de visualización

Fuente: Elaborado por el autor

Ubidots también nos permite configurar los periodos de tiempo para la visualización de los datos, en lapsos de tiempo: 5 minutos, 30 minutos, 1 hora, 6 horas y un día. Como se muestra en la figura 71.

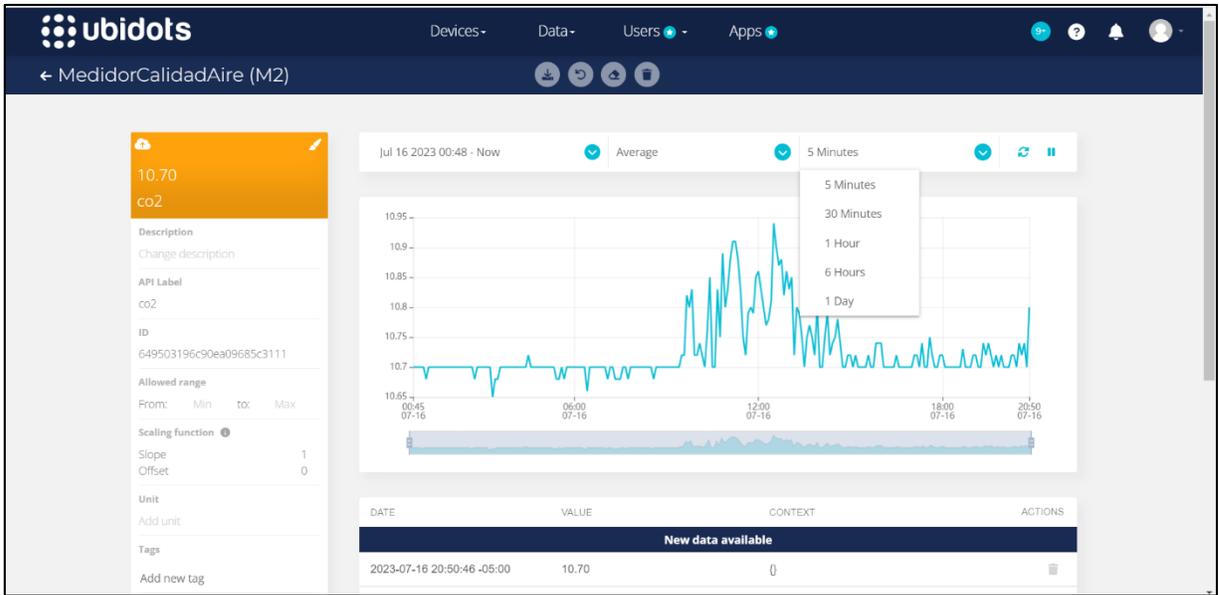


Figura 71. Dashboard de Ubidots de los periodos de tiempo para la visualización de los datos

Fuente: Elaborado por el autor

Como se observa en la figura 72 y 73, se puede configurar alertas de envío de información, en dicha configuración se puede establecer los rangos máximos, el tiempo de medición, las repeticiones, los días, las horas, donde se activarán las alertas. Los servidores por los cuales se pueden enviar estas notificaciones son mediante: correo electrónico, mensajes de textos, Telegram, llamada telefónica entre otros.

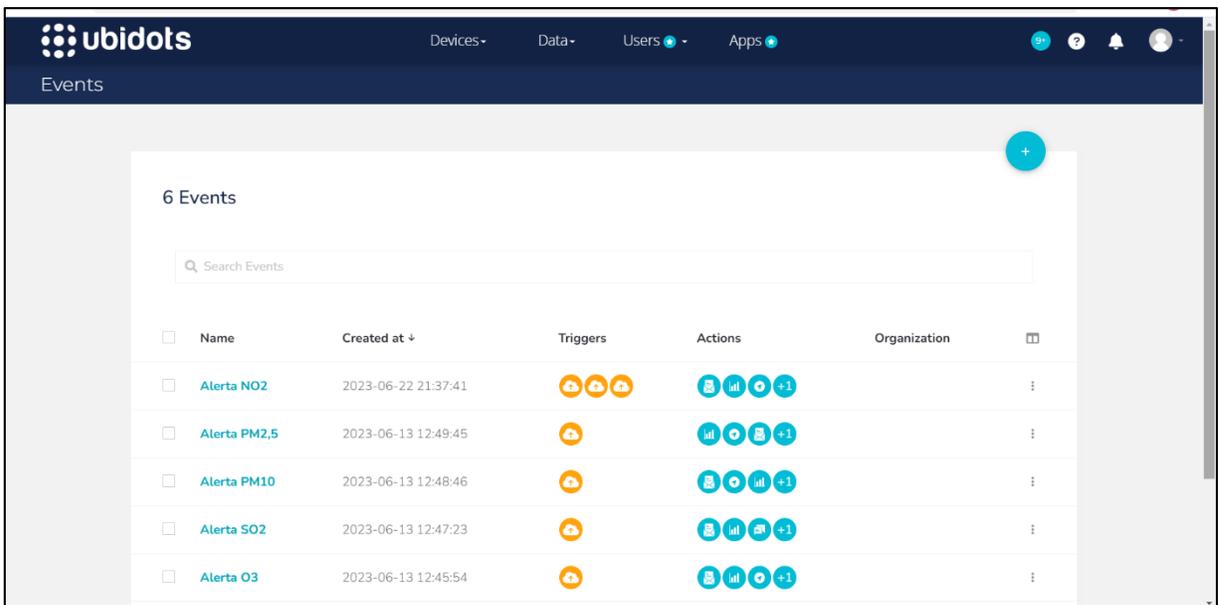


Figura 72. Dashboard de eventos de Ubidots de configuración de alertas

Fuente: *Elaborado por el autor*

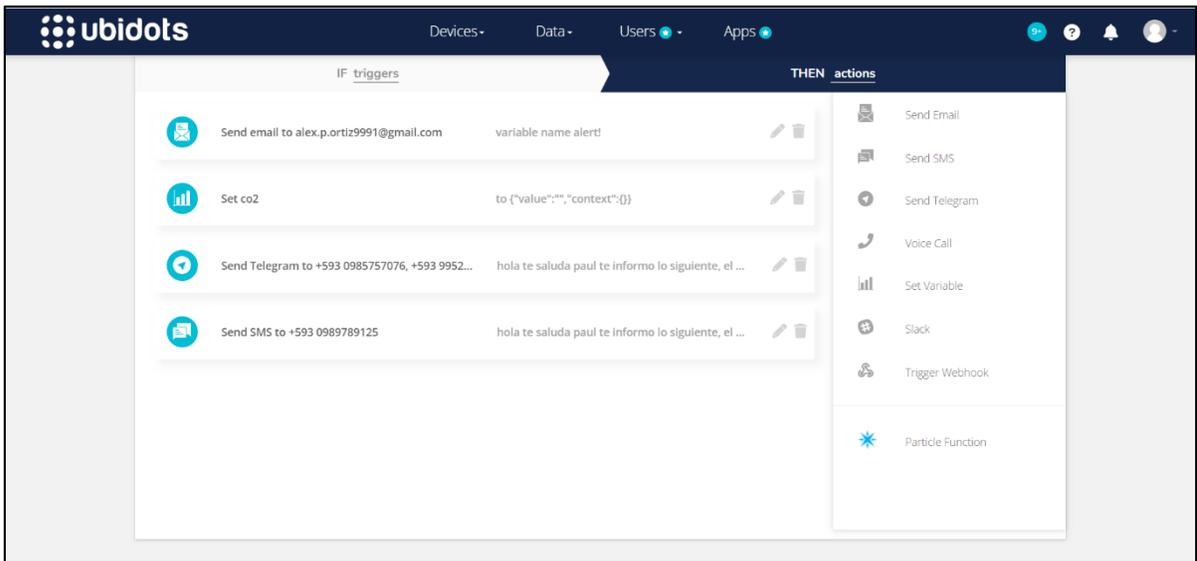


Figura 73. *Dashboard de Ubidots de los servidores de envío de alertas*

Fuente: *Elaborado por el autor*

A continuación, se presenta los resultados de los dos nodos sensores, primeramente, se muestra la información del nodo sensor 1 y posteriormente la información del nodo sensor 2.

Información Nodo Sensor 1

Se presenta el Dashboard de visualización de los valores de los gases obtenidos por el nodo sensor 1 ubicado en el mercado mayorista; mediante widgets, permitiendo una correcta interpretación de los datos, figura 74.

Los gases medidos son presentados en el siguiente orden: dióxido de carbono, ozono, dióxido de nitrógeno, dióxido de azufre, material particulado PM2,5 y PM10

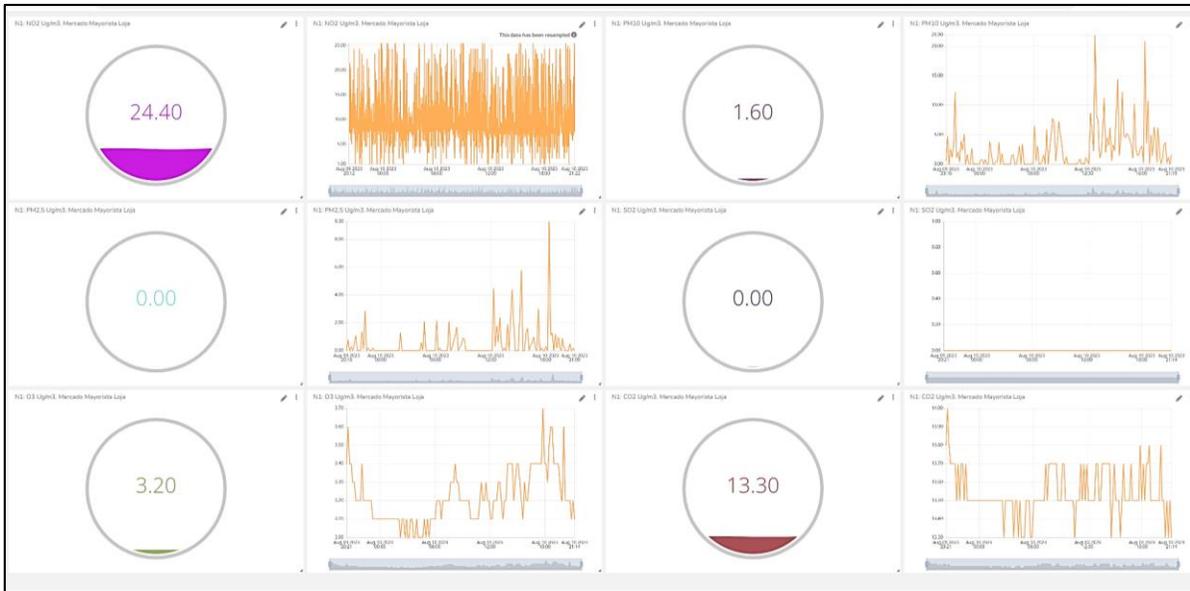


Figura 74. Dashboard de visualización de los datos del nodo 1 ubicado en el mercado mayorista

Fuente: Elaborado por el autor

A continuación, se presenta de manera detalla cada uno de los gases medidos y procesados, a través del nodo sensor 1 ubicado en el mercado mayorista en la plataforma de Ubidots:

Dióxido de Carbono (CO2)

En la figura 75, se muestran los valores de CO2 arrojados por el nodo 1 ubicado en el mercado mayorista en un periodo de tiempo desde el 13 de junio al 13 de agosto.

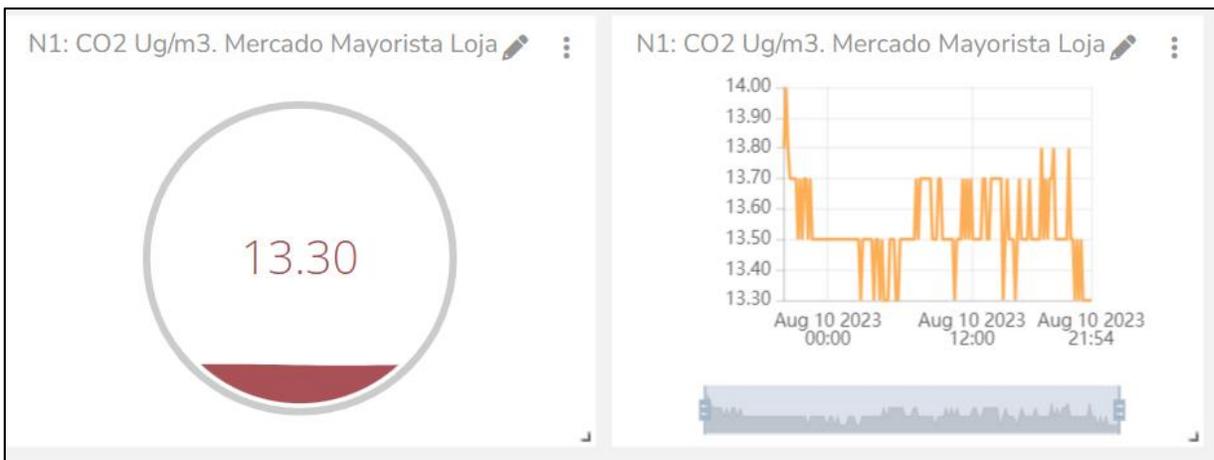


Figura 75. Widgets de medición del gas CO2 en el mercado mayorista

Autor: Elaborado por el autor

La figura 76, presenta la gráfica de los valores mínimos obtenidos del nodo 1 ubicado en el mercado mayorista, el valor mínimo es de 10,70 Ug/m3.

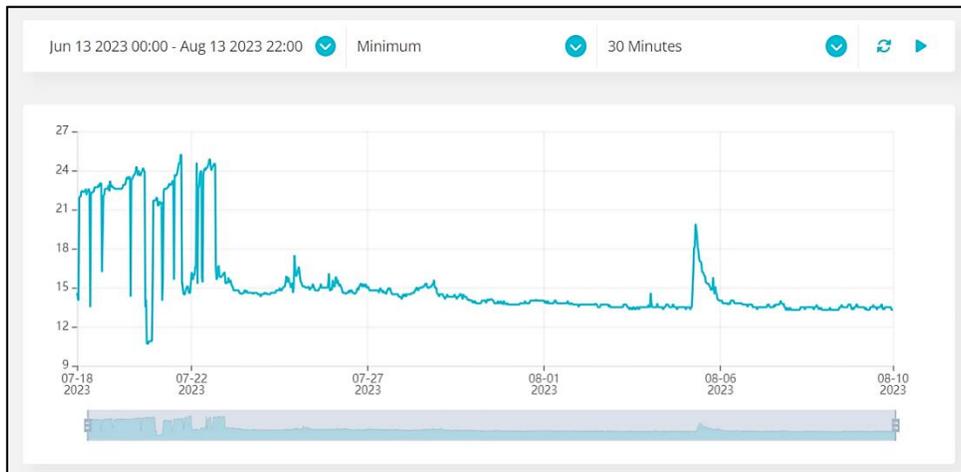


Figura 76. Valores mínimos obtenidos del CO2

Autor: Elaborado por el autor

El valor máximo arrojado por el CO2 es de 25,60 Ug/m3, obtenido del nodo sensor 1, como se muestra en la figura 77.



Figura 77. Valores máximos obtenidos del CO2

Autor: Elaborado por el autor

En la figura 78, se pueden observar los datos obtenidos por el nodo 1, donde se destaca que la media del CO2 en períodos de 30 minutos es de 21,36 Ug/m3.

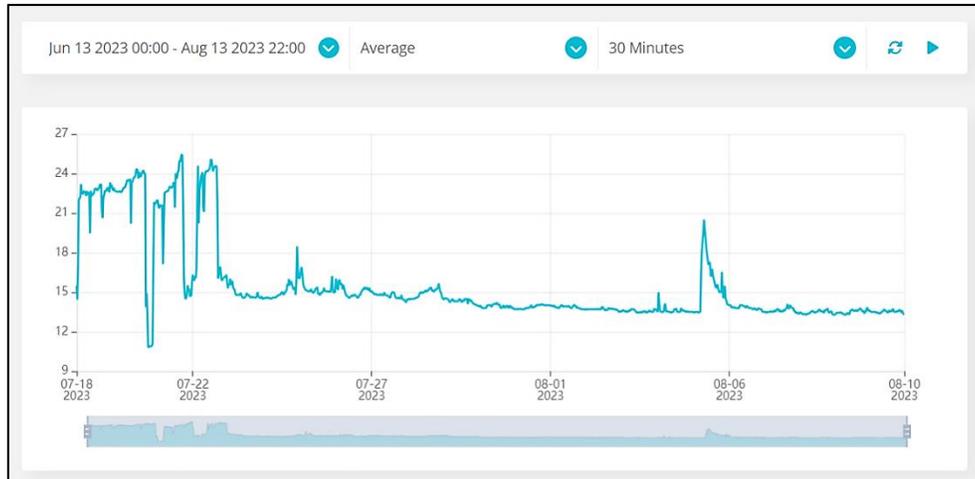


Figura 78. Valores promedio obtenidos del CO2

Autor: Elaborado por el autor

Ozono (O3)

Los valores de ozono registrados por el nodo sensor 1, ubicado en el mercado mayorista, durante el período comprendido desde el 13 de junio al 13 de agosto, a continuación, se muestra los widgets de visualización, figura 79.

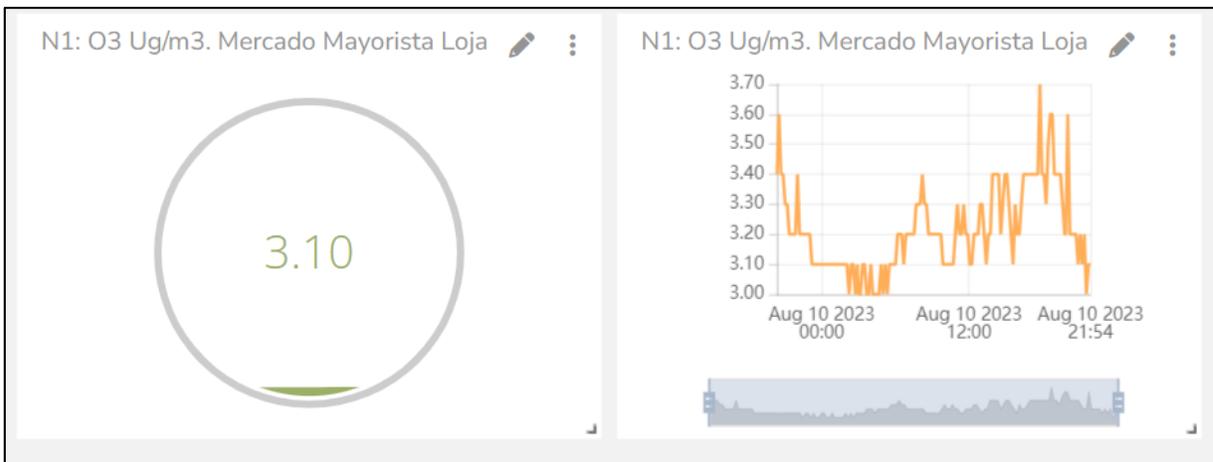


Figura 79. Widgets de Visualización de los datos obtenidos del O3

Fuente: Elaborado por el autor

La figura 80 muestra el rango de valores mínimos registrados por el ozono que varían entre los valores de 2,50 a 2,90 Ug/m3.

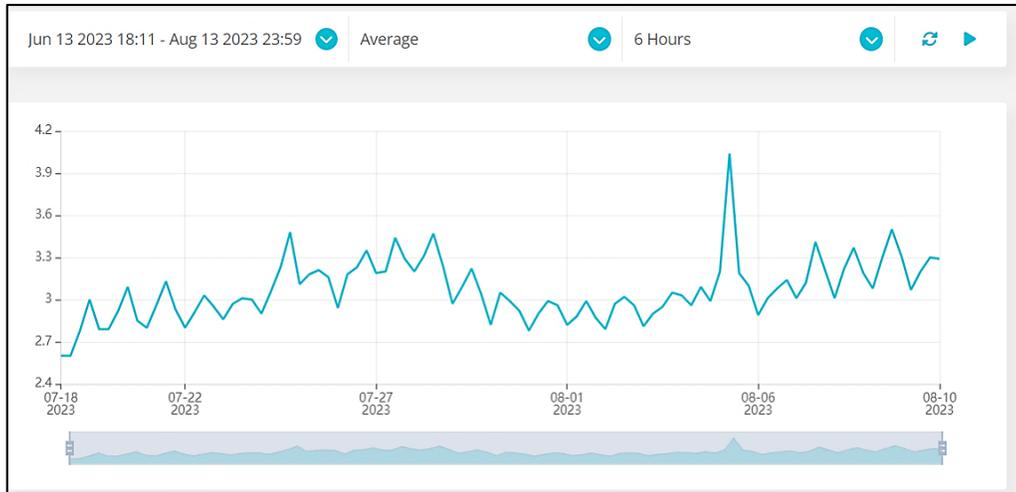


Figura 80. Valores mínimos obtenidos por el O3

Fuente: Elaborado por el autor

Como se puede visualizar en la figura 81 se presenta el rango de valores máximos arrojados por el O3 que varían desde 2,90 a 6,20 Ug/m3.

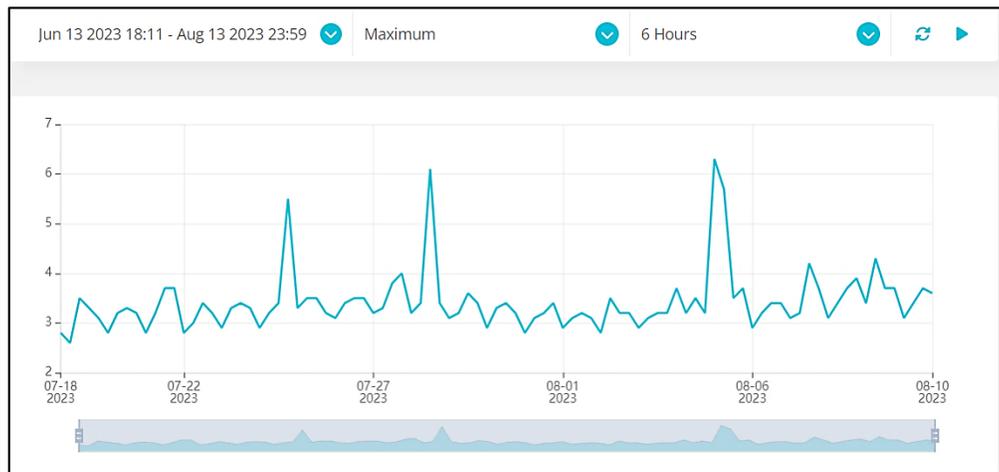


Figura 81. Valores máximos obtenidos del O3

Autor: Elaborado por el autor

El valor promedio del O3 obtenido del nodo sensor 1, considerando una media de 6 horas, es de 2,88 Ug/m3, figura 82.

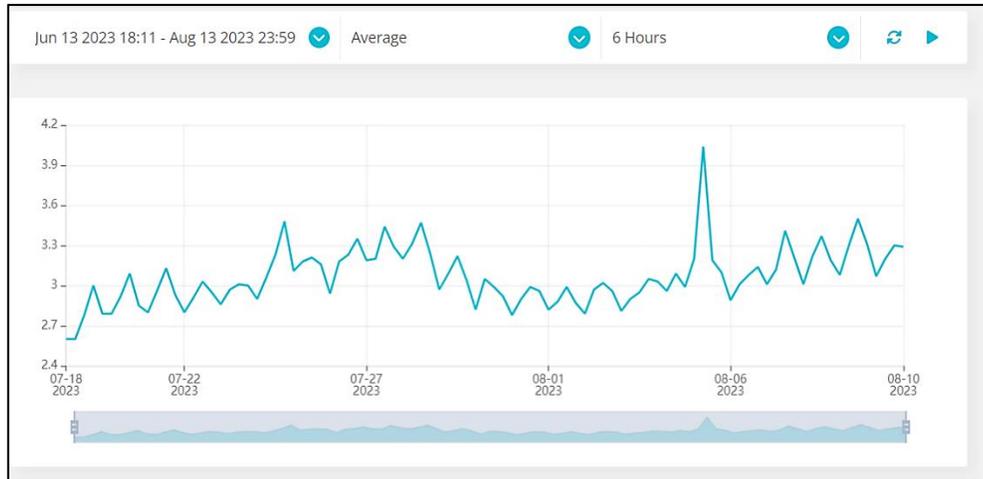


Figura 82. Valores promedio obtenidos del O3

Autor: Elaborado por el autor

Dióxido de Nitrógeno (NO2)

Se presenta los valores recibidos del dióxido de nitrógeno enviados desde el nodo 1, ubicado en el Mercado Mayorista desde el 13 de junio al 13 de agosto. La figura 83, se muestra la lectura actual y el historial de los datos.

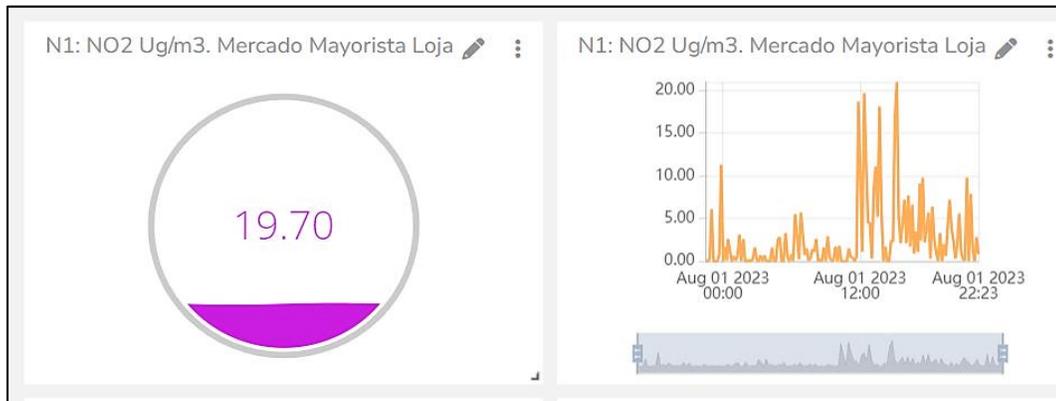


Figura 83. Dashboard de los gráficos a partir de los valores obtenidos del NO2

Autor: Elaborado el autor

El rango de valores mínimos obtenidos del NO2 van desde el 1 Ug/m3 a 10,50 Ug/m3 como se aprecia en la figura 84.

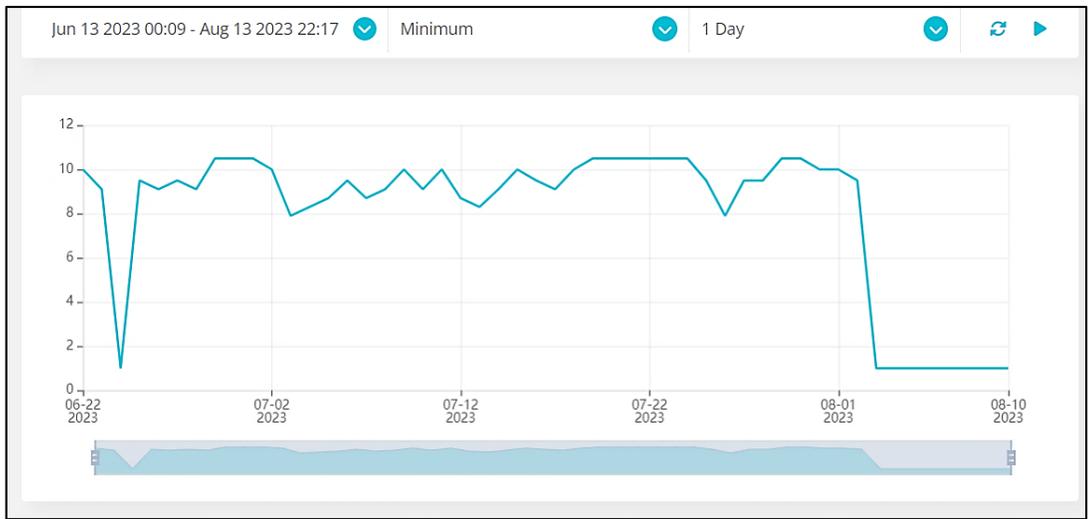


Figura 84. Valores mínimos obtenidos del NO2

Autor: Elaborado el autor

La figura 85, presenta el rango de valores máximos obtenidos del NO2 por el nodo 1 ubicado en el mercado mayorista, cuyos valores van de 19,70 Ug/m3 a 25,50 Ug/m3.

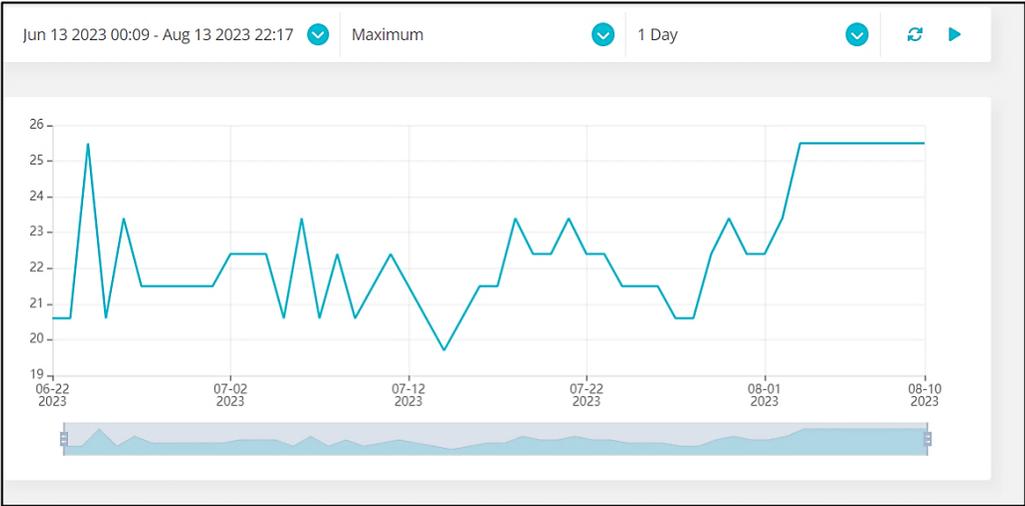


Figura 85. Valores máximos obtenidos del NO2

Autor: Elaborado el autor

El valor del NO2 en una media de 24 horas es de 15,95 Ug/m3 como se muestra en la figura 86.

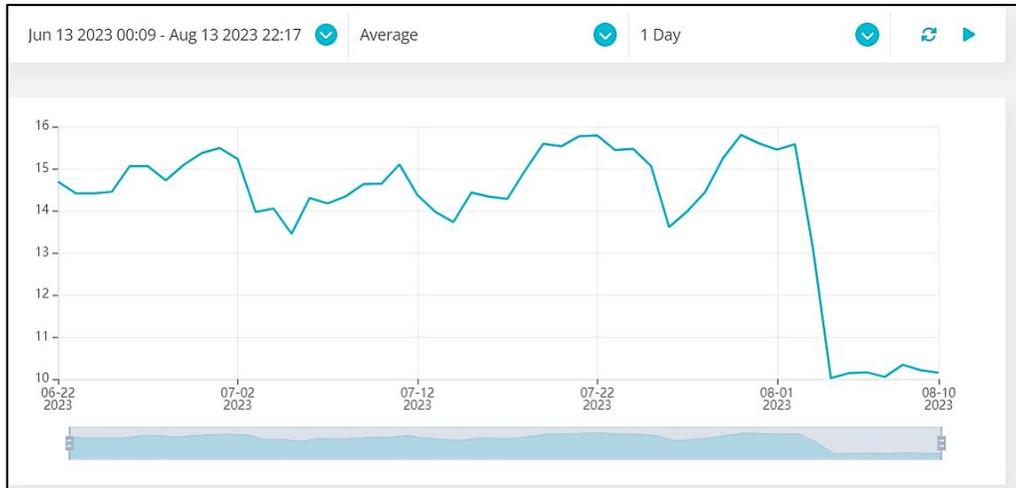


Figura 86. Media de los valores obtenidos del NO2

Autor: Elaborado el autor

Dióxido de Azufre (SO2)

El nodo sensor 1 no presenta valores del dióxido de azufre, por tal motivo no se adjuntó las gráficas de sus resultados.

Material particulado PM (2,5)

Los valores del material particulado PM_{2,5} obtenidos del nodo sensor 1, ubicado en el mercado mayorista, durante el período del 13 de junio al 13 de agosto, se presentan en el dashboard de la figura 87.

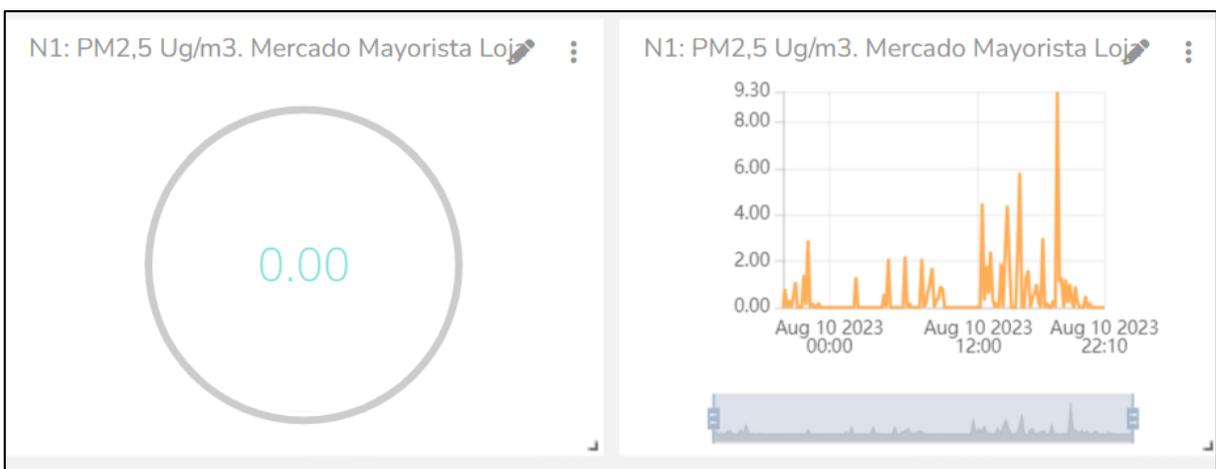


Figura 87. Widgets de visualización de los datos obtenidos del PM_{2,5}

Autor: *Elaborado el autor*

La figura 88 se presenta los valores mínimos obtenidos por el nodo 1, ubicado en el mercado mayorista, con un rango que va desde 0 Ug/m³ hasta 2 Ug/m³.



Figura 88. *Valores mínimos obtenidos del PM2,5*

Autor: *Elaborado el autor*

En la figura 89 se pueden observar los valores máximos arrojados por el nodo 1, los cuales varían desde 2 Ug/m³ hasta 22,30 Ug/m³, con un registro diario en periodos de 24 horas.

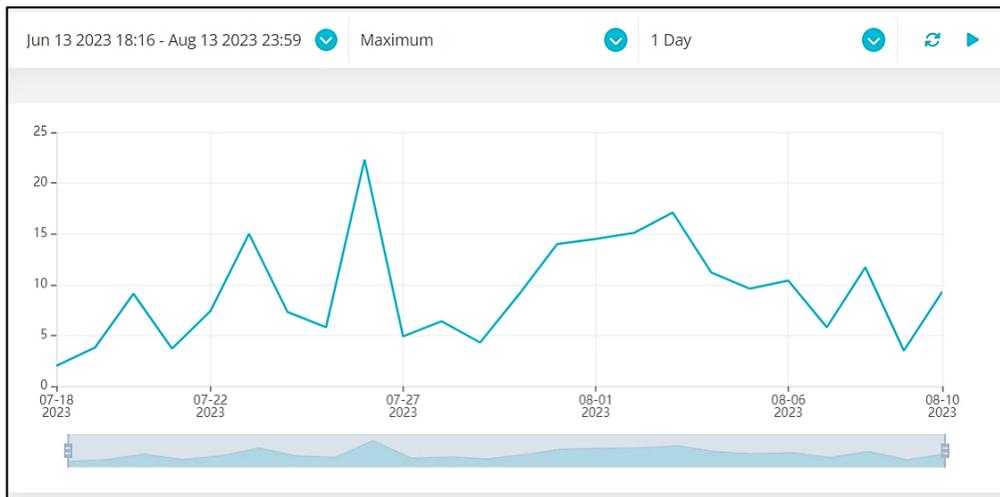


Figura 89. *Valores máximos obtenidos del PM2,5*

Autor: *Elaborado el autor*

La media de 24 horas arroja un valor de 0,33 Ug/m³, en la figura 90 se visualiza los datos obtenidos y almacenados en la plataforma de Ubidots.

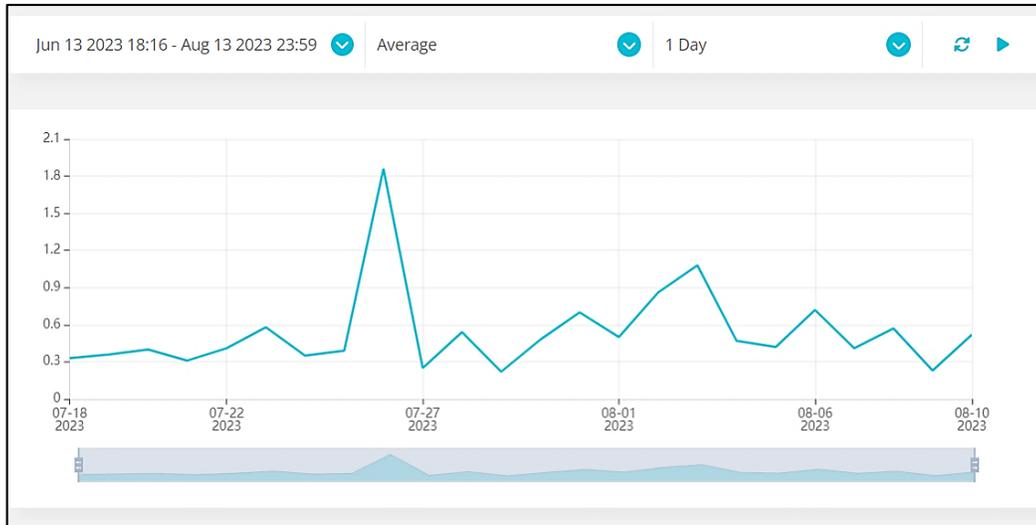


Figura 90. valores promedios obtenidos del PM2,5

Autor: Elaborado el autor

Material particulado (PM10)

Los datos obtenidos del PM10 registrados por nodo 1 ubicado en el mercado mayorista en los meses de junio a agosto, se obtuvo los siguientes valores presentados en la figura 91.

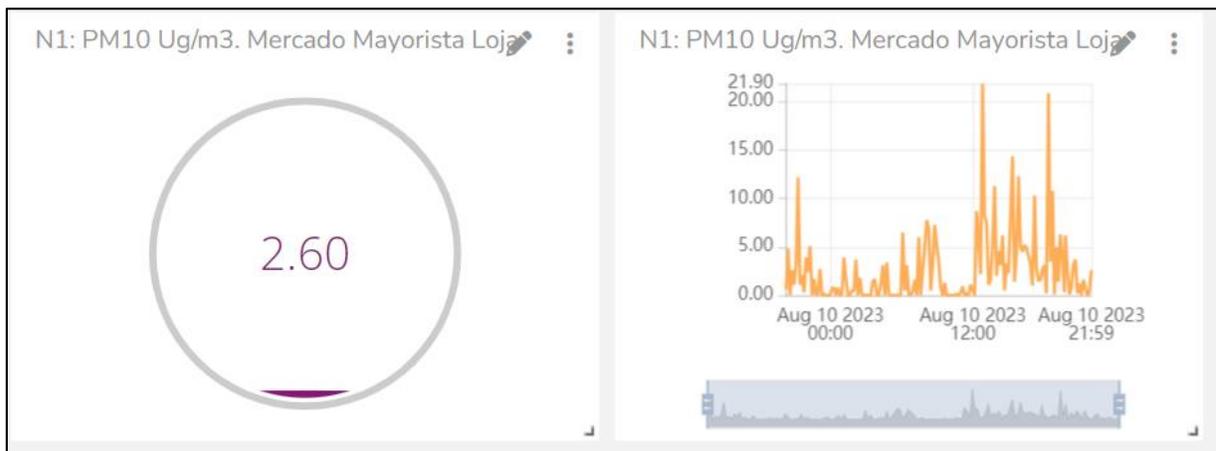


Figura 91. Widgets de visualización de los datos registrados del PM10

Autor: Elaborado por el autor

Los valores mínimos obtenidos del gas PM10 arrojados por el nodo 1, es 0 Ug/m3, como se puede visualizar en la figura 92.

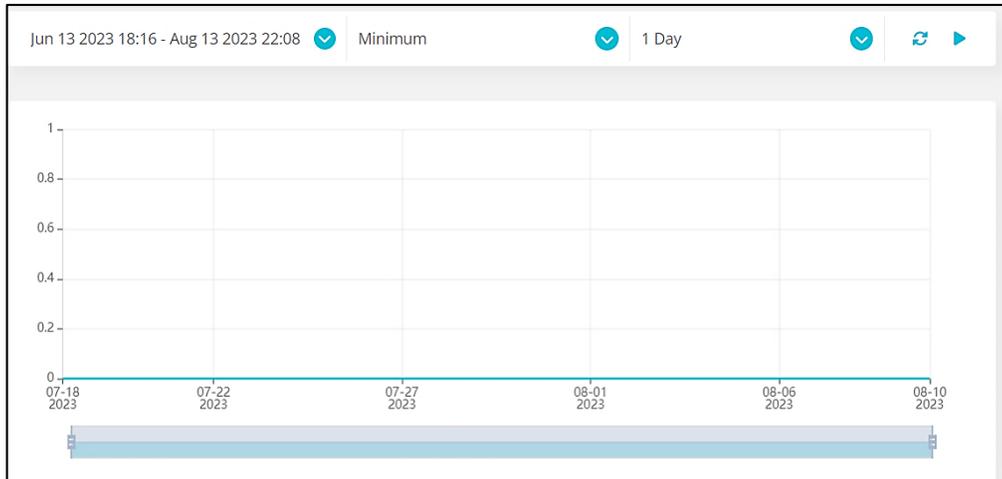


Figura 92. Valores mínimos obtenidos del PM10

Autor: Elaborado por el autor

En la figura 93, se presenta los valores máximos del PM10 que se encuentran en el rango de 5 a 25 Ug/m3.

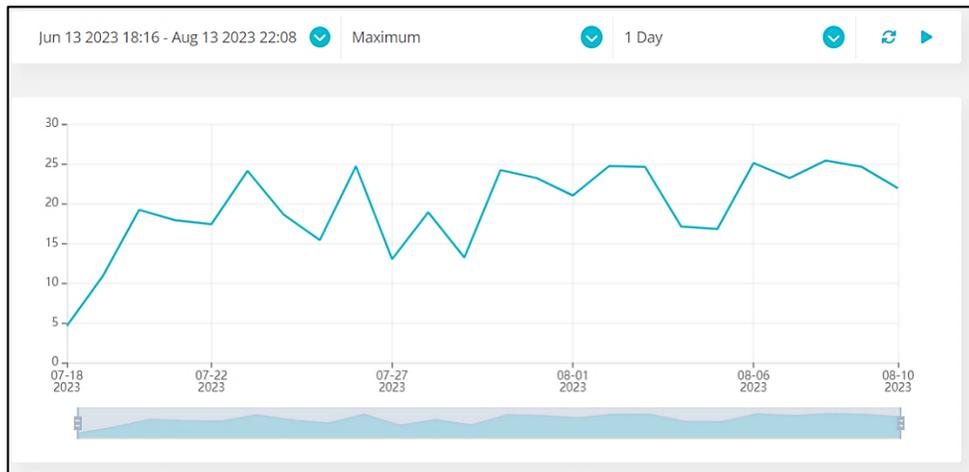


Figura 93. Valores máximos obtenidos del PM10

Autor: Elaborado el autor

El valor promedio registrado por el PM10 considerando una media de 24 horas, es de 1,67 Ug/m3, como se muestra en la figura 95.

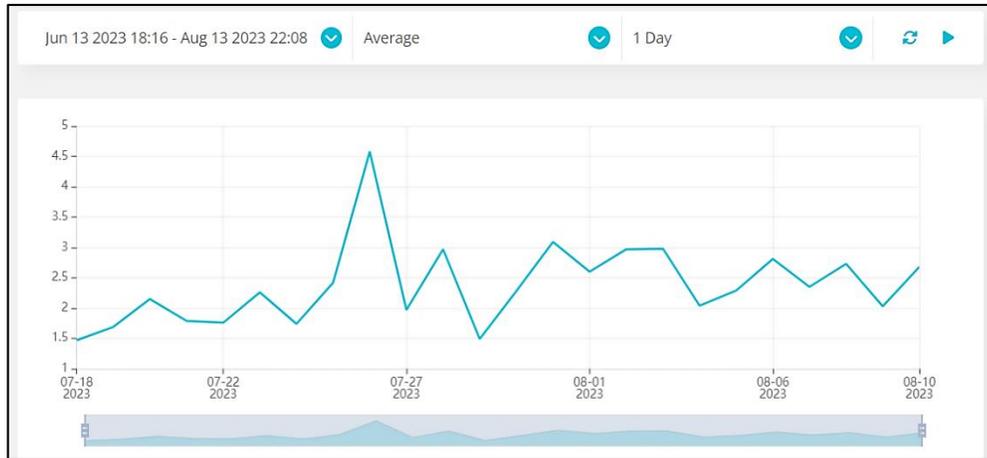


Figura 94. Valores promedio obtenidos del PM10

Autor: Elaborado el autor

A continuación, se muestra las notificaciones recibidas de cada uno de los gases, con lo que respecta al Nodo 1 ubicado en el mercado mayorista.

En la figura 95, se visualiza las notificaciones del dióxido de carbono (CO₂), recibidas de las alertas creadas en Ubidots tanto para el correo electrónico como para SMS y Telegram.

a)

NO **Notificaciones Calidad Aire**
 3 miembros

El valor mínimo es 10,70 Ug/m3.
 El valor máximo es 25,60 Ug/m3.
21:18

AlertasGases

Nodo 1: CO2

Fecha: 2023-07-22 21:28:04 -0500

Lectura Actual: 16.1 Ug/m3

En una media de 30 minutos 21,36 Ug/m3.

El valor mínimo es 10,70 Ug/m3.

El valor máximo es 25,60 Ug/m3.

21:28

AlertasGases

Nodo 1: CO2

Fecha: 2023-07-22 21:38:08 -0500

Lectura Actual: 16.4 Ug/m3

En una media de 30 minutos 21,36 Ug/m3.

El valor mínimo es 10,70 Ug/m3.

El valor máximo es 25,60 Ug/m3.

21:38

b)

Nodo 1: CO2Fecha: 2023-07-24
15:45:02 -0500Lectura Actual: 15.1
 Ug/m3En una media de 30 minutos ... El
 valor maximo es 25,60 Ug/m3

Nodo 1: CO2Fecha: 2023-07-24
15:55:07 -0500Lectura Actual: 15.6
 Ug/m3En una media de 30 minutos ... El
 valor maximo es 25,60 Ug/m3

Nodo 1: CO2Fecha: 2023-07-24
15:55:07 -0500Lectura Actual: 15.6
 Ug/m3En una media de 30 minutos ... El
 valor maximo es 25,60 Ug/m3

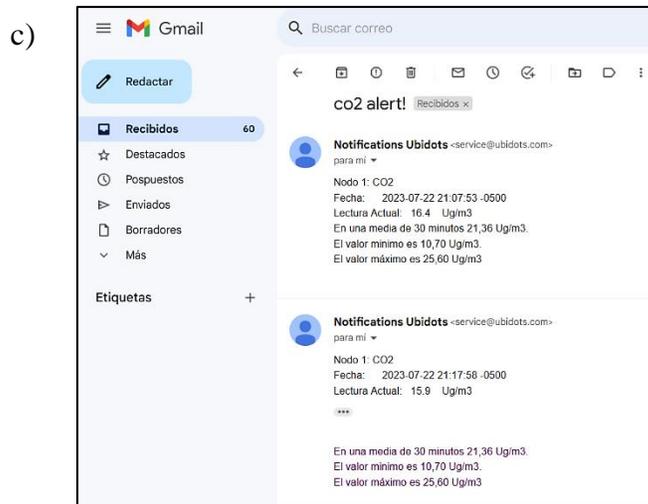
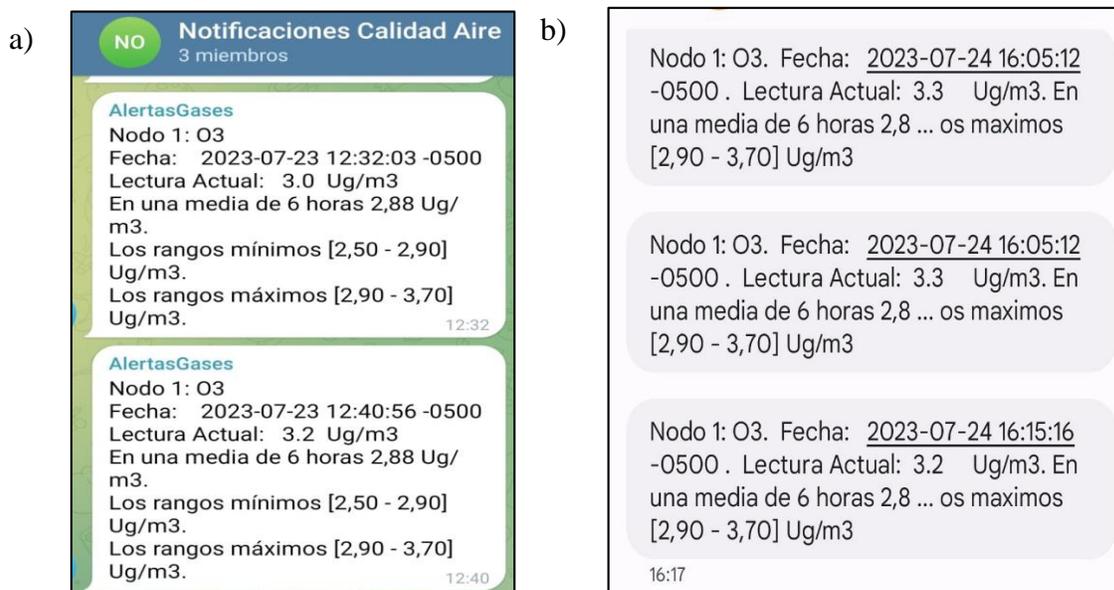


Figura 95. Notificaciones de alertas del CO₂; a) Telegram, b) SMS y c) Correo electrónico

Autor: Elaborado por el autor

Las notificaciones de Ozono(O₃) registradas en Telegram, SMS y correo electrónico se muestran en la figura 96 y son activadas en el momento en que se sobrepasa el valor máximo permisible.



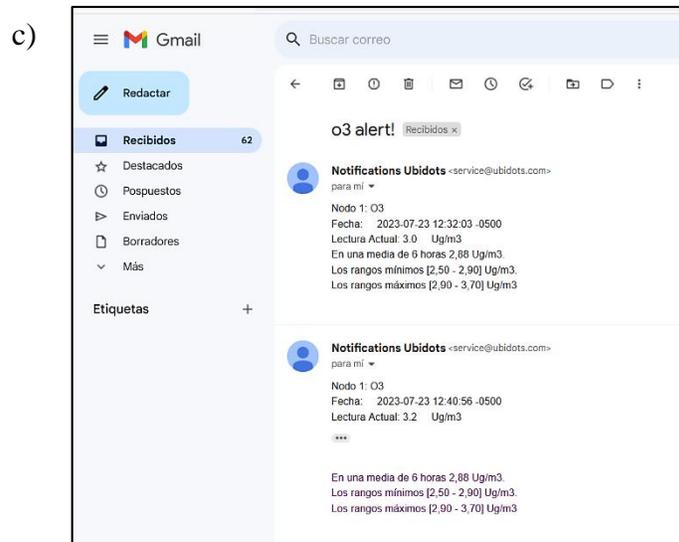
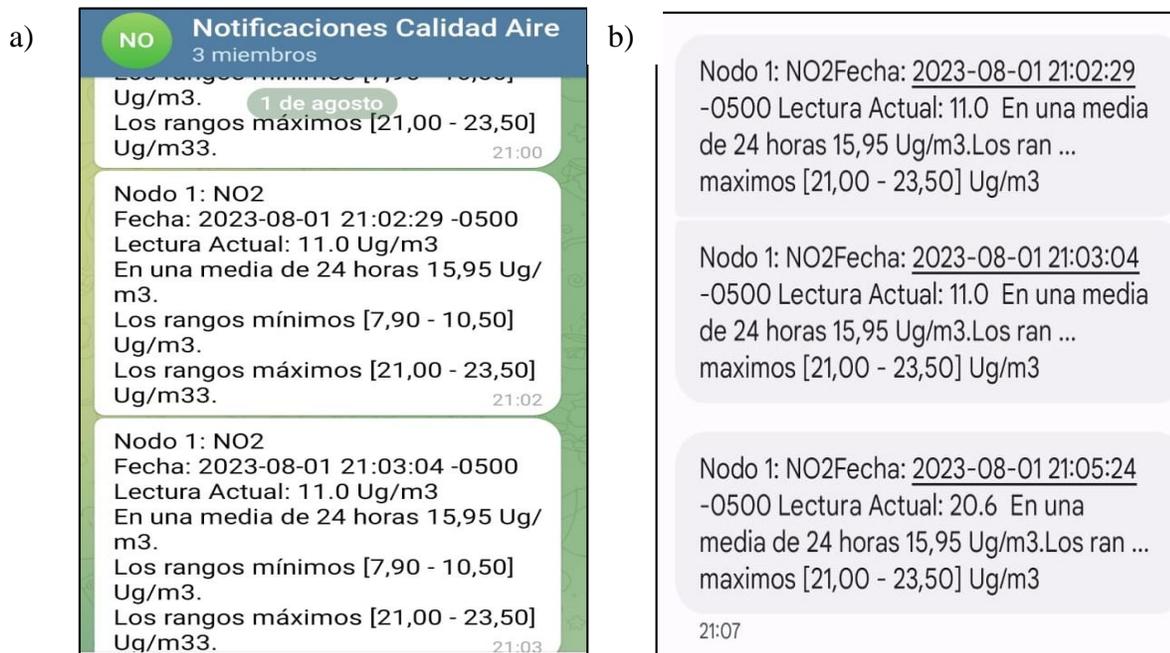


Figura 96. Notificaciones de alertas del O3; a) Telegram, b) SMS y c) Correo electrónico

Autor: Elaborado por el autor

En la figura 97, se presentan las notificaciones de alertas obtenidas para el Dióxido de Nitrógeno (NO₂). Estas alertas se generan cuando el gas tiene un valor más alto del permitido, y se envían inmediatamente mensajes a través de Telegram, SMS y correo electrónico



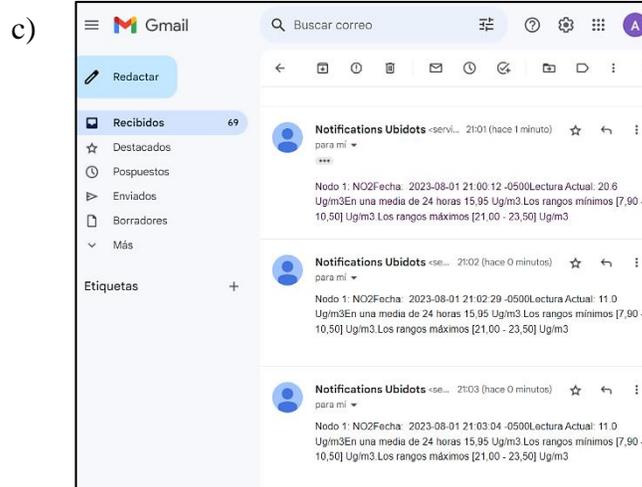
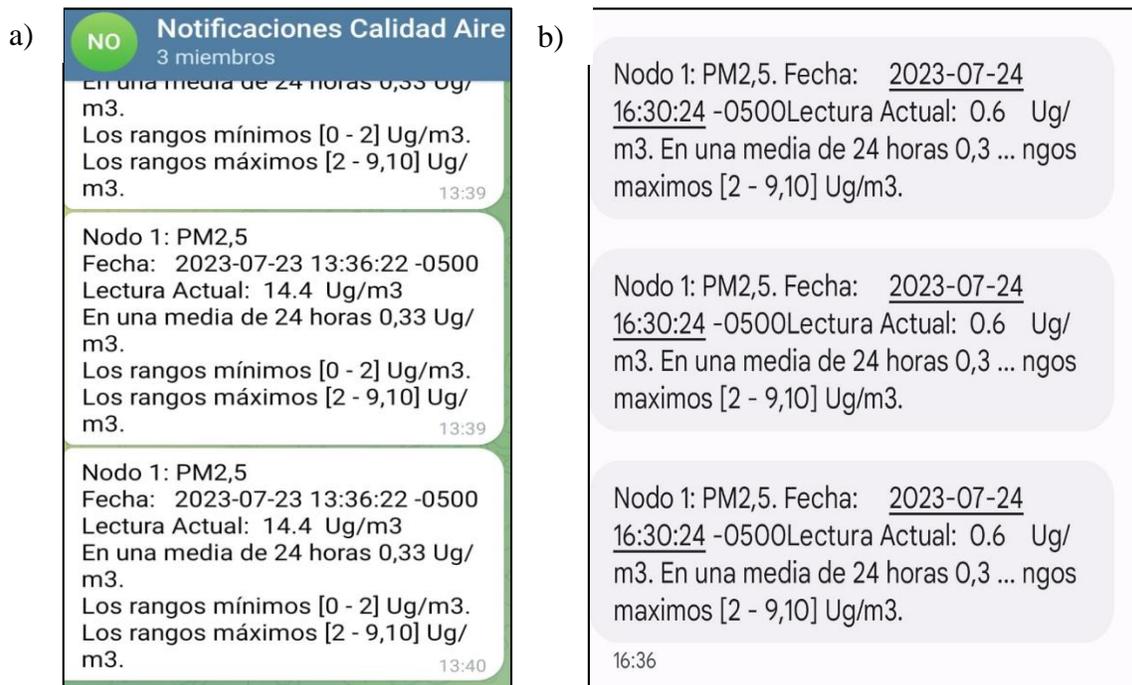


Figura 97. Notificaciones de alertas del NO₂; a) Telegram, b) SMS y c) Correo electrónico

Autor: Elaborado por el autor

En la figura 98, se visualizan las notificaciones obtenidas de las alertas del Material Particulado (PM_{2,5}). Estas notificaciones están configuradas para ser enviadas a través de Telegram, SMS y correo electrónico.



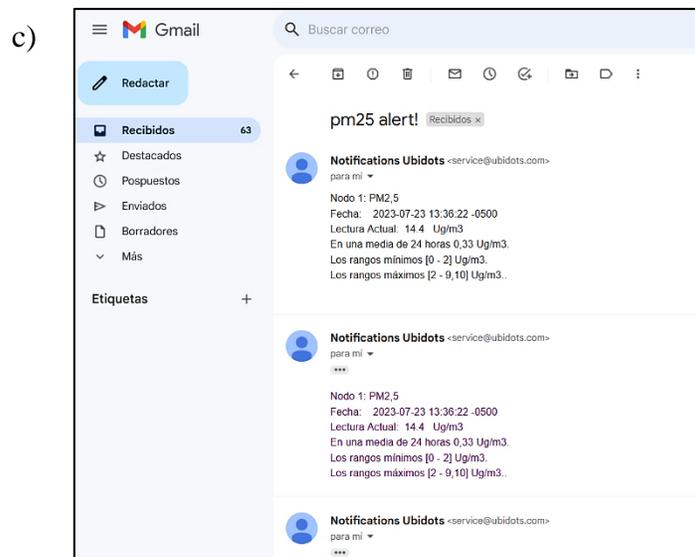
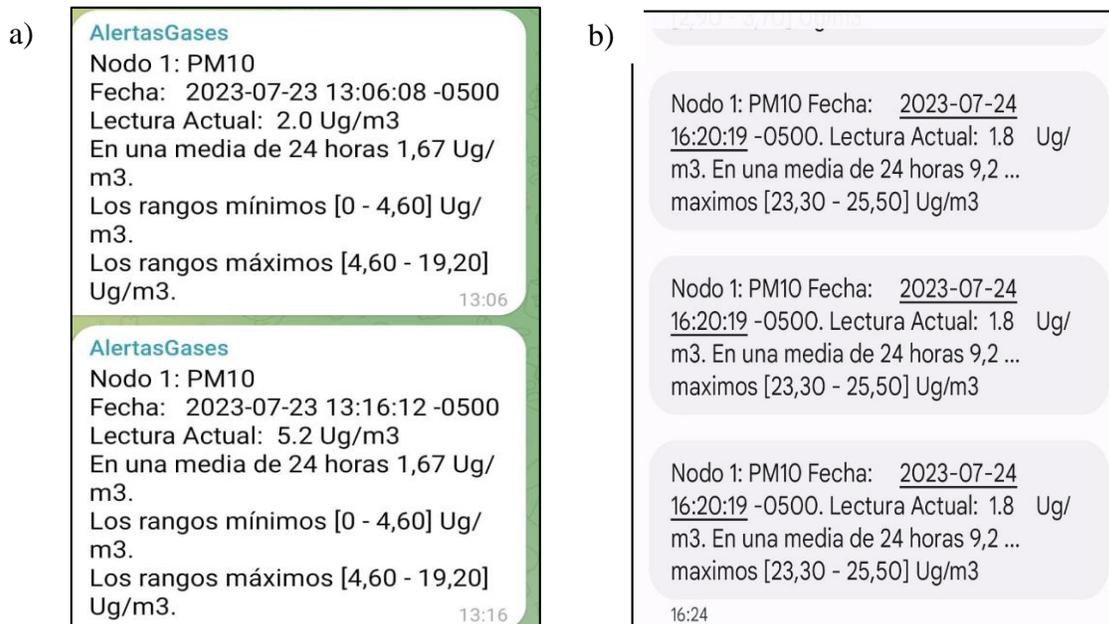


Figura 98. Notificaciones de alertas del PM2,5; a) Telegram, b) SMS y c) Correo electrónico

Autor: Elaborado por el autor

En la figura 99, se presentan las notificaciones de alertas obtenidas para el Material Particulado (PM10). Estas alertas se generan cuando el gas tiene un valor más alto del permitido, y se envían inmediatamente mensajes a través de Telegram, SMS y correo electrónico.



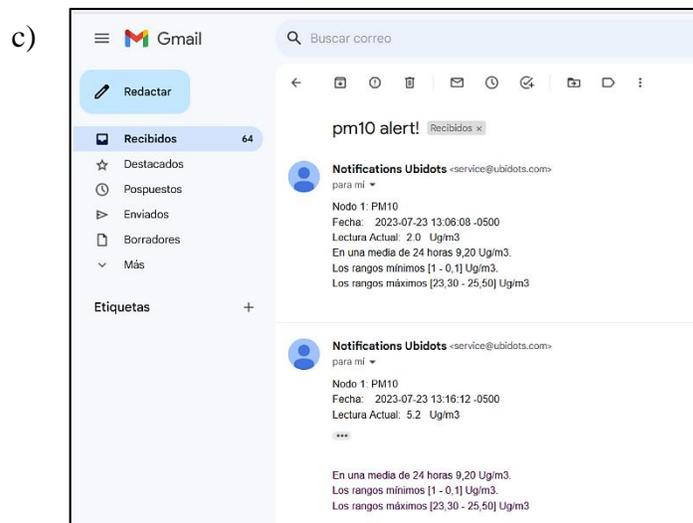


Figura 99. *Notificaciones de alertas del PM10; a) Telegram, b) SMS y c) Correo electrónico*

Autor: *Elaborado por el autor*

Es importante señalar que en el nodo 1 ubicado en el mercado mayorista el dióxido de azufre no presento valores, debido que no existió contaminación de este gas, por tal motivo no se agregó los resultados de este gas y tampoco se generó las alertas.

Información Nodo Sensor 2

En la figura 100, se presenta el Dashboard de manera detalla los gases medidos y procesados, a través del nodo sensor 2 ubicado en el terminal terrestre en la plataforma de Ubidots los gases medidos son presentados en el siguiente orden: dióxido de carbono, ozono, dióxido de nitrógeno, dióxido de azufre, material particulado PM2,5 y PM10



Figura 100. Dashboard de Ubidots para visualizar los datos de los gases.

Fuente: Elaborado por el autor

Dióxido de carbono (CO2)

En la figura 101, se muestran los valores de CO2 arrojados por el nodo 2 ubicado en el terminal terrestre en un periodo de tiempo desde el 13 de junio al 13 de agosto.

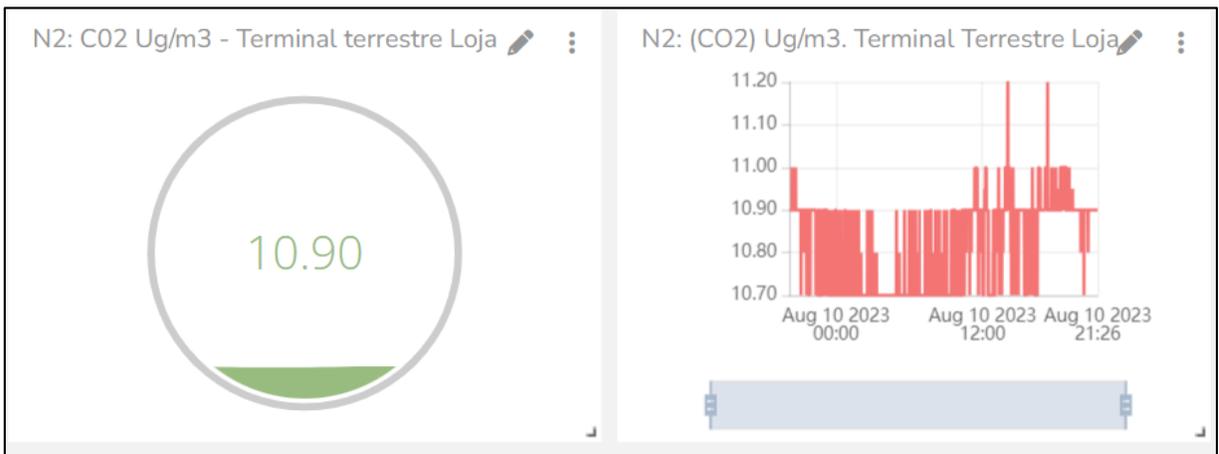


Figura 101. Widgets de medición del gas CO2 en el terminal terrestre

Fuente: Elaborado por el autor

En la siguiente figura 102, se observa los valores mínimos del gas CO2, obtenidos con el nodo 2 en el terminal terrestre. El rango de valores mínimos que se obtuvo va desde 10.50 Ug/m3 a 10.90 Ug/m3, en lapsos de tiempo de cada 30 minutos.

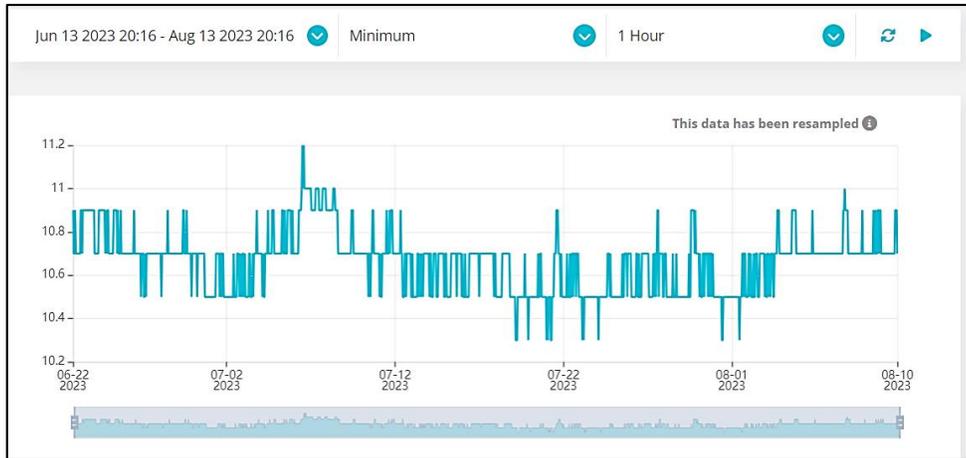


Figura 102. Valores mínimos de los valores arrojados de CO2

Fuente: Elaborado por el autor

En la figura 103, muestra los valores máximos de CO2, obtenidos con el nodo 2 ubicado en el terminal terrestre. El rango de valores máximos que se ha ido obteniendo va desde 10.90 Ug/m3 a 13.70 Ug/m3, en lapsos de tiempo de cada 30 minutos.

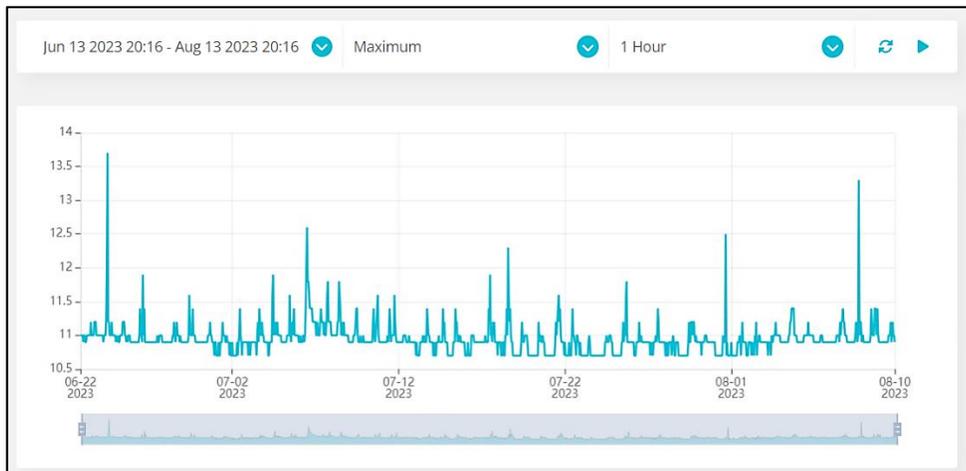


Figura 103. Valores máximos arrojados por el CO2

Fuente: Elaborado por el autor

La media cada 30 minutos del CO2 es de 10,83 Ug/m3, como se muestra en la figura 104.

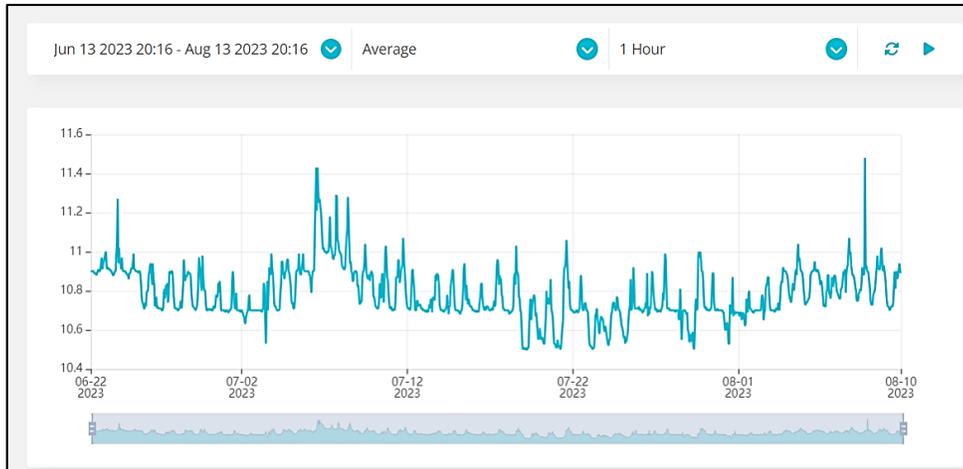


Figura 104. Valor promedio del CO2

Fuente: Elaborado por el autor

Ozono (O3)

A continuación, se detalla los valores de ozono obtenidos del nodo 2 ubicado en el terminal terrestre desde el 13 de junio al 13 de agosto. En la figura 105, en la parte izquierda se presenta el widget donde se visualiza el valor de los datos recibidos, los cuales se actualizan constantemente y al lado derecho se muestra un gráfico de línea para comprender de mejor manera el almacenamiento de los datos.

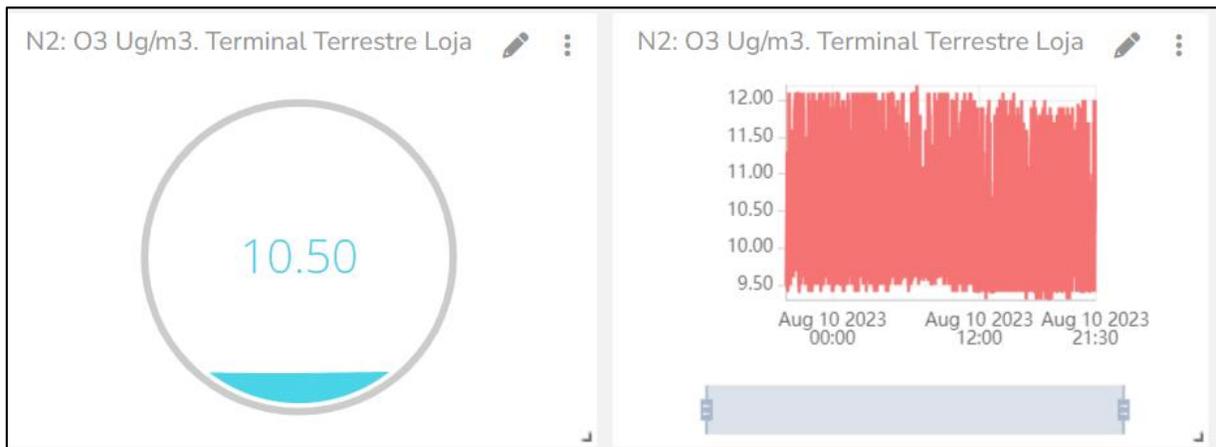


Figura 105. Widgets de visualización de los datos de O3

Fuente: Elaborado por el autor

Como se puede visualizar en la figura 106, se muestra la gráfica de los valores mínimos obtenidos del nodo 2, siendo estos valores comprendidos en el rango de 9,60 Ug/m³ y 10,10 Ug/m³. Los periodos de tiempos de cada dato presentado es cada 6 horas.



Figura 106. Valores mínimos obtenidos de O₃

Fuente: Elaborado por el autor

Los rangos de los valores máximos de O₃ se encuentran entre 11 Ug/m³ a 12,4 Ug/m³ presentados en la figura 107, en periodos de cada 6 horas.

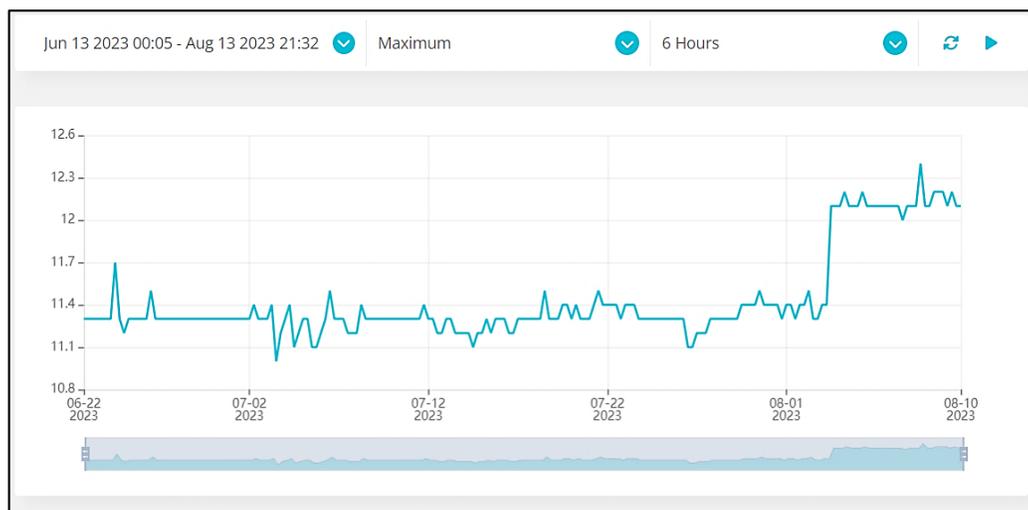


Figura 107. Valores máximos almacenados de O₃

Fuente: Elaborado por el autor

El valor de la media de 6 horas obtenido del ozono es de 10,63 Ug/m³, como se muestra en la figura 108.

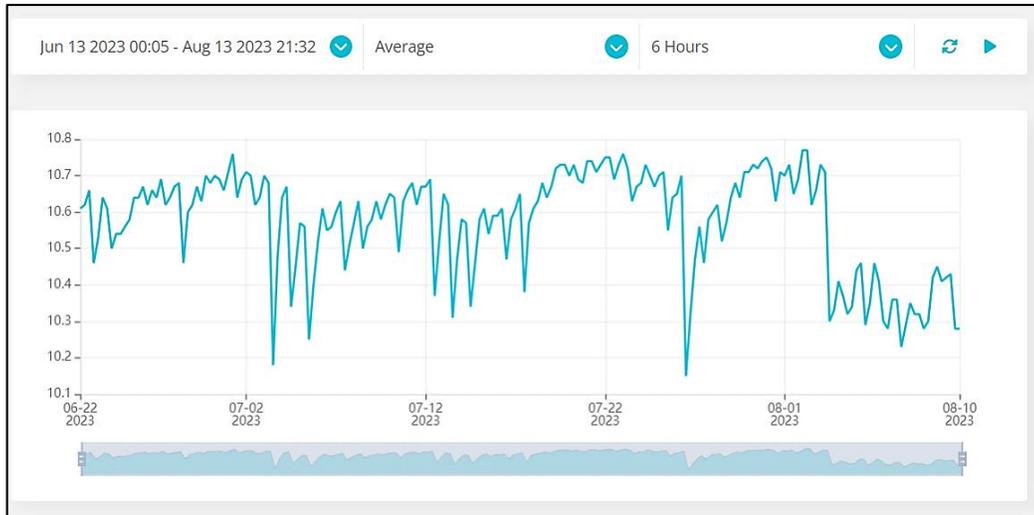


Figura 108. Valores promedios obtenidos de O3

Fuente: Elaborado por el autor

Dióxido de Nitrógeno (NO2)

Se presenta los valores recibidos del dióxido de nitrógeno enviados desde el nodo 2, ubicado en el terminal terrestre desde el 13 de junio al 13 de agosto. La figura 109, muestra los datos recibidos y la gráfica para dar un correcto entendimiento de la información.

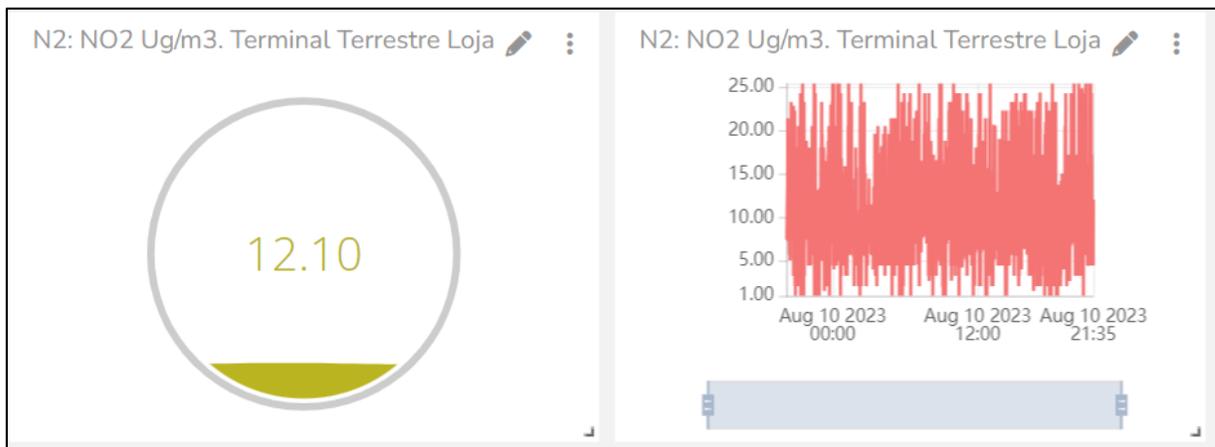


Figura 109. Widgets de visualización de los datos de NO2

Fuente: Elaborado por el autor

La figura 110, presenta los valores mínimos mostrados cada 24 horas, los rangos mínimos de NO2 van desde 1 Ug/m3 a 10,50 Ug/m3.

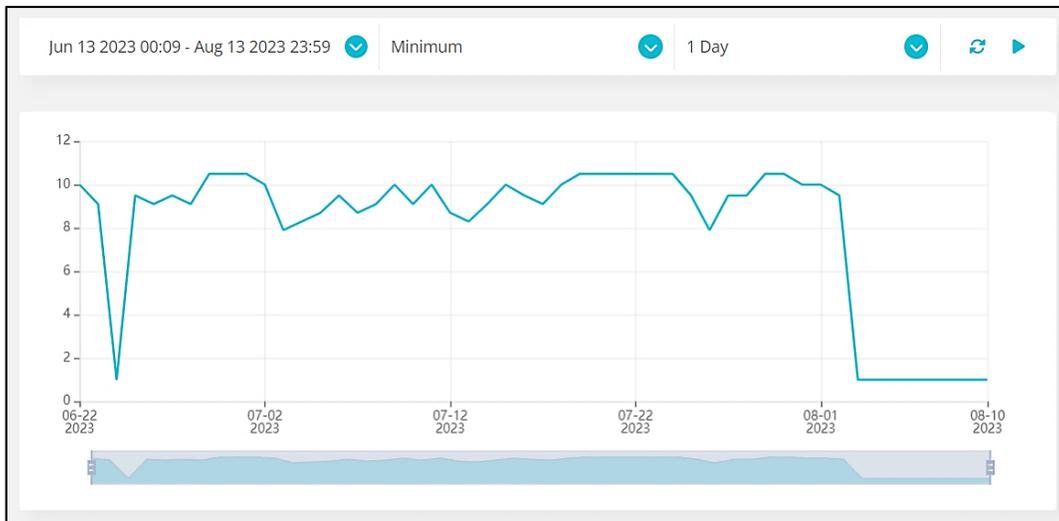


Figura 110. Valores mínimos obtenidos de NO2

Fuente: Elaborado por el autor

El rango de valores máximos de NO2 obtenidos del nodo 2 van desde 19,70 Ug/m3 a 25,50 Ug/m, como se aprecia en la figura 111.



Figura 111. Valores máximos obtenidos de NO2

Fuente: Elaborado por el autor

El valor en una media de 24 horas del NO2 es de 15,6 Ug/m3, como muestra la figura 112. Se presenta los valores promedios obtenidos del nodo 2.

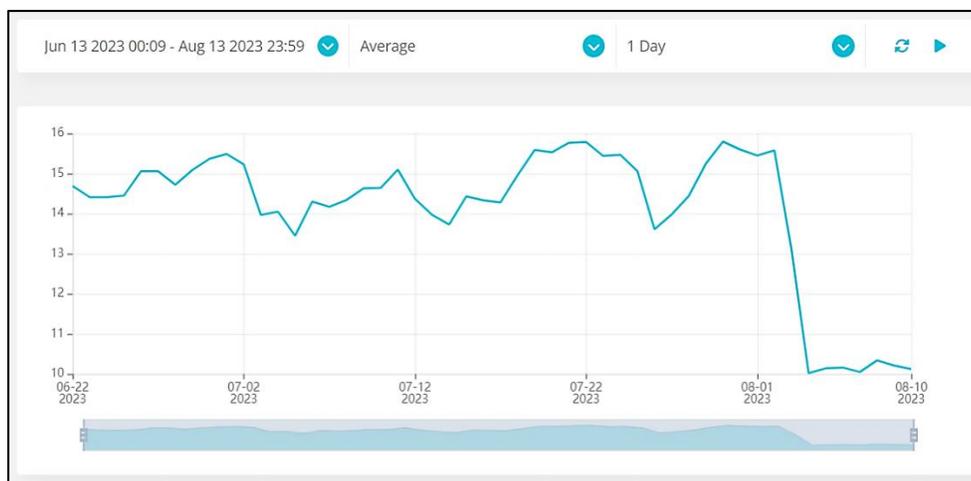


Figura 112. Promedio de los valores arrojados de NO2

Fuente: Elaborado por el autor

Dióxido de Azufre (SO2)

El siguiente gas medido es el dióxido de azufre, sus datos se obtuvieron del nodo 2 ubicado en el terminal terrestre del 13 de junio al 13 de agosto. En la parte inferior se muestra el dashboard figura 113, donde presenta los valores y el gráfico de línea donde se observa la información.

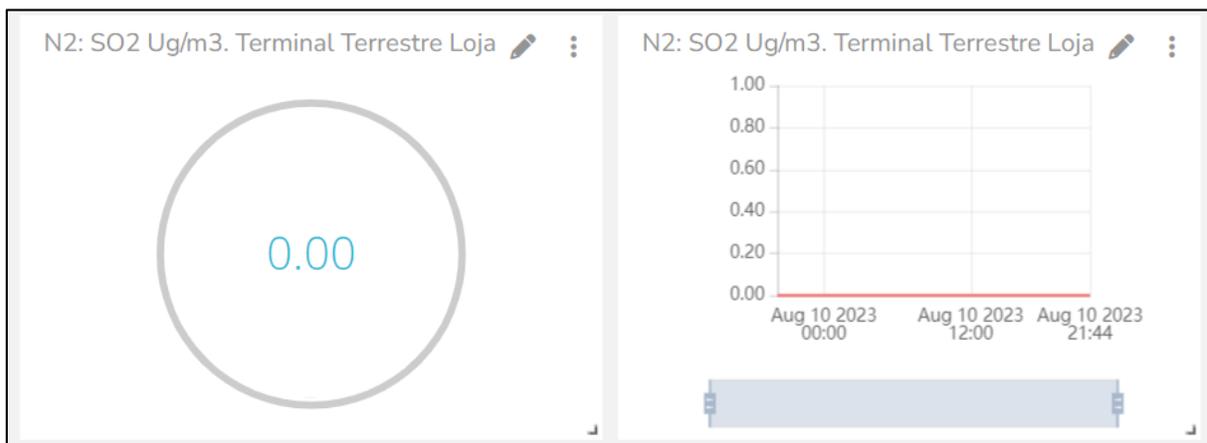


Figura 113. Widgets de visualización de los datos

Fuente: Elaborado por el autor

El rango mínimo del dióxido de azufre es de 0 Ug/m3 como se aprecia en la figura 114, en un lapso de tiempo de cada 24 horas. Estos datos son obtenidos por el nodo 2 ubicado en el terminal terrestre.

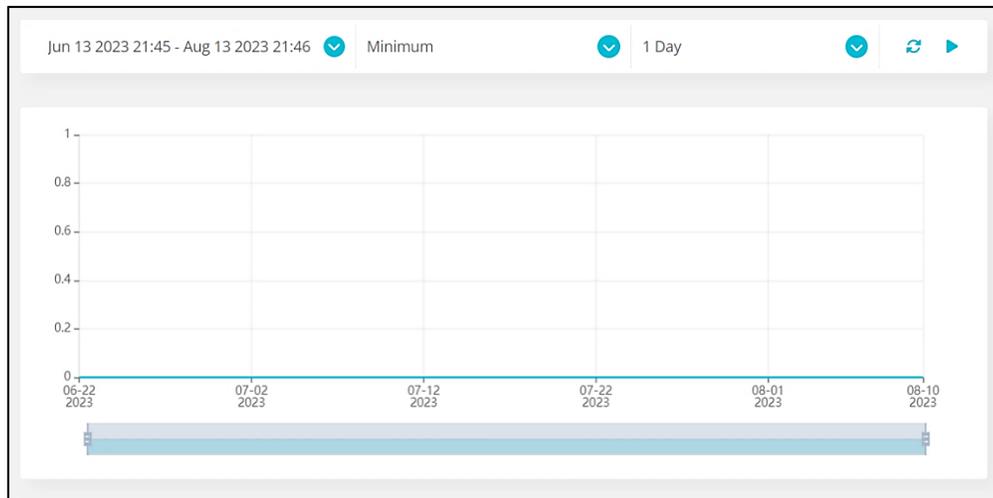


Figura 114. Valores mínimos obtenidos del SO2

Fuente: Elaborado por el autor

La figura 115, se observa el valor máximo de 23 Ug/m3 obtenido por el nodo 2 que se encuentra ubicado en el terminal terrestre.



Figura 115. Valores máximos obtenidos del SO2

Fuente: Elaborado por el autor

La media obtenida en 24 horas del SO2, arroja un valor de 0,1 Ug/m3, como se observa en la figura 116, presenta todos los valores que se encuentran almacenados en la plataforma.

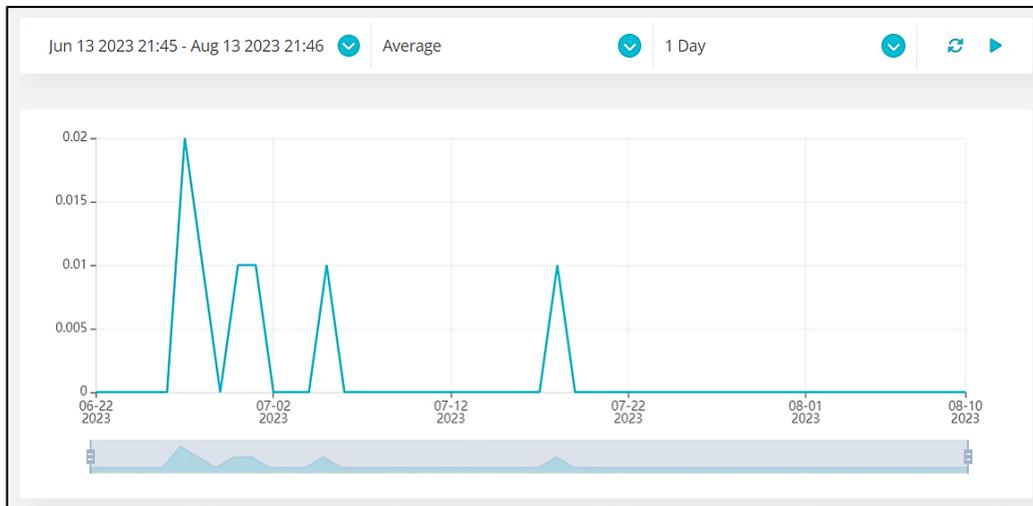


Figura 116. Valores promedios obtenidos de SO2

Fuente: Elaborado por el autor

Material Particulado (PM2,5)

Los valores del material particulado PM2,5 obtenidos del nodo sensor 2 ubicado en terminal terrestre del 13 de junio al 13 de agosto, se presentan en el dashboard figura 117.

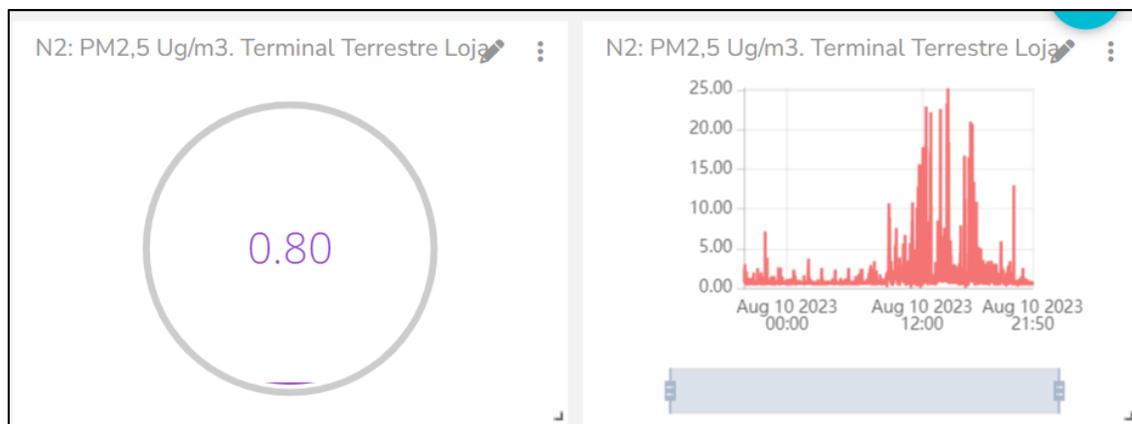


Figura 117. Widgets de visualización de los datos de PM2,5

Fuente: Elaborado por el autor

La figura 118, muestras los valores mínimos obtenidos del nodo 2 que se encuentra situado en el terminal terrestre, el rango de valores va de 0 Ug/m3 a 0,2 Ug/m3.

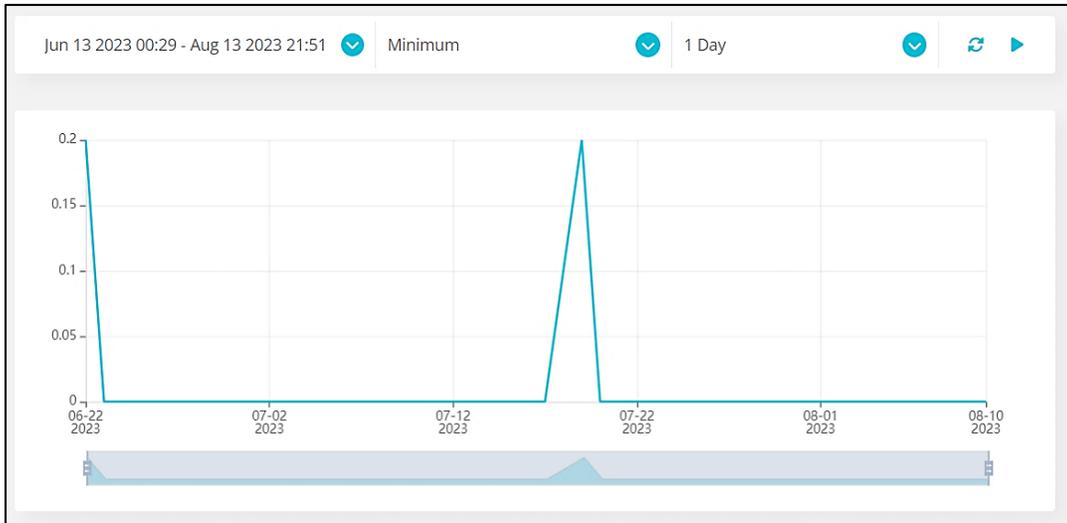


Figura 118. Valores mínimos obtenidos del PM2,5

Fuente: Elaborado por el autor

Los valores máximos arrojados del nodo 2 se observan en la figura 119, que van desde 24,10Ug/m3 a 25,5 Ug/m3, con periodos de tiempo de cada 24 horas.

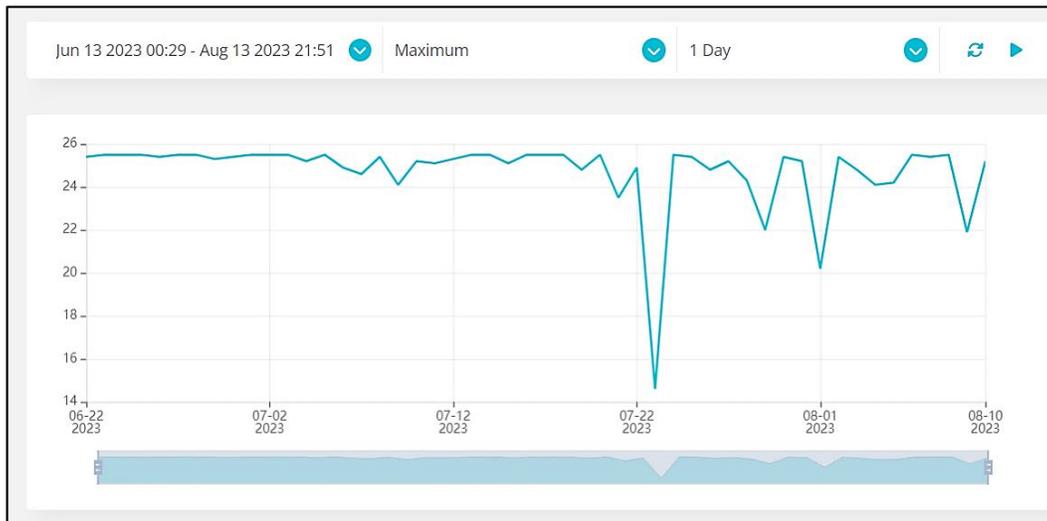


Figura 119. Valores máximos obtenidos del PM2,5

Fuente: Elaborado por el autor

La media de 24 horas arroja un valor de 3,89 Ug/m3, en la figura 120 se visualiza los datos obtenidos y almacenados en la plataforma de Ubidots

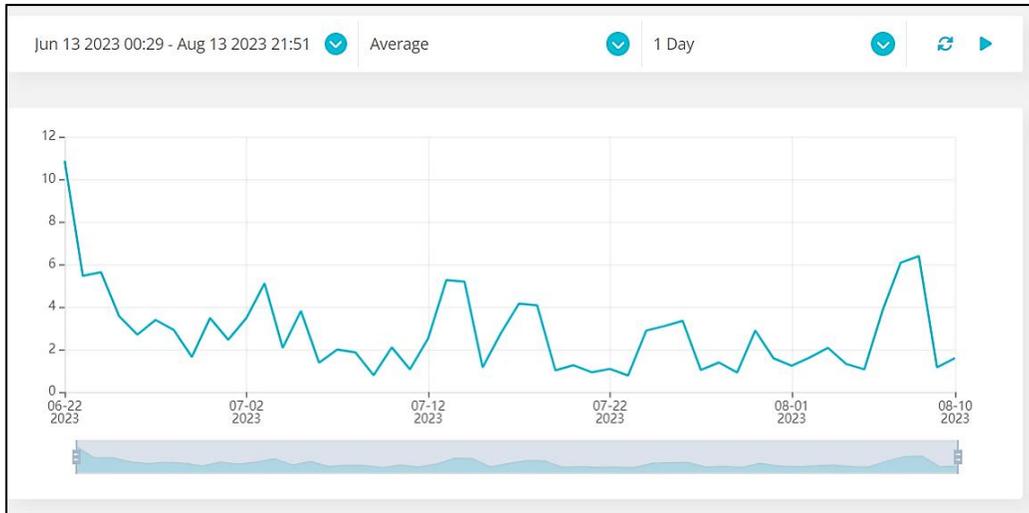


Figura 120. Valores promedio obtenidos del PM2,5

Fuente: Elaborado por el autor

Material Particulado (PM10)

El ultimo gas medido es el PM10 del nodo 2 ubicado en el terminal terrestre en los meses de junio a agosto, se obtuvo los siguientes valores presentados en la figura 121.

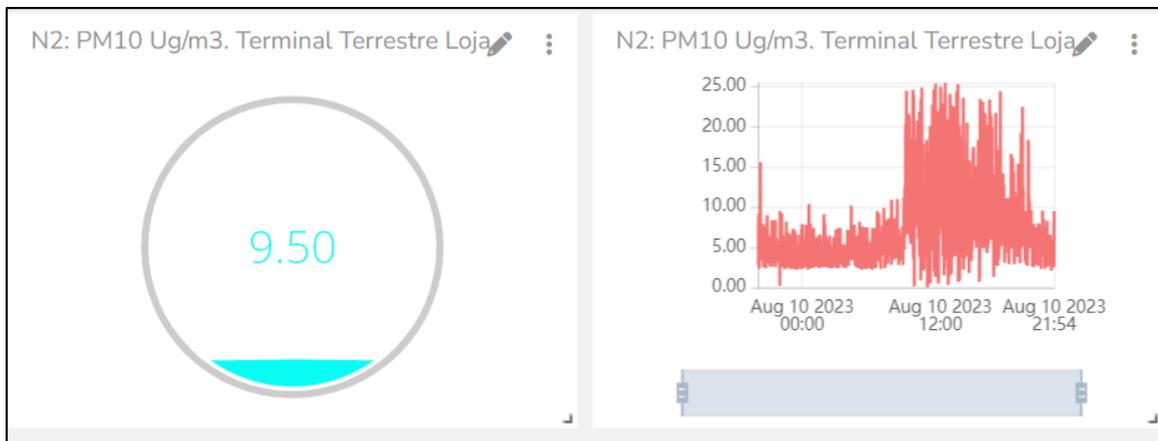


Figura 121. Widgets de visualización de los datos del PM10

Fuente: Elaborado por el autor

Los valores mínimos obtenidos por el nodo 2 van en el rango de 0 Ug/m3 a 0,6 Ug/m3 como se presenta en la figura 122. La gráfica presenta los valores en periodos de tiempo cada 24 horas.



Figura 122. Valores mínimos obtenidos del PM10

Fuente: Elaborado por el autor

En la figura 123, se visualiza los datos máximos obtenidos cada 24 horas, estos valores van de 25,30 Ug/m3 a 25,50 Ug/m3.



Figura 123. Valores máximos obtenidos del PM10

Fuente: Elaborado por el autor

El valor obtenido del PM10 es de 9,20 Ug/m3 en una media de 24 horas, como se observa en la figura 124.

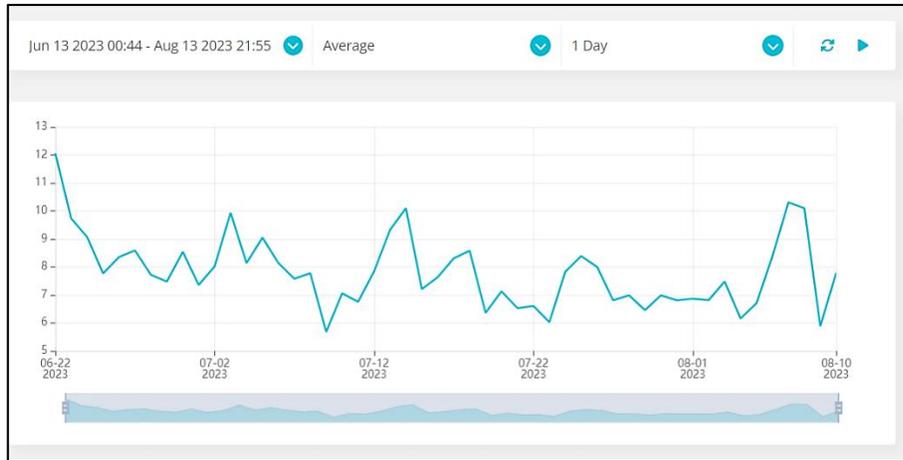


Figura 124. Valores promedios obtenidos del PM10

Fuente: *Elaborado por el autor*

La plataforma web Ubidots además de presentar y almacenar la información permite enviar notificaciones de alertas; a través de correo electrónico, SMS y Telegram. A continuación, se muestra las notificaciones recibidas de cada uno de los gases, con lo que respecta al Nodo 2 ubicado en el terminal terrestre.

Se presentan las notificaciones de alertas en Telegram, SMS y correo electrónico del dióxido de carbono (CO₂), figura 125.



Figura 125. Notificaciones de alertas del CO2; a) Telegram, b) SMS y c) Correo electrónico

Fuente: Elaborado por el autor

Como se aprecia en la figura 126, se presenta las notificaciones de las alertas obtenidas del ozono (O3), en Telegram, SMS y correo electrónico.

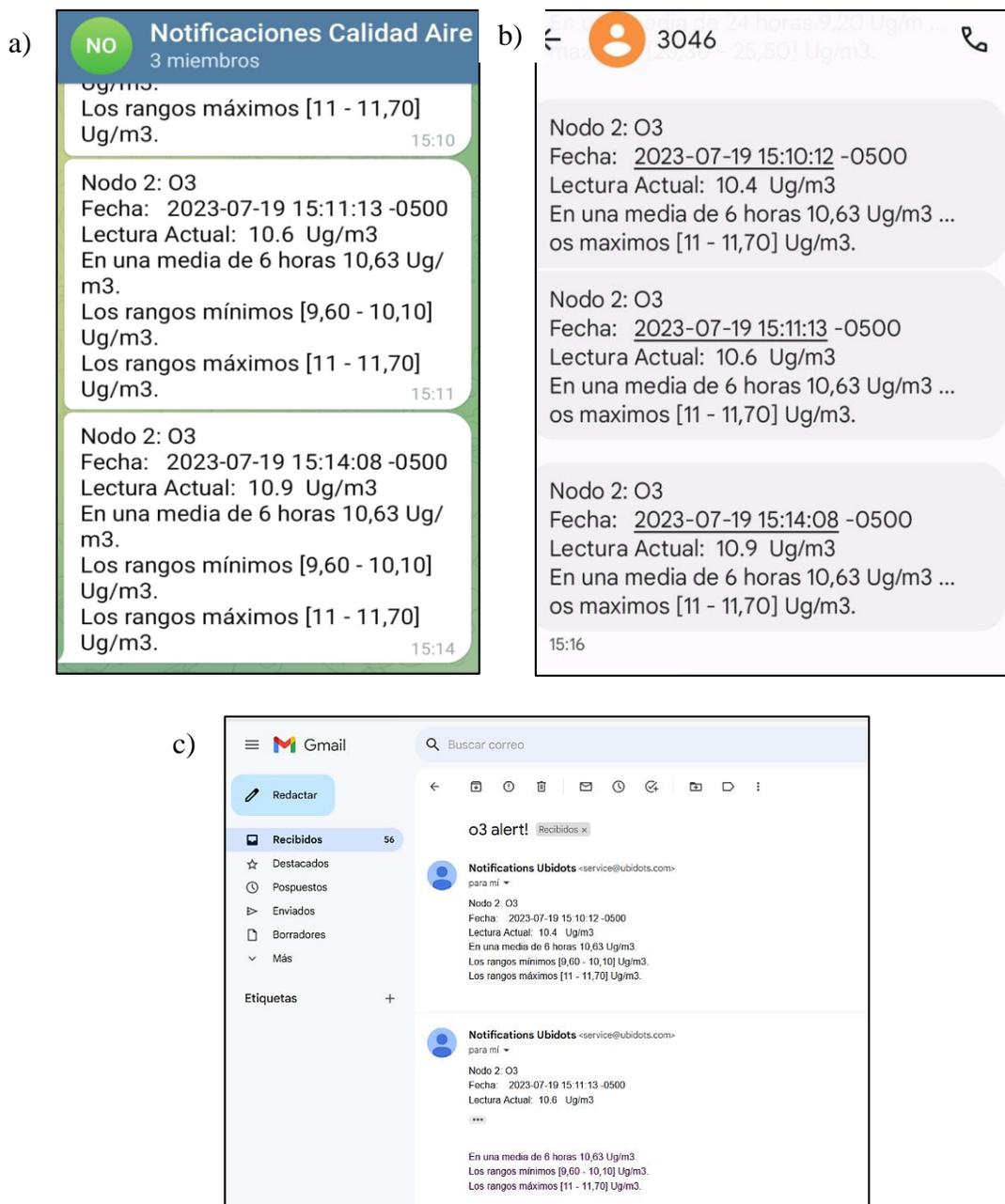


Figura 126. Notificaciones de alertas del O3; a) Telegram, b) SMS y c) Correo electrónico

Fuente: Elaborado por el autor

En la figura 127 se visualiza las notificaciones del dióxido de nitrógeno (NO₂), recibidas de las alertas creadas en Ubidots tanto para el correo electrónico como para SMS y Telegram.

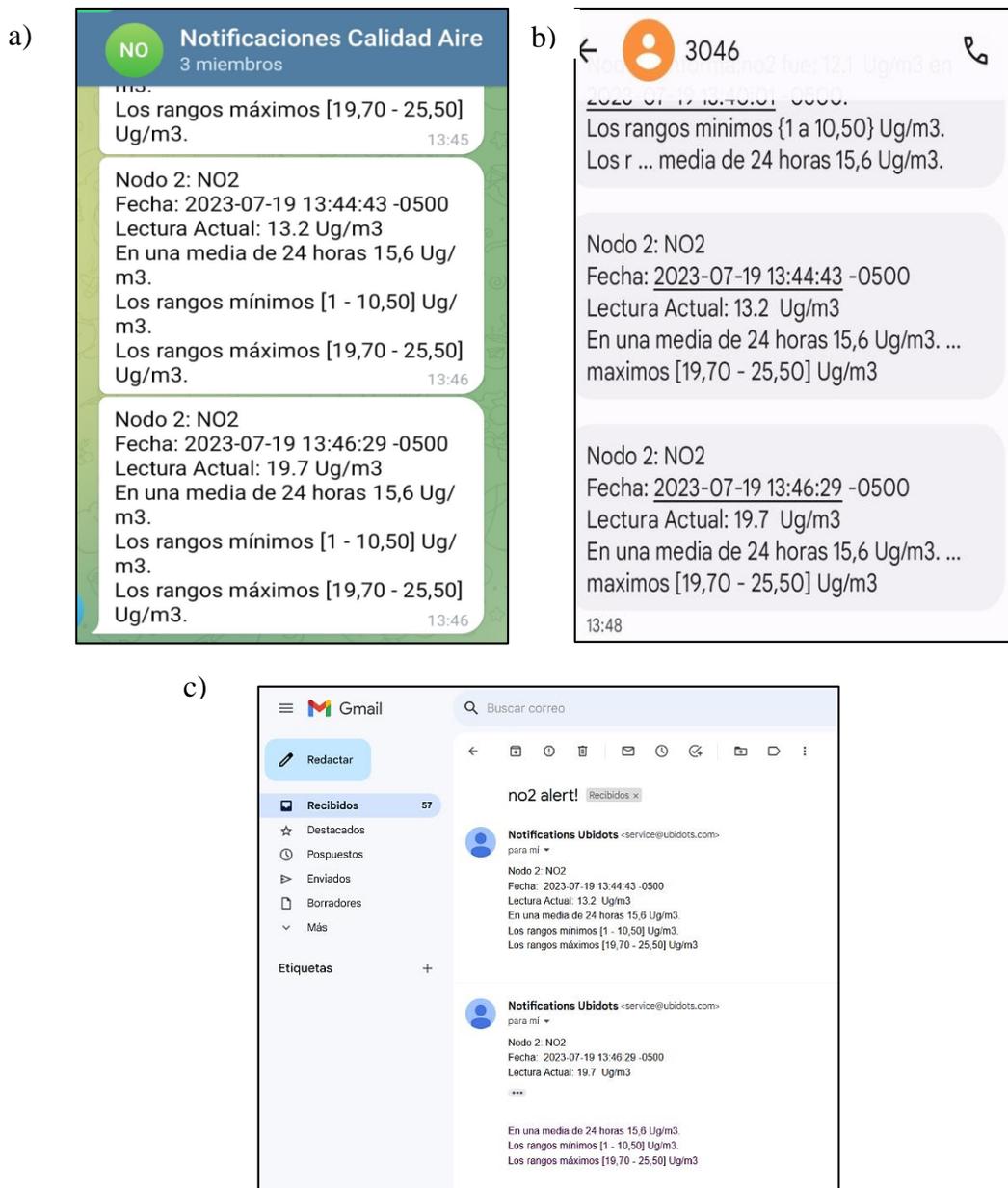


Figura 127. Notificaciones de alertas del gas NO₂, a) Telegram, b) SMS y c) Correo electrónico

Fuente: Elaborado por el autor

Se muestra las notificaciones de las alertas recibidas del Material particulado (PM_{2,5}), en Telegram, SMS y correo electrónico, figura 128.

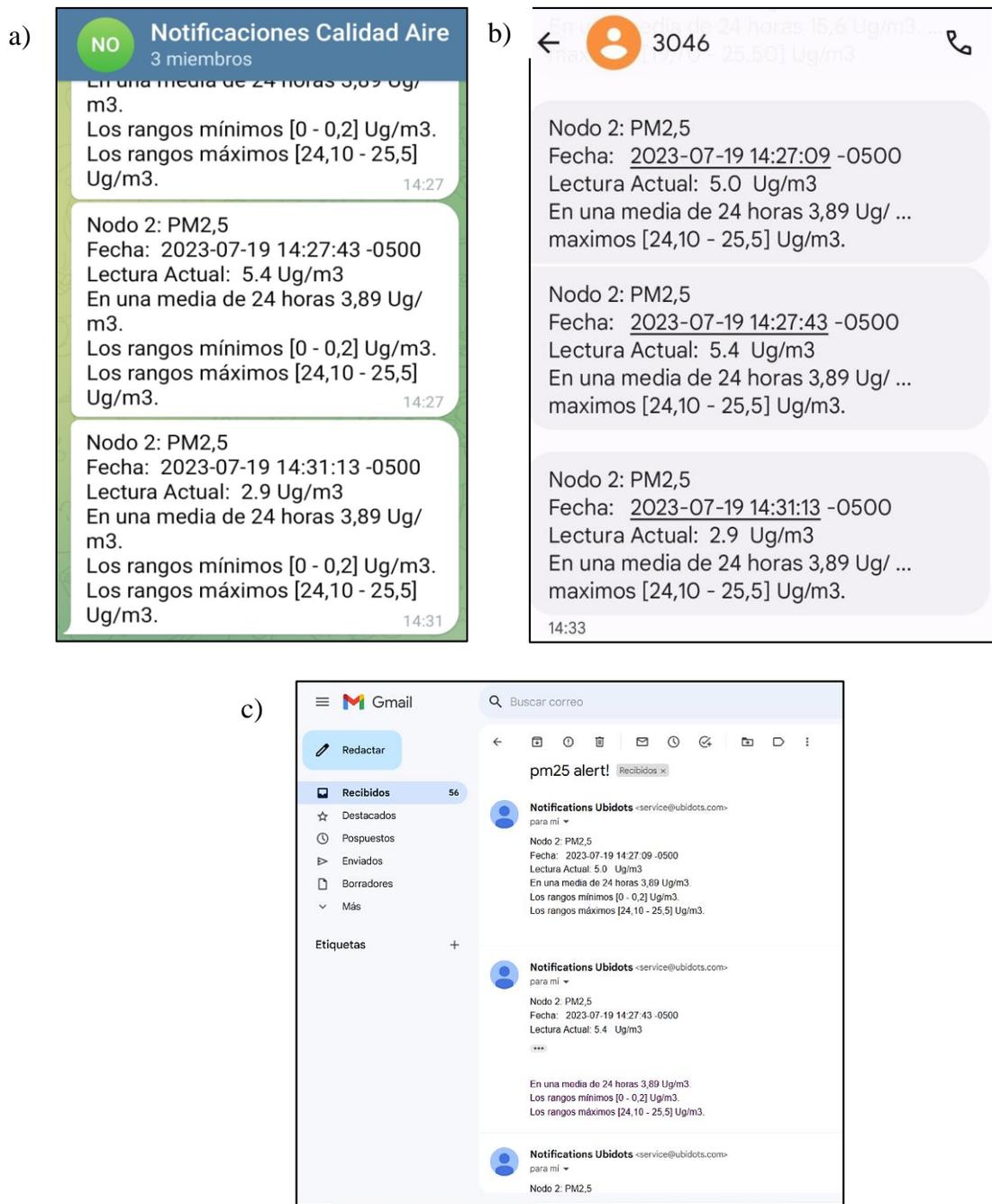


Figura 128. Notificaciones de alertas del PM2,5; a) Telegram, b) SMS y c) Correo electrónico

Fuente: Elaborado por el autor

En la figura 129, se puede observar las notificaciones de las alertas de material particulado (PM10) que son enviadas a Telegram, SMS y correo electrónico.

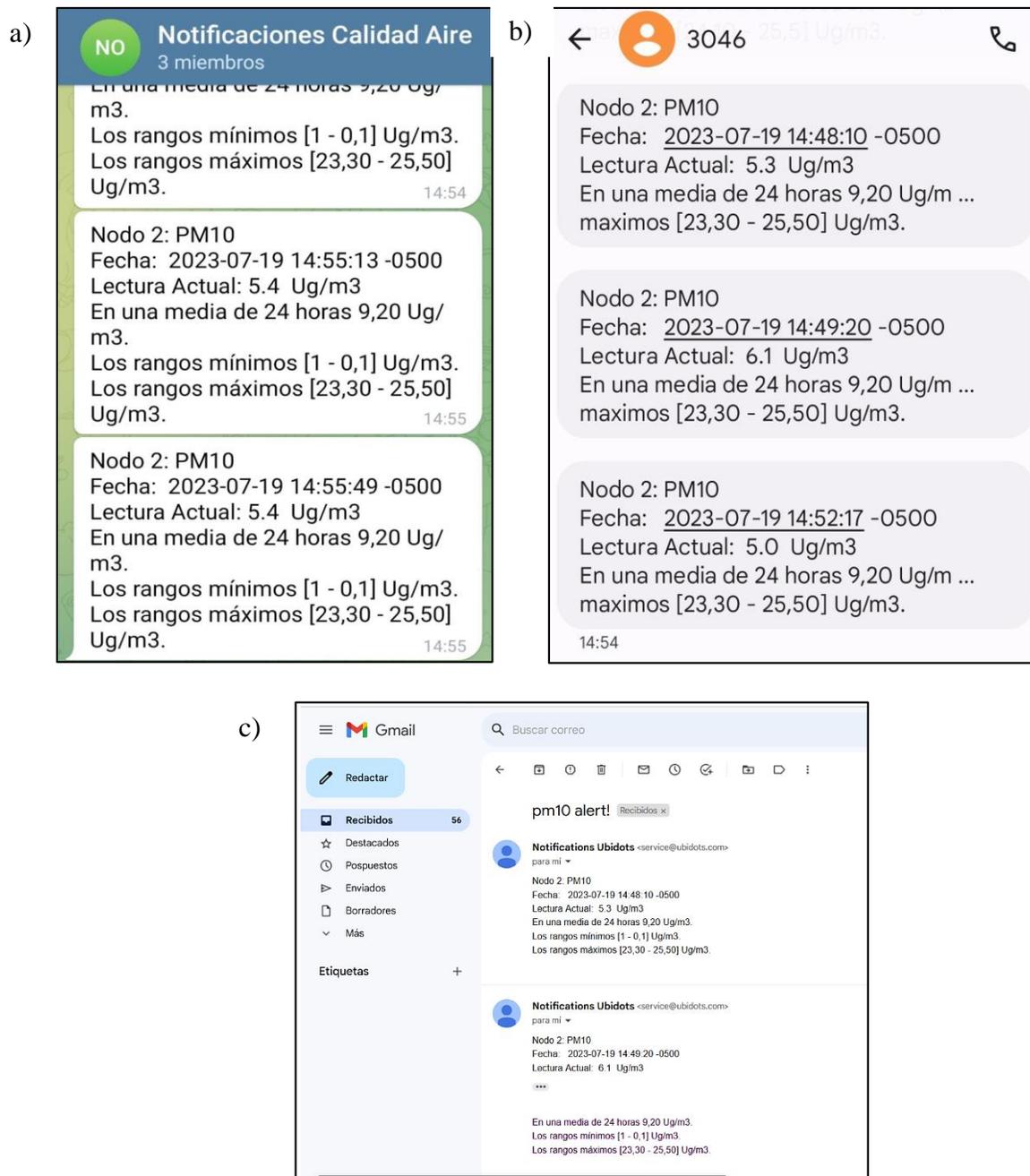


Figura 129. Notificaciones de alertas del PM10; a) Telegram, b) SMS y c) Correo electrónico

Fuente: Elaborado por el autor

Las notificaciones de alertas del dióxido de azufre (SO₂), del nodo 2, no han sido adjuntadas debido a que sus valores son de cero, rara vez presenta un valor y no ha permitido realizar estas activaciones.

Además de los beneficios que nos ofrece Ubidots al permitir visualizar y almacenar toda la información, se procedió a descargar todos los datos obtenidos del nodo sensor 1 y nodo sensor 2 y almacenarlos en Excel. Esto nos ha permitido realizar tablas y gráficos dinámicos para presentar los datos de una manera diferente, además de servir como una base de datos para visualizar, almacenar y respaldar la información.

En la figura 130 y 131 se presentan los valores de los gases almacenados en Excel. Estos datos nos proporcionarán la base para crear la tabla y gráfica dinámica, lo que facilitará la visualización de la información del nodo sensor 1 y a continuación el nodo sensor 2 de manera efectiva.

	A	B	C	D	E	F	G
1	Fecha	PM10	PM2,5	CO2	NO2	O3	SO2
2	17/7/2023 20:44	5,9	1	10,7	15,8	10,8	0,0
3	17/7/2023 20:43	7,8	0,7	10,7	18,1	11	0,0
4	17/7/2023 20:42	6,2	1,1	10,7	17,3	10,9	0,0
5	17/7/2023 20:42	8,6	0,6	10,7	10	10	0,0
6	17/7/2023 20:41	11,1	1,5	10,7	12,1	10,3	0,0
7	17/7/2023 20:41	2,9	1	10,7	11,5	10,2	0,0
8	17/7/2023 20:40	9,2	0,3	10,7	15,8	10,8	0,0
9	17/7/2023 20:40	5,2	3,8	10,7	15,8	10,8	0,0
10	17/7/2023 20:39	3,3	0,6	10,7	16,6	10,9	0,0
11	17/7/2023 20:38	5,4	0,3	10,7	17,3	10,9	0,0
12	17/7/2023 20:38	9,9	0,5	10,7	17,3	10,9	0,0
13	17/7/2023 20:37	3	1,4	10,7	10	10	0,0
14	17/7/2023 20:37	6,6	0,3	10,7	15,8	10,8	0,0
15	17/7/2023 20:36	5,6	0,6	10,7	12,6	10,4	0,0
16	17/7/2023 20:35	2,9	0,5	10,7	13,8	10,5	0,0
17	17/7/2023 20:35	5,3	0,2	10,7	10	10	0,0
18	17/7/2023 20:34	7,4	0,6	10,7	11	10,1	0,0
19	17/7/2023 20:34	2,6	1,2	10,7	10,5	10,1	0,0
20	17/7/2023 20:33	10,2	0,3	10,7	13,8	10,5	0,0
21	17/7/2023 20:33	3,4	0,7	10,7	19,7	11,2	0,0
22	17/7/2023 20:32	5	0,2	10,7	17,3	10,9	0,0
23	17/7/2023 20:31	5,1	0,6	10,7	15,8	10,8	0,0
24	17/7/2023 20:31	6,3	0,6	10,7	15,1	10,7	0,0
25	17/7/2023 20:30	6,3	0,6	10,7	18,1	11	0,0
26	17/7/2023 20:30	5,1	0,5	10,7	18,9	11,1	0,0
27	17/7/2023 20:29	2,7	0,5	10,7	18,9	11,1	0,0
28	17/7/2023 20:28	7,2	0,3	10,9	10	10	0,0
29	17/7/2023 20:28	5,4	0,5	10,7	15,1	10,7	0,0
30	17/7/2023 20:27	8,2	0,5	10,7	11,5	10,2	0,0
31	17/7/2023 20:27	7	0,8	10,7	15,1	10,7	0,0
32	17/7/2023 20:26	6,1	0,7	10,9	10	10	0,0

Figura 130. Datos almacenados en Excel de los gases obtenidos por el NODO 2

Autor: Elaborado por el autor

	Fecha	PM2,5	CO2	O3	PM10	NO2
2	1/8/2023 20:42	0,1	13,8	2,9	0,3	19,7
3	1/8/2023 20:32	0	13,7	2,9	2,4	12,6
4	1/8/2023 20:22	0	13,7	2,9	3,8	12,6
5	1/8/2023 20:11	0	13,7	2,9	7,2	13,2
6	1/8/2023 20:01	0,8	13,8	2,9	4	13,2
7	1/8/2023 19:51	2,1	13,8	3,1	0,6	20,6
8	1/8/2023 19:41	0	13,8	3	2	13,8
9	1/8/2023 19:31	0	13,8	3	0	14,5
10	1/8/2023 19:21	0	13,8	2,9	3,3	11
11	1/8/2023 19:11	0	13,8	2,9	0	11
12	1/8/2023 19:01	0,6	13,8	3	1	12,6
13	1/8/2023 18:51	0	13,8	3	2,5	16,6
14	1/8/2023 18:41	0	13,8	3	6,4	20,6
15	1/8/2023 18:31	0	14	3,1	0,3	16,6
16	1/8/2023 18:21	0,6	13,8	3	5,7	14,5
17	1/8/2023 18:10	0	13,8	2,9	3,7	20,6
18	1/8/2023 18:00	1,5	13,7	2,9	2,2	11,5
19	1/8/2023 17:50	1	13,8	2,9	9,8	12,1
20	1/8/2023 17:40	0,3	13,8	3	2,4	13,8
21	1/8/2023 17:30	2,6	14,5	3,2	9,1	16,6
22	1/8/2023 17:20	0	13,8	2,9	1,1	12,1
23	1/8/2023 17:15	4,3	13,8	2,9	3,5	17,3
24	1/8/2023 17:05	0	13,8	3	0,9	17,3
25	1/8/2023 16:55	1,2	14	3,2	6,6	16,6
26	1/8/2023 16:45	0	14	3,1	1,7	20,6
27	1/8/2023 16:35	0,4	14	3,2	7,7	17,3
28	1/8/2023 16:24	0	14,1	3,2	2,1	13,8

Figura 131. Datos almacenados en Excel de los gases obtenidos por el NODO 1

Autor: Elaborado por el autor

En la figura 132 y 133 se pueden observar la tabla y gráfica dinámica creadas en Excel, las cuales nos permiten obtener de forma rápida y eficiente valores promedios, máximos, mínimos, sumatorias y otros datos relevantes. Cabe señalar que se pretendió realizar comparaciones entre los valores de los diferentes gases para encontrar patrones de cambio, pero no se evidenció relación alguna entre gases o los cambios que sufren los gases respecto al resto.

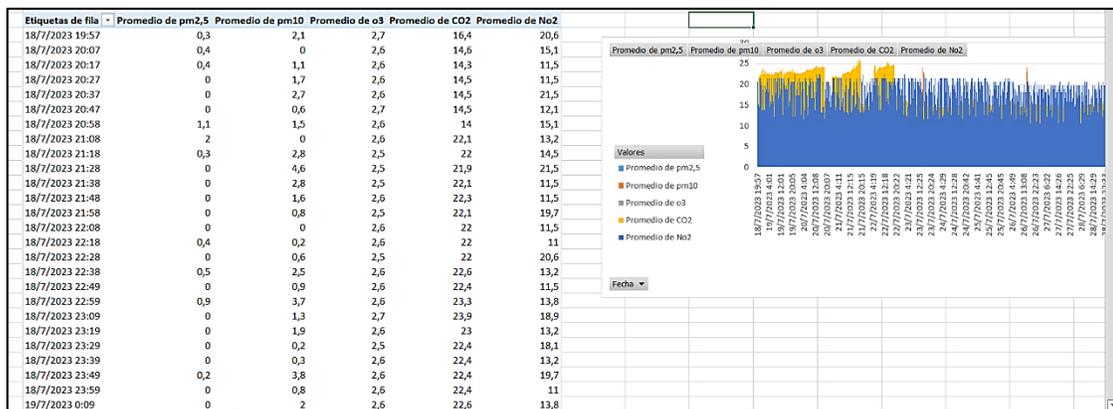


Figura 132. Visualización de los datos en tabla y gráfica dinámica en Excel

Autor: Elaborado por el autor

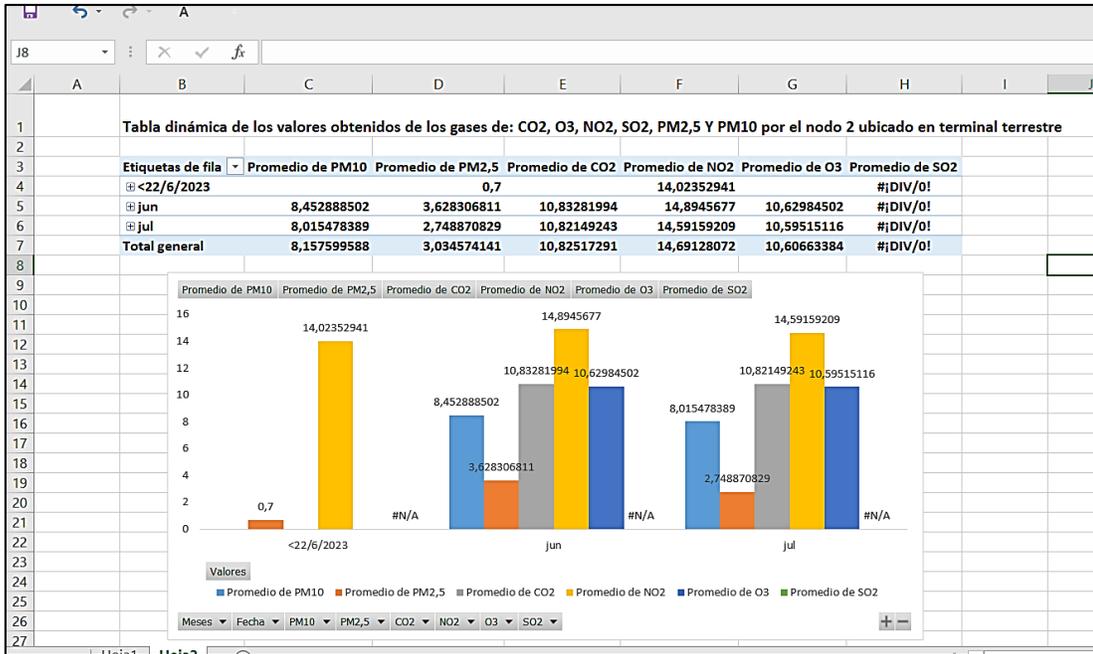


Figura 133. Visualización de los datos en tabla y gráfica dinámica en Excel

Autor: Elaborado por el autor

7. Discusión

Con el objetivo de reconocer los gases a medir para los fines de este estudio, se ha realizado un estudio bibliográfico pertinente que ha permitido identificar cuáles son los tipos de gases que más contaminan el aire y afectan a la salud de las personas. Una vez obtenida esta información se procedió al análisis minucioso de cada gas identificado con el fin de conocer los valores máximos y mínimos permitidos dentro de los estándares de la OMS y la Normativa Ecuatoriana de Calidad de Aire, y a su vez conocer los efectos en la salud al estar expuestos en prolongados tiempos. Así entonces los gases que se miden para este estudio son: dióxido de carbono, ozono, dióxido de nitrógeno y dióxido de azufre y el material particulado PM_{2,5} y PM₁₀.

Para el desarrollo del segundo objetivo se realizó una investigación y comparación de las tecnologías LPWAN, donde se evidenció que la tecnología LoRa presenta las características adecuadas para el desarrollo del proyecto; siendo éstas: velocidad de transmisión, distancia de transmisión, ancho de banda, disponibilidad, costo, calidad de servicio, alta tolerancia a interferencia y las frecuencias utilizadas están aprobadas por la Agencia de Regulación y control de Telecomunicaciones (ARCOTEL), que nos facilita una eficiente comunicación entre el emisor (módulos) y el receptor (gateway).

Para dar cumplimiento al tercer objetivo, se diseñó e implementó la red de LoRaWAN para el monitoreo de los gases ambientales, la cual está conformada por: tres sensores de gases (CO₂, NO₂, SO₂, O₃) y un cuarto para la medición del material particulado (PM_{2,5} y PM₁₀); esta información es leída a través de la comunicación analógica (gases) y digital (PM) por el Arduino UNO. Utilizando la comunicación I2C se conecta el arduino UNO al módulo heltec cubecell, para el traspaso de la información, con la finalidad de poder realizar un enlace inalámbrico mediante el protocolo LoRaWAN, permitiendo así transmitir los datos al Gateway; el cual se encargará de almacenar la información y enviar a la nube (TTN).

Para procesar la información que es recibida a través de la nube se utilizó la plataforma Ubidots donde nos permite visualizar los datos de los gases de CO₂, O₃, NO₂, SO₂ y material particulado PM_{2,5}/ PM₁₀ en Ug/m³ cada cierto periodo de tiempos: 5 minutos, 30 minutos, 1 hora, 6 horas y cada un día, dependiendo del sensor a leer. Los datos se los puede visualizar como mediciones, mínimas, máximas y promedios; además nos permite realizar alerta de aviso

cuando se excede los límites máximos permitidos; las notificaciones son enviadas mediante correo electrónico, Telegram y SMS, entre otras; la actualización de los datos es constante.

La plataforma web Ubidots nos sirve como una base de datos, donde nos permite recolectar y almacenar gran cantidad de información, la cual puede ser analizada mediante diagramas, gráficos de lineales, widgets, etc. A su vez se puede realizar la descarga de toda la información para ser almacenada en otras herramientas informáticas alternas como Excel.

Una vez obtenidos los resultados de los niveles de emisión de los gases en el ambiente durante los meses de junio-agosto, se procede a realizar una comparación con los valores máximos permisibles tanto de la Organización Mundial de la Salud y la Normativa Ecuatoriana de Calidad de Aire.

Dentro de las mediciones realizadas en los nodos 1 (ubicado en el mercado mayorista) y el nodo 2 (ubicado en el terminal terrestre) se obtuvieron los siguientes resultados:

El primer gas medido es el dióxido de carbono (CO_2), sus valores presentados en una media de 30 minutos tanto para el nodo 1 (21,36 Ug/m³) y el nodo 2 (10,83 Ug/m³), estando por debajo de los límites permisibles de la OMS (30000 Ug/m³) y la NECA (30000 Ug/m³) establecidos en una media de 30 minutos.

El segundo gas medido fue el ozono (O_3), sus valores presentados en una media de 8 horas tanto para el nodo 1 (2,88 Ug/m³) y el nodo 2 (10,63 Ug/m³), encontrándose por debajo de los límites permitidos de la OMS (100 Ug/m³) y la NECA (100 Ug/m³) establecidos en una media de 8 horas.

El tercer gas medido es el dióxido de nitrógeno (NO_2), sus valores presentados en una media de 24 horas tanto para el nodo 1 (15,95 Ug/m³) y el nodo 2 (15,6 Ug/m³), encontrándose situados por debajo de los límites permitidos de la OMS (100 Ug/m³) y la NECA (125 Ug/m³) establecidos en una media de 24 horas. El núcleo urbano de Loja la mayoría de vehículos que circulan por ahí son livianos y por la carencia de industrias hace posible que se mantengan los niveles de NO_2 bajos, dando como resultado la calidad del aire sea relativamente buena respecto a los límites establecidos por las normativas ambientales y de salud.

El cuarto gas medido tenemos el dióxido de azufre (SO_2), sus valores presentados en una media de 24 horas tanto para el nodo 1 (0,00 Ug/m³) y el nodo 2 (0,1 Ug/m³), encontrándose situados

por debajo de los límites permitidos de la OMS (20 Ug/m³) y la NECA (40 Ug/m³) establecidos en una media de 24 horas. Sus valores son bajos debido que en Loja existe más circulación de vehículos livianos y escasa presencia de industrias, por lo que estos niveles no son elevados. Se debe tratar de mantener o disminuir estos valores, ya que; si estos niveles en un futuro aumentan provocarían daños en la salud.

El quinto y sexto valor medido fue de material particulado dividiéndose en PM_{2,5} y PM₁₀, obteniendo en PM_{2,5} sus valores presentados en una media de 24 horas para el nodo 1 (0,33 Ug/m³) y el nodo 2 (3,89 Ug/m³), estos valores se encuentran por debajo de los límites permitidos de la OMS (25 Ug/m³) y la NECA (50 Ug/m³) establecidos en una media de 24 horas. Para el PM₁₀ sus valores presentados en una media de 24 horas para el nodo 1 (1,67) y el nodo 2 (9,20 Ug/m³), estos valores se encuentran por debajo de los límites permitidos de la OMS (50 Ug/m³) y la NECA (100 Ug/m³) establecidos en una media de 24 horas.

8. Conclusiones

Las conclusiones del presente Trabajo de Integración Curricular son las siguientes:

- Se implementó una red de sensores inalámbricos como una herramienta fundamental para evaluar y comprender la contaminación del aire en la ciudad de Loja. Proporcionando datos precisos, su accesibilidad y su enfoque sostenible la convierten en una solución tecnológica valiosa para el monitoreo ambiental.
- La revisión bibliográfica permitió conocer cuáles son los gases que más contaminan el aire y afectan a la salud humana.
- Se usó la tecnología LoRa ya que ofrece una solución eficiente y confiable para el envío de datos de manera inalámbrica, presentando características como comunicación de largo alcance, bajo consumo de energía, alta escalabilidad, bajo costo y frecuencias de operación permitidas.
- Las técnicas de modulación de espectro para transmitir datos, confiere a LoRa una mayor inmunidad a las interferencias y la capacidad de funcionar en entornos con ruido electromagnético. Por esta razón la red implementada demostró funcionar de manera adecuada con respecto a la propuesta planteada.
- Se diseñó una red de sensores basada en el protocolo LoraWAN, la cual permite la monitorización en tiempo real de diversas variables y condiciones del entorno, proporcionando datos actualizados que son valiosos para tomar decisiones informadas y rápidas en caso de detectarse niveles de gases y material particulado preocupantes. Esto facilita la identificación de fuentes de contaminación y la adopción de medidas correctivas antes de que se agraven los problemas ambientales o de salud.
- Al recopilar datos de manera continua y sistemática, la red de sensores inalámbrica permite el seguimiento de tendencias y patrones de calidad del aire a lo largo del tiempo. Esto es útil para identificar cambios estacionales, variaciones diarias y tendencias a largo plazo, lo que proporciona información valiosa para políticas ambientales y de salud pública.
- Los valores de medición obtenidos de los gases de dióxido de carbono, ozono, dióxido de nitrógeno, dióxido de azufre y el material particulado PM_{2,5} y PM₁₀, ubicados en el terminal terrestre y el mercado mayorista registran promedios de concentraciones que se

encuentran dentro de los límites máximos permisibles establecidos por la Organización Mundial de la Salud y la Normativa ecuatoriana de Calidad del Aire.

- Se implementó una interfaz web a través de la herramienta Ubidots, la cual permite tener acceso permanente a la información tomada de los sensores. A su vez, la misma facilitó la configuración, gestión de dispositivos; proporcionando acceso a datos en tiempo real, lo que permite a los usuarios monitorear y analizar la información de los dispositivos conectados de manera inmediata; lo cual es fundamental para tomar decisiones oportunas y reaccionar rápidamente a cambios o eventos inesperados.
- Se generó una base de datos con la herramienta Ubidots y su posterior almacenamiento en formato Excel, la cual permite acceder y proporcionar información valiosa para su análisis.
- El proyecto actual considera la lectura y envío de información de 6 sensores, pero la nube de TTN restringe solamente la recepción de 4 de ellos, por lo cual se tuvo que elaborar rutinas que permitan posterior al envío de la información de los 4 primeros, enviar la lectura de los dos restantes.
- Se trató de evaluar cada uno de los gases y sus cambios, frente a los otros gases, donde no se encontró un patrón de variación uniforme para su respectiva comparación.

9. Recomendaciones

Desarrollado el Trabajo de Integración Curricular, se establecen las siguientes recomendaciones:

- Para poder adquirir los diferentes dispositivos que conforman los prototipos de medición, se debe investigar las frecuencias permitidas en el Ecuador, para evitar inconvenientes de interferencia y en el envío de datos.
- Al utilizar prototipos medidores de gases contaminantes, se recomienda realizar la calibración y actualización de los datos necesarios acordes a los reglamentos en nuestro país o la normativa internacional.
- Se evaluar la compatibilidad técnica de los nodos y el Gateway para la comunicación efectiva entre los mismos, caso contrario no se establece la interconexión entre ambos.
- La tecnología LoRa nos permite obtener buenos resultados en el envío de datos, pero se debe tener muy presente que debe existir línea de vista.
- Si el prototipo va a medir varios gases adicionales, se debe considerar las conexiones electrónicas, como puertos disponibles, voltaje, corriente y compatibilidad, por lo cual será necesario evaluar cuál es el equipo que atendería esta necesidad.
- Debido a que la nube TTN solo recepta la información de 4 gases, cuando se quiera enviar más de 4 lecturas de gases es necesario la elaboración de rutinas correspondientes.
- La conexión de la nube con la plataforma web requiere seguir una serie de pasos que deben cumplirse de manera ordenada y precisa. Esto se debe a que existen códigos y claves únicas involucradas en el proceso. Además, dentro de estos códigos, es necesario identificar las líneas que deben eliminarse y comprender cuáles de ellas son útiles.
- Para obtener datos actuales y constantes se recomienda que los periodos de medición sean cortos o de acuerdo a la necesidad.
- Ubidots permite una amplia gama de funciones que incluyen visualización, almacenamiento, alertas, base de datos y actualización constante de los datos, lo que resulta sumamente útil para desarrollar proyectos de IoT que requieran de estas funcionalidades.
- La generación de una base de datos permite tomar medidas de gestión y control.

- Se recomienda ubicar mayor número de nodos sensores en los diferentes lugares de la ciudad, para conocer con mayor exactitud la calidad del aire.
- Es importante que el Municipio de Loja y el Ministerio de Ambiente, implementen prototipos inteligentes para el monitoreo de los gases, a su vez, informen a la ciudadanía los efectos que producen en la salud.
- Se recomienda a trabajos futuros presentar un análisis dedicado, para comprobar si existen patrones entre un gas respecto a los otros gases.

10. Bibliografía

- Akpakwu, G. A., Silva, B. J., Hancke, G. P., & Abu-Mahfouz, A. M. (2017). A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges. *IEEE Access*, 6, 3619–3647. <https://doi.org/10.1109/ACCESS.2017.2779844>
- Ballester, F., José María, T., & Pérez-Hoyos, S. (2009). Efectos de la contaminación atmosférica sobre la salud: Una introducción. In *Revista española de salud pública* (Vol. 73, Issue 2, pp. 109–121). https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1135-57271999000200002%0Ahttp://scielo.isciii.es/pdf/resp/v73n2/efectos_contam.pdf
- Bankov, D., Khorov, E., & Lyakhov, A. (2017). *On the Limits of LoRaWAN Channel Access*. November, 10–14. <https://doi.org/10.1109/ent.2016.011>
- Belmonte, M. (2019). *Universidad politécnica de cartagena*. 0–209.
- Boffetta, P., Merler, E., & Vainio, H. (1993). Carcinogenicity of mercury and mercury compounds. *Scandinavian Journal of Work, Environment and Health*, 19(1), 1–7. <https://doi.org/10.5271/sjweh.1510>
- Borja, A. G. (2021). *Implementación de una red LPWAN para el control de consumo de agua potable utilizando tecnología LoRa*. 1–95. <http://repositorio.espe.edu.ec/bitstream/21000/25837/1/T-ESPE-044763.pdf>
- Candia, A., & Varela, L. N. (2017). *WSN para servicios públicos metropolitanos (Tesis de pregrado)*. 166. http://sedici.unlp.edu.ar/bitstream/handle/10915/75748/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- Casals, L., Mir, B., Vidal, R., & Gomez, C. (2017). Modeling the energy performance of LoRaWAN. *Sensors (Switzerland)*, 17(10). <https://doi.org/10.3390/s17102364>
- Centro Nacional de Prevención de Desastres. (2019). *El ozono como contaminante del aire y riesgo para la salud | Centro Nacional de Prevención de Desastres | Gobierno | gob.mx*. <https://www.gob.mx/cenapred/articulos/el-ozono-como-contaminante-del-aire-y-riesgo-para-la-salud>

- Chaudhari, B. S., Zennaro, M., & Borkar, S. (2020). LPWAN technologies: Emerging application characteristics, requirements, and design considerations. *Future Internet*, 12(3). <https://doi.org/10.3390/fi12030046>
- EPA. (2022). *Efectos del material particulado (PM) sobre la salud y el medioambiente*. Agencia de Protección Ambiental de Estados Unidos. <https://espanol.epa.gov/espanol/efectos-del-material-particulado-pm-sobre-la-salud-y-el-medioambiente>
- GEO-Loja. (2007). *Perspectivas del Medio Ambiente Urbano: GEO Loja*. [http://www.naturalezaycultura.org/docs/Geo Loja.pdf](http://www.naturalezaycultura.org/docs/Geo%20Loja.pdf)
- Heredia, A., & Lucero, P. (2021). *Diseño e implementación de una red inalámbrica de sensores con tecnología LoRa para monitoreo industrial orientado a OPC de arquitectura unificada*. [http://dspace.ucuenca.edu.ec/bitstream/123456789/35875/1/Trabajo de Titulación.pdf](http://dspace.ucuenca.edu.ec/bitstream/123456789/35875/1/Trabajo%20de%20Titulaci%C3%B3n.pdf)
- Instituto Nacional de Seguridad y Salud en el Trabajo. (2018). Dióxido de nitrógeno. In *Documentación Límites Exposición Profesional* (pp. 1–11). www.insst.es/dlep-documentacion-toxicologica
- MAATE. (2011). Texto Unificado de Legislación Secundaria del Ministerio del Ambiente. *Libro VI, Anexo 4, 10*. <http://extwprlegs1.fao.org/docs/pdf/ecu112183.pdf> http://www.quitoambiente.gob.ec/ambiente/images/Secretaria_Ambiente/red_monitoreo/informacion/norma_ecuato_calidad.pdf
- Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, 5(1), 1–7. <https://doi.org/10.1016/J.ICTE.2017.12.005>
- Moya, E. J. G. (2016). *Diseño De Una Red De Sensores Inalámbricos (Wsn) Para Monitorear Parámetros Relacionados Con La Agricultura*. 2–215. <https://bibdigital.epn.edu.ec/handle/15000/16809>
- OMS. (2021). *Contaminación del aire ambiente (exterior)*. <https://www.who.int/es/news->

room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health

- OYARZÚN G, M. (2010). Contaminación aérea y sus efectos en la salud. In *Revista chilena de enfermedades respiratorias* (Vol. 26, Issue 1). <https://doi.org/10.4067/s0717-73482010000100004>
- Porrás Calderón, W. A., & Salah García, O. Y. (2019). Implementación de una Red LPWAN de Sensores para el Monitoreo de Condiciones Ambientales. *Met&Flu*, 48–67. https://www.cdtdegas.com/images/Descargas/Nuestra_revista/MetFlu14/9_RedLPWAN.pdf
- S, P. P. O. R., & May, P. E. L. (2020). Índice de calidad del aire: cómo medir la calidad del aire residencial | S&P. In *2019* (pp. 1–6). <https://www.solerpalau.com/es-es/blog/indice-calidad-aire/>
- Santamaría Buitrago, F. A., & Archila Córdoba, D. M. (2013). Estado Del Arte De Las Redes De Sensores Inalámbricos. *Tecnología Investigación y Academia*, 1(2), 2344–8288.
- Secretaría del ambiente. (2019). *Norma Ecuatoriana de la Calidad del Aire*. <http://www.quitoambiente.gob.ec/index.php/informes#informe-calidad-del-aire-2019>
- Segovia, M., & Vidal, L. (2014). *Eguridad en redes de sensores inalámbricos*. <https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/19019/1/2078.pdf>
- Sharon Shea. (2017, September). *What is LPWAN (low-power wide area network)? - Definition from WhatIs.com*. <https://www.techtarget.com/iotagenda/definition/LPWAN-low-power-wide-area-network>
- Verde, E. (2019). *El impacto medioambiental del dióxido de carbono - conócelo aquí*. <https://www.ecologiaverde.com/el-impacto-medioambiental-del-dioxido-de-carbono-1334.html>

11. Anexos

Anexo 1. Conexión de los Sensores con el Arduino Uno

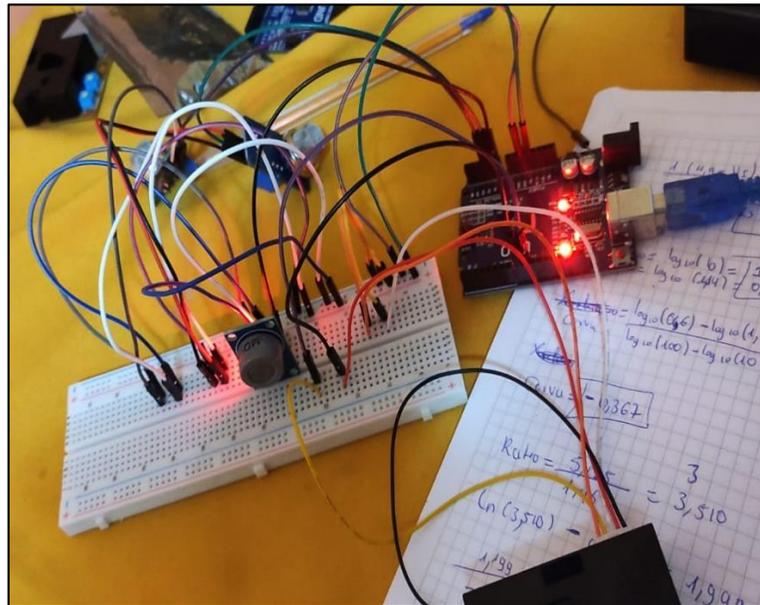


Figura 134. Conexión de los dispositivos

Fuente: Elaborado por el autor, 2023

```
#include <Wire.h>
#include <DSM501.h>

#define DSM501_PM10 9
#define DSM501_PM25 8

DSM501 dsm501(DSM501_PM10, DSM501_PM25);

void setup() {
  Serial.begin(9600); //for output information

  //Initialize DSM501
  dsm501.begin(MIN_WIN_SPAN);
}
```

```

void loop() {

    int adc_MQ7 = analogRead(A0);    //Lemos la salida analógica del MQ
    int adc_MQ131 = analogRead(A1); //Lemos la salida analógica del MQ
    int adc_MQ136 = analogRead(A2); //Lemos la salida analógica del MQ

    Serial.print("\n");
    float voltaje1 = adc_MQ7 * (5.0 /
1023.0);          //Convertimos la lectura en un valor de
voltaje
    float Rs1 = 1000 * ((5 - voltaje1) /
voltaje1);       //Calculamos Rs con un RL de 1k
    double DióxidoCarbono = 19.837 * pow(Rs1 / 0.22, -0.659); //
calculamos la concentración de alcohol con la ecuación obtenida.
    double Microgramocubico1 = DióxidoCarbono * 28 * pow(10, 3) / 24.5;

    Serial.print("  adc1:");
    Serial.println(adc_MQ7);
    Serial.print("  voltaje1:");
    Serial.println(voltaje1);
    Serial.print("  Rs1:");
    Serial.println(Rs1);
    Serial.print("  DióxidoCarbono:");
    Serial.print(DióxidoCarbono);
    Serial.println(" PPM");
    Serial.print("  DióxidoCarbono:");
    Serial.print(Microgramocubico1);
    Serial.print(" Ug/m3");
    Serial.print("\n");
}

```

```

Serial.print("\n");
float voltaje2 = adc_MQ131 * (5.0 / 1023.0); //Convertimos la
lectura en un valor de voltaje
float Rs2 = 1000 * ((5 - voltaje2) / voltaje2); //Calculamos Rs
con un RL de 1k
double Ozono = 17.88 * pow(Rs2 / 0.81, -0.905); // calculamos la
concentración de alcohol con la ecuación obtenida.
double Microgramocubico2 = Ozono * 48 * pow(10, 3) / 24.5;

float Rs2_1 = 1 * ((5 - voltaje2) / voltaje2); //Calculamos Rs con
un RL de 1k
float Ro2_1 = (((5 - voltaje2) / voltaje2) / 12.14);
float ratio2_1 = Rs2_1 / 0.74;
double valor2_1 = logf(ratio2_1) - 0.954;
float val2_1 = valor2_1 / -0.42;
float v2_1 = val2_1 + 0.698;
double DioxidoNitrogeno = pow(10, v2_1);
float Microgramocubico2_1 = DioxidoNitrogeno * 46 * pow(10, 3) /
24.5;

Serial.print("  adc2:");
Serial.println(adc_MQ131);
Serial.print("  voltaje2:");
Serial.println(voltaje2);
Serial.print("  Rs2_1:");
Serial.println(Rs2_1);
Serial.print("  Ro2_1:");
Serial.println(Ro2_1);
Serial.print("  Ozono:");
Serial.print(Ozono);
Serial.println("PPM");

```

```

Serial.print("  Ozono:");
Serial.print(Microgramocubico2);
Serial.println(" Ug/m3:");

Serial.print("  DioxidoNitrogeno:");
Serial.print(DioxidoNitrogeno);
Serial.println(" PPM");
Serial.print("  DioxidoNitrogeno:");
Serial.print(Microgramocubico2_1);
Serial.print(" Ug/m3");
Serial.print("\n");

Serial.print("\n");
float voltaje3 = adc_MQ136 * (5.0 / 1023.0); //Convertimos la
lectura en un valor de voltaje
float Rs3 = 1 * ((5 - voltaje3) / voltaje3); //Calculamos Rs con
un RL de 1k
//float  Ro3= (((5-voltaje3)/voltaje3)/3.51);
float ratio3 = Rs3 / 0.5;
double valor3 = logf(ratio3) - 0.056;
float val3 = valor3 / -0.367;
float v3 = val3 + 1;
float DioxidoAzufre = pow(10, v3);
float Microgramocubico3 = DioxidoAzufre * 64 * pow(10, 3) / 24.5;

Serial.print("  adc3:");
Serial.println(adc_MQ136);
Serial.print("  voltaje3:");
Serial.println(voltaje3);
Serial.print("  Rs3:");

```

```

Serial.println(Rs3);
Serial.print("  Ro3:");
Serial.println(Ro3);
Serial.print("  DioxidoAzufre:");
Serial.print(DioxidoAzufre);
Serial.println(" PPM");
Serial.print("  DioxidoAzufre:");
Serial.print(Microgramocubico3);
Serial.print(" Ug/m3:");
delay(5000);
Serial.print("\n");

// llamar a dsm501 para gestionar las actualizaciones
dsm501.update();

// obtener densidad PM de partículas de más de 10 µm
Serial.print("\n");
Serial.print("PM_10: ");
Serial.print(dsm501.getParticleWeight(0));
Serial.println(" ug/m3");

// obtener densidad PM de partículas de más de 2.5 µm
Serial.print("PM_2.5: ");
Serial.print(dsm501.getParticleWeight(1));
Serial.println(" ug/m3");

}

```

Anexo 2. Conexión I2C entre el Arduino UNO y el Modulo Cubecell.

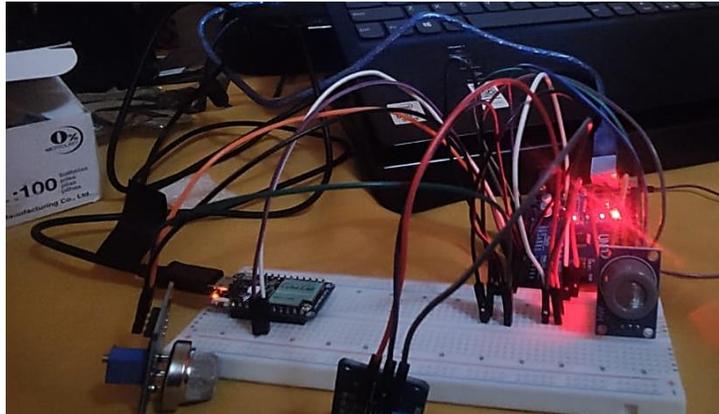


Figura 135. Conexión I2C.

Fuente: Elaborado por el autor

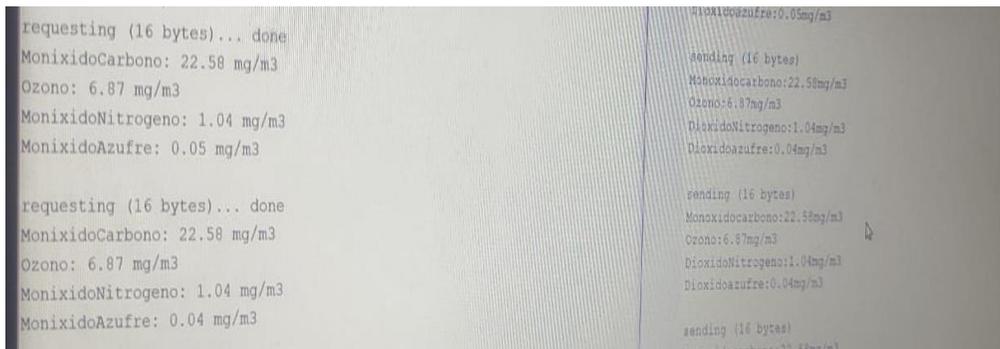


Figura 136. Visualización del envío y recepción de los datos mediante I2C.

Fuente: Elaborado por el auto

- **Código Modulo Cubecell Maestro**

```
#include <Wire.h>
#include <DSM501.h>

int periodo = 1000;
unsigned long t_n = 0;
int contador = 0;
float dato1;
float dato2;
float dato2_1;
```

```

float dato3;
float dato10;
float dato25;

struct TransmitData
{
    float Microgramocubico1;
    float Microgramocubico2;
    float Microgramocubico2_1;
    float Microgramocubico3;
    float Microgramocubico10;
    float Microgramocubico25;
};

TransmitData data;

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

void loop() {
    tiempo();
    if (contador == 30) {
        Serial.print("requesting ("); Serial.print(sizeof data);
Serial.print(" bytes)... ");
        if (Wire.requestFrom(2, sizeof data)) {
            Wire.readBytes((byte*) &data, sizeof data);
            Serial.println("done");

            Serial.print("MonixidoCarbono: ");

```

```

Serial.print(data.Microgramocubico1);
data.Microgramocubico1 = dato1;
Serial.println(" ug/m3");

Serial.print("Ozono: ");
Serial.print(data.Microgramocubico2);
  data.Microgramocubico2 = dato2;
Serial.println(" ug/m3");

Serial.print("MonixidoNitrogeno: ");
Serial.print(data.Microgramocubico2_1);
  data.Microgramocubico2_1 = dato2_1;
Serial.println(" ug/m3");

Serial.print("MonixidoAzufre: ");
Serial.print(data.Microgramocubico3);
  data.Microgramocubico3 = dato3;
Serial.println(" ug/m3");

Serial.print(data.Microgramocubico10);
  data.Microgramocubico10 = dato10;
Serial.println(" ug/m3");

Serial.print(data.Microgramocubico25);
  data.Microgramocubico25 = dato25;
Serial.println(" ug/m3");
Serial.print("\n");
contador=0;
}
else {
  Serial.println("could not connect");
}

```

```

    }
} else {
    Serial.print("tiempo=");
    Serial.println(contador);
}
delay(1000);
}

void tiempo() {
    if (millis() > t_n + periodo) {
        t_n = millis();
        contador++;
    }
}
}

```

- **Código Ardiono UNO Esclavo**

```

#include <Wire.h>
#include <DSM501.h>

#define DSM501_PM10 5
#define DSM501_PM25 6

DSM501 dsm501(DSM501_PM10, DSM501_PM25);
struct TransmitData {
    float Microgramocubico1;
    float Microgramocubico2;
    float Microgramocubico2_1;
    float Microgramocubico3;
    float Microgramocubico10;
    float Microgramocubico25;
}

```

```

};

TransmitData data;

void setup() {
  Serial.begin(9600);
  Wire.begin(2);
  Wire.onRequest(requestEvent);

  //Serial.begin(9600); //for output information
  //Initialize DSM501
  dsm501.begin(MIN_WIN_SPAN);
}

void loop() {
  //delay(1000);
  sensormq7();
  Serial.print("DióxidoCarbono:");
  Serial.print(data.Microgramocubico1);
  Serial.print("ug/m3");

  sensormq131();
  Serial.print("Ozono:");
  Serial.print(data.Microgramocubico2);
  Serial.println("ug/m3");

  Serial.print("DioxidoNitrogeno:");
  Serial.print(data.Microgramocubico2_1);
  Serial.print("ug/m3");

  sensormq136();

```

```

Serial.print("Dioxidoazufre:");
Serial.print(data.Microgramocubico3);
Serial.println("ug/m3");

sensordsm501();
Serial.print("PM_10: ");
Serial.print(dsm501.getParticleWeight(0));
data.Microgramocubico10=dsm501.getParticleWeight(0);
Serial.println(" ug/m3");

// obtener densidad PM de partículas de más de 2.5 µm
Serial.print("PM_2.5: ");
Serial.print(dsm501.getParticleWeight(1));
data.Microgramocubico25=dsm501.getParticleWeight(1);
Serial.println(" ug/m3");

Serial.print("\n");

}

void requestEvent() {
  Serial.print("sending (");
  Serial.print(sizeof data);
  Serial.println(" bytes)");
  Wire.write((byte *)&data, sizeof data);
}

void sensormq7() {
  int adc_MQ7 = analogRead(A0); //Lemos la salida analógica del MQ

  Serial.print("\n");

```

```

    float voltaje1 = adc_MQ7 * (5.0 /
1023.0);          //Convertimos la lectura en un valor de
voltaje
    float Rs1 = 1000 * ((5 - voltaje1) /
voltaje1);       //Calculamos Rs con un RL de 1k
    float DióxidoCarbono = 19.837 * pow(Rs1 / 0.22, -0.659); //
calculamos la concentración de alcohol con la ecuación obtenida.
    data.Microgramocubico1=DióxidoCarbono * 28 * pow(10, 3) / 24.5;
}
void sensormq131() {
    int adc_MQ131 = analogRead(A1);

    Serial.print("\n");
    float voltaje2 = adc_MQ131 * (5.0 / 1023.0);    //Convertimos la
lectura en un valor de voltaje
    float Rs2 = 1000 * ((5 - voltaje2) / voltaje2); //Calculamos Rs
con un RL de 1k
    float Ozono = 17.88 * pow(Rs2 / 0.81, -0.905); //
    data.Microgramocubico2 = Ozono * 48 * pow(10, 3) / 24.5;

    float Rs2_1=1*((5-voltaje2)/voltaje2); //Calculamos Rs con un RL
de 1k
    float Ro2_1= (((5-voltaje2)/voltaje2)/12.14);
    float ratio2_1= Rs2_1/0.74;
    float valor2_1= logf(ratio2_1)-0.954;
    float val2_1= valor2_1/-0.42;
    float v2_1= val2_1+0.698;
    float DioxidoNitrogeno= pow(10, v2_1);
    data.Microgramocubico2_1 = DioxidoNitrogeno*46*pow(10,3) / 24.5;
}

```

```

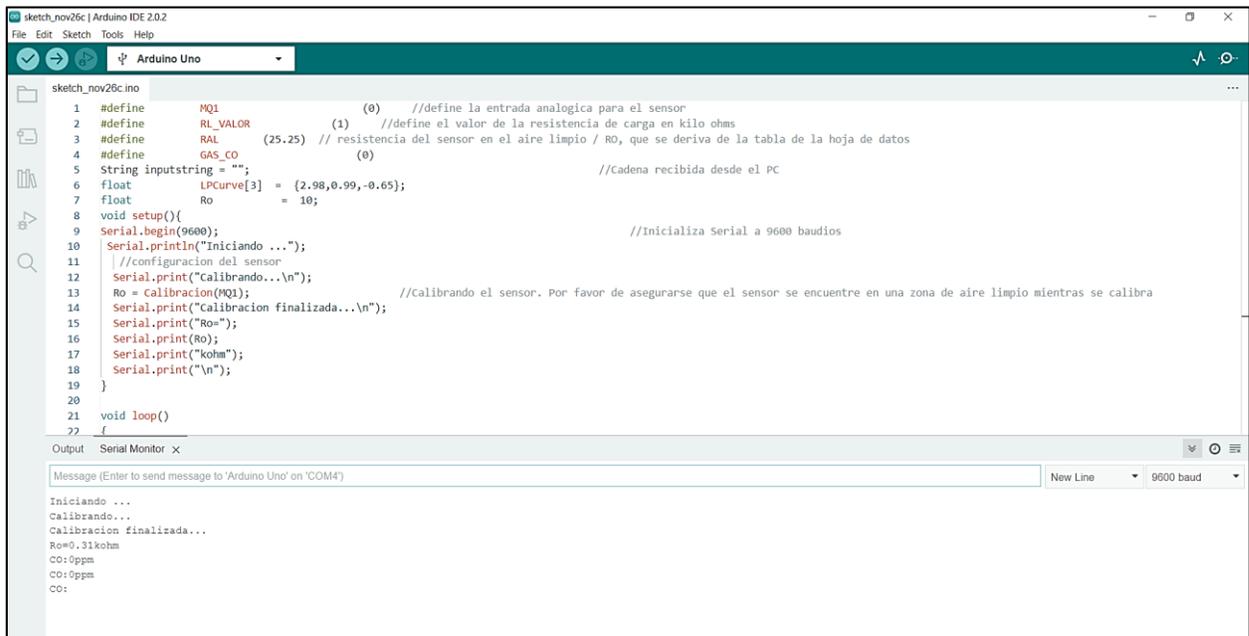
}
void sensormq136(){
int adc_MQ136 = analogRead(A2); //Lemos la salida analógica del MQ

Serial.print("\n");
float voltaje3 = adc_MQ136 * (5.0 / 1023.0); //Convertimos la
lectura en un valor de voltaje
float Rs3=1*((5-voltaje3)/voltaje3); //Calculamos Rs con un RL de
1k
float ratio3= Rs3/0.5;
double valor3= logf(ratio3)-0.056;
float val3= valor3/-0.367;
float v3= val3+1;
float DioxidoAzufre= pow(10, v3);
data.Microgramocubico3= DioxidoAzufre*64*pow(10,3) / 24.5;
}
void sensordsm501()
{
// llamar a dsm501 para gestionar las actualizaciones
dsm501.update();

}

```

Anexo 3. Calibración sensor MQ7



```
1 #define      MQ1                (0) //define la entrada analogica para el sensor
2 #define      RL_VALOR          (1) //define el valor de la resistencia de carga en kilo ohms
3 #define      RAL                (25.25) // resistencia del sensor en el aire limpio / RO, que se deriva de la tabla de la hoja de datos
4 #define      GAS_CO            (0)
5 String inputstring = ""; //Cadena recibida desde el PC
6 float        LPCurve[3] = {2.98,0.99,-0.65};
7 float        Ro              = 10;
8 void setup(){
9   Serial.begin(9600); //Inicializa Serial a 9600 baudios
10  Serial.println("Iniciando ...");
11  //configuracion del sensor
12  Serial.print("Calibrando...\n");
13  Ro = Calibracion(MQ1); //Calibrando el sensor. Por favor de asegurarse que el sensor se encuentre en una zona de aire limpio mientras se calibra
14  Serial.print("Calibracion finalizada...\n");
15  Serial.print("Ro=");
16  Serial.print(Ro);
17  Serial.print("kohm");
18  Serial.print("\n");
19 }
20
21 void loop()
22 {
23 }
```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on 'COM4')

New Line 9600 baud

```
Iniciando ...
Calibrando...
Calibracion finalizada...
Ro=0.31kohm
CO:0ppm
CO:0ppm
CO:
```

Figura 137. Proceso de Calibración en el IDE de Arduino.

Fuente: Elaborado por el autor

Código utilizado para la calibración de los sensores MQ7.

```
#define      MQ1                (0) //define la entrada
analógica para el sensor
#define      RL_VALOR          (1) //define el valor de la
resistencia de carga en kilo ohms
#define      RAL                (25.25) // resistencia del sensor en el
aire limpio / RO, que se deriva de la tabla de la hoja de datos
#define      GAS_CO            (0)
String inputstring =
""; //Cadena
recibida desde el PC
float        LPCurve[3] = {2.98,0.99,-0.65};
float        Ro              = 10;
void setup(){
```

```

Serial.begin(9600);
    //Inicializa Serial a 9600 baudios
Serial.println("Iniciando ...");
    //configuración del sensor
Serial.print("Calibrando...\n");
    Ro = Calibracion(MQ1); //Calibrando el
sensor. Por favor de asegurarse que el sensor se encuentre en una
zona de aire limpio mientras se calibra
    Serial.print("Calibracion finalizada...\n");
Serial.print("Ro=");
Serial.print(Ro);
Serial.print("kohm");
Serial.print("\n");
}

void loop()
{
    Serial.print("CO:");
    Serial.print(porcentaje_gas(lecturaMQ(MQ1)/Ro,GAS_CO) );
    Serial.print( "ppm" );
    Serial.print(" ");
    Serial.print("\n");
    delay(200);
}

float calc_res(int raw_adc)
{
    return ( ((float)RL_VALOR*(1023-raw_adc)/raw_adc));
}

float Calibracion(float mq_pin){

```

```

int i;
float val=0;
  for (i=0;i<50;i++)
{
    //tomar múltiples muestras
    val += calc_res(analogRead(mq_pin));
    delay(500);
}
val =
val/50;

//calcular el valor medio
val = val/RAL;
return val;
}

```

```

float lecturaMQ(int mq_pin){
  int i;
  float rs=0;
  for (i=0;i<5;i++) {
    rs += calc_res(analogRead(mq_pin));
    delay(50);
  }
rs = rs/5;
return rs;
}

```

```

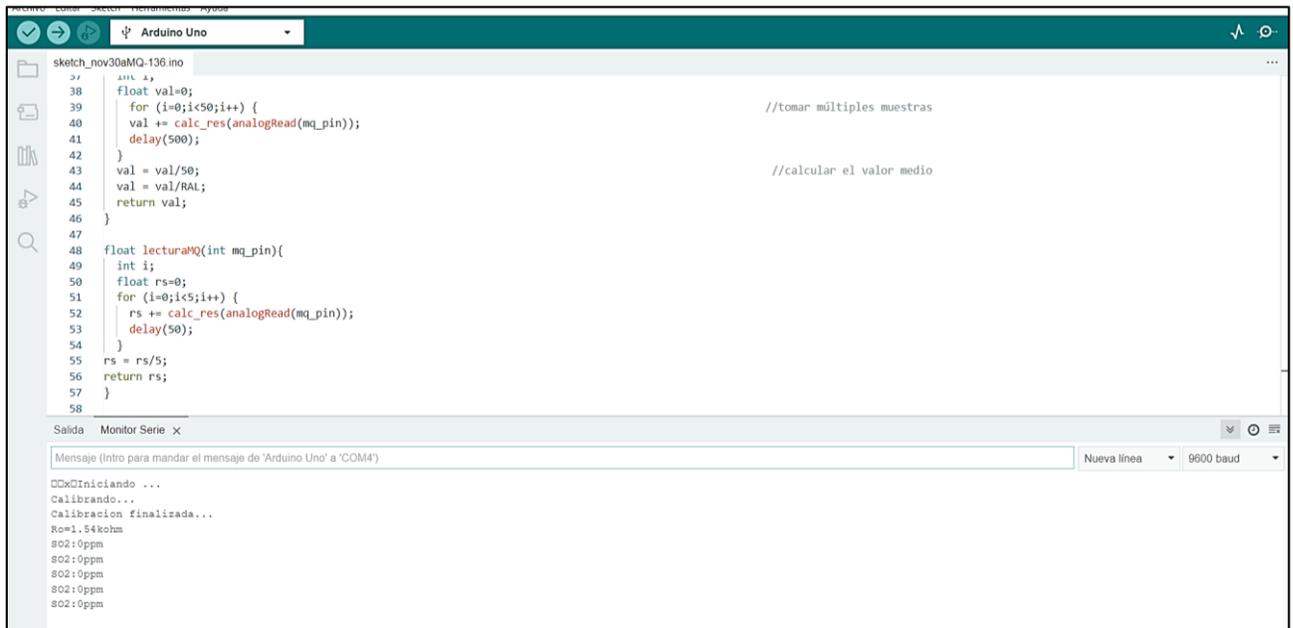
int porcentaje_gas(float rs_ro_ratio, int gas_id){
  if ( gas_id == GAS_CO ) {
    return porcentaje_gas(rs_ro_ratio,LPCurve);
  }
return 0;
}

```

```
}
```

```
int porcentaje_gas(float rs_ro_ratio, float *pcurve){  
    return (pow(10, (((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) +  
pcurve[0])));  
}
```

Anexo 4. Calibración sensor MQ136



```
sketch_nov30aMQ-136.ino
37
38 float val=0;
39 for (i=0;i<50;i++) { //tomar múltiples muestras
40   val += calc_res(analogRead(mq_pin));
41   delay(500);
42 }
43 val = val/50; //calcular el valor medio
44 val = val/RAL;
45 return val;
46 }
47
48 float lecturaMQ(int mq_pin){
49   int i;
50   float rs=0;
51   for (i=0;i<5;i++) {
52     rs += calc_res(analogRead(mq_pin));
53     delay(50);
54   }
55   rs = rs/5;
56   return rs;
57 }
58
```

Salida Monitor Serie x

```
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM4')
[Ox]Iniciando ...
Calibrando...
Calibracion finalizada...
Ro=1.54kohm
SO2:0ppm
SO2:0ppm
SO2:0ppm
SO2:0ppm
SO2:0ppm
```

Figura 138. *Proceso de calibración en el IDE de arduino.*

Fuente: *Elaborado por el autor*

Código utilizado para la calibración de los sensores MQ136

```
#define MQ136 (0) //define la entrada
analogica para el sensor
#define RL_VALOR (1) //define el valor de la
resistencia de carga en kilo ohms
#define RAL (3.51) // resistencia del sensor en el
aire limpio / R0, que se deriva de la tabla de la hoja de datos
#define GAS_SO2 (0)
String inputstring =
""; //Cadena
recibida desde el PC
float LPCurve[3] = {0.98,0.99,-0.27};
float Ro = 10;
void setup(){
```

```

Serial.begin(9600);
    //Inicializa Serial a 9600 baudios
Serial.println("Iniciando ...");
    //configuracion del sensor
Serial.print("Calibrando...\n");
    Ro = Calibracion(MQ136);           //Calibrando el
sensor. Por favor de asegurarse que el sensor se encuentre en una
zona de aire limpio mientras se calibra
    Serial.print("Calibracion finalizada...\n");
Serial.print("Ro=");
Serial.print(Ro);
Serial.print("kohm");
Serial.print("\n");
}

void loop()
{
    Serial.print("SO2:");
    Serial.print(porcentaje_gas(lecturaMQ(MQ136)/Ro,GAS_SO2) );
    Serial.print( "ppm" );
    Serial.print(" ");
    Serial.print("\n");
    delay(200);
}

float calc_res(int raw_adc)
{
    return ( ((float)RL_VALOR*(1023-raw_adc)/raw_adc));
}

float Calibracion(float mq_pin){

```

```

int i;
float val=0;
  for (i=0;i<50;i++)
{
    //tomar múltiples muestras
    val += calc_res(analogRead(mq_pin));
    delay(500);
}
val =
val/50;

//calcular el valor medio
val = val/RAL;
return val;
}

```

```

float lecturaMQ(int mq_pin){
  int i;
  float rs=0;
  for (i=0;i<5;i++) {
    rs += calc_res(analogRead(mq_pin));
    delay(50);
  }
rs = rs/5;
return rs;
}

```

```

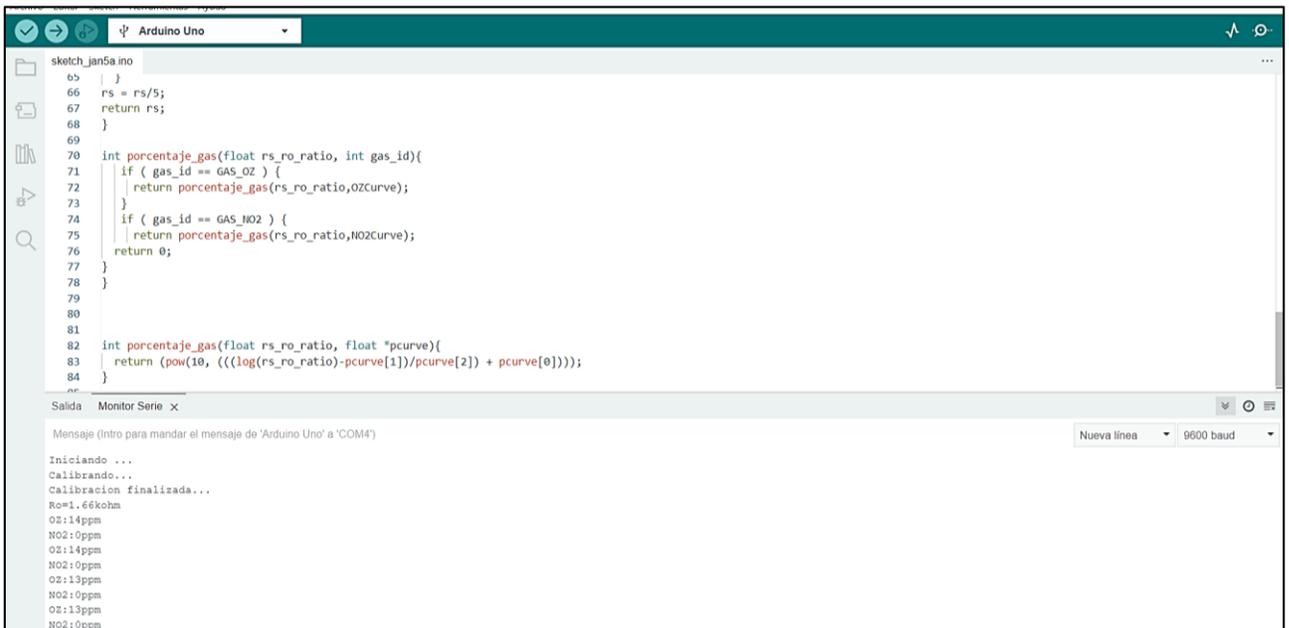
int porcentaje_gas(float rs_ro_ratio, int gas_id){
  if ( gas_id == GAS_SO2 ) {
    return porcentaje_gas(rs_ro_ratio,LPCurve);
  }
return 0;
}

```

```
}
```

```
int porcentaje_gas(float rs_ro_ratio, float *pcurve){  
    return (pow(10, (((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) +  
pcurve[0])));  
}
```

Anexo 5. Calibración sensor MQ131



```
sketch_jan5a.ino
65 | }
66 | rs = rs/5;
67 | return rs;
68 | }
69 |
70 | int porcentaje_gas(float rs_ro_ratio, int gas_id){
71 |   if ( gas_id == GAS_OZ ) {
72 |     return porcentaje_gas(rs_ro_ratio,OZCurve);
73 |   }
74 |   if ( gas_id == GAS_NO2 ) {
75 |     return porcentaje_gas(rs_ro_ratio,NO2Curve);
76 |   }
77 |   return 0;
78 | }
79 |
80 |
81 |
82 | int porcentaje_gas(float rs_ro_ratio, float *pcurve){
83 |   return (pow(10, (((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) + pcurve[0])));
84 | }
```

Salida Monitor Serie x

Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM4')

Nueva línea 9600 baud

Iniciando ...
Calibrando...
Calibracion finalizada...
Ro=1.66kohm
O2:14ppm
NO2:0ppm
OZ:14ppm
NO2:0ppm
OZ:13ppm
NO2:0ppm
OZ:13ppm
NO2:0ppm

Figura 139. *Proceso de calibración en el IDE de arduino.*

Fuente: *Elaborado por el autor*

Código utilizado para la calibración de los sensores MQ131.

```
#define MQ131 (0) //define la entrada  
analogica para el sensor  
#define RL_VALOR (1) //define el valor de la  
resistencia de carga en kilo ohms  
#define RAL (12.14) // resistencia del sensor en el  
aire limpio / RO, que se deriva de la tabla de la hoja de datos  
#define GAS_OZ (0)  
#define GAS_NO2 (1)
```

```
String inputstring =  
""; //Cadena  
recibida desde el PC  
float OZCurve[3] = {2.88,0.99,-0.90};  
float Ro = 10;
```

```

float          NO2Curve[3]  = {2.74,0.99,-0.44};

void setup(){
Serial.begin(9600);
          //Inicializa Serial a 9600 baudios
Serial.println("Iniciando ...");
  //configuracion del sensor
Serial.print("Calibrando...\n");
  Ro = Calibracion(MQ131);          //Calibrando el
sensor. Por favor de asegurarse que el sensor se encuentre en una
zona de aire limpio mientras se calibra
Serial.print("Calibracion finalizada...\n");
Serial.print("Ro=");
Serial.print(Ro);
Serial.print("kohm");
Serial.print("\n");
}

void loop()
{
  Serial.print("OZ:");
  Serial.print(porcentaje_gas(lecturaMQ(MQ131)/Ro,GAS_OZ) );
  Serial.print( "ppm" );
  Serial.print("  ");
  Serial.print("\n");
  delay(200);

  Serial.print("NO2:");
  Serial.print(porcentaje_gas(lecturaMQ(MQ131)/Ro,GAS_NO2) );
  Serial.print( "ppm" );
  Serial.print("  ");
}

```

```

    Serial.print("\n");
    delay(200);
}

float calc_res(int raw_adc)
{
    return ( ((float)RL_VALOR*(1023-raw_adc)/raw_adc));
}

float Calibracion(float mq_pin){
    int i;
    float val=0;
    for (i=0;i<50;i++)
    {
        //tomar múltiples muestras
        val += calc_res(analogRead(mq_pin));
        delay(500);
    }
    val =
val/50;
//calcular el valor medio

    val = val/RAL;
    return val;
}

float lecturaMQ(int mq_pin){
    int i;
    float rs=0;
    for (i=0;i<5;i++) {
        rs += calc_res(analogRead(mq_pin));
        delay(50);
    }
}

```

```
    }  
    rs = rs/5;  
    return rs;  
}
```

```
int porcentaje_gas(float rs_ro_ratio, int gas_id){  
    if ( gas_id == GAS_OZ ) {  
        return porcentaje_gas(rs_ro_ratio,OZCurve);  
    }  
    if ( gas_id == GAS_NO2 ) {  
        return porcentaje_gas(rs_ro_ratio,NO2Curve);  
    }  
    return 0;  
}  
}
```

```
int porcentaje_gas(float rs_ro_ratio, float *pcurve){  
    return (pow(10, (((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) +  
pcurve[0]))));  
}
```

Anexo 6. Calibración Sensor DSM501

```
#include <DSM501.h>

#define DSM501_PM10 9
#define DSM501_PM25 8

DSM501 dsm501(DSM501_PM10, DSM501_PM25);

void setup() {
    Serial.begin(9600); //for output information

    //Initialize DSM501
    dsm501.begin(MIN_WIN_SPAN);
}

void loop() {
    / llamar a dsm501 para gestionar las actualizaciones
    dsm501.update();

    // obtener densidad PM de partículas de más de 10 µm
    Serial.print("\n");
    Serial.print("PM_10: ");
    Serial.print(dsm501.getParticleWeight(0));
    Serial.println(" ug/m3");
    // obtener densidad PM de partículas de más de 2.5 µm
    Serial.print("PM_2.5: ");
    Serial.print(dsm501.getParticleWeight(1));
    Serial.println(" ug/m3");
}
```

Anexo 7. Código en el IDE de arduino para el envío de los datos del módulo 1

```
#include "LoRaWan_APP.h"
#include "Arduino.h"
#include "Wire.h"

struct TransmitData {
  float s1;
  float s2;
  float s3;
  float s4;
  float s5;
  float s6;
};

bool paquete1 = false; // Variable para el envío del paquete 1
bool paquete2 = true; // Variable para el envío del paquete 2

TransmitData data;

/*
  set LoraWan_RGB to Active,the RGB active in loraWan
  RGB red means sending;
  RGB purple means joined done;
  RGB blue means RxWindow1;
  RGB yellow means RxWindow2;
  RGB green means received done;
*/

/* OTAA para*/
uint8_t devEui[] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0x9C, 0xDA };
uint8_t appEui[] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };
```

```
uint8_t appKey[] = { 0x47, 0xCC, 0x3E, 0x4D, 0xE2, 0x5F, 0xBA, 0x93, 0xF6, 0xD6, 0x0A,  
0x35, 0x16, 0x17, 0xFD, 0xD6 };
```

```
/* ABP para*/
```

```
uint8_t nwkSKey[] = { 0x15, 0xb1, 0xd0, 0xef, 0xa4, 0x63, 0xdf, 0xbe, 0x3d, 0x11, 0x18,  
0x1e, 0x1e, 0xc7, 0xda, 0x85 };
```

```
uint8_t appSKey[] = { 0xd7, 0x2c, 0x78, 0x75, 0x8c, 0xdc, 0xca, 0xbf, 0x55, 0xee, 0x4a,  
0x77, 0x8d, 0x16, 0xef, 0x67 };
```

```
uint32_t devAddr = (uint32_t)0x007e6ae1;
```

```
/*LoraWan channelsmask, default channels 0-7*/
```

```
uint16_t userChannelsMask[6] = { 0x00FF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 };
```

```
/*LoraWan region, select in arduino IDE tools*/
```

```
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;
```

```
/*LoraWan Class, Class A and Class C are supported*/
```

```
DeviceClass_t loraWanClass = LORAWAN_CLASS;
```

```
/*the application data transmission duty cycle. value in [ms].*/
```

```
uint32_t appTxDutyCycle = 300000;
```

```
/*OTAA or ABP*/
```

```
bool overTheAirActivation = LORAWAN_NETMODE;
```

```
/*ADR enable*/
```

```
bool loraWanAdr = LORAWAN_ADR;
```

```
/* set LORAWAN_Net_Reserve ON, the node could save the network info to flash, when  
node reset not need to join again */
```

```
bool keepNet = LORAWAN_NET_RESERVE;
```

```
/* Indicates if the node is sending confirmed or unconfirmed messages */
```

```
bool isTxConfirmed = LORAWAN_UPLINKMODE;
```

```
/* Application port */
```

```
uint8_t appPort = 2;
```

```
/*!
```

Number of trials to transmit the frame, if the LoRaMAC layer did not receive an acknowledgment. The MAC performs a datarate adaptation, according to the LoRaWAN Specification V1.0.2, chapter 18.4, according to the following table:

Transmission nb | Data Rate

-----|-----

1 (first) | DR

2 | DR

3 | max(DR-1,0)

4 | max(DR-1,0)

5 | max(DR-2,0)

6 | max(DR-2,0)

7 | max(DR-3,0)

8 | max(DR-3,0)

Note, that if NbTrials is set to 1 or 2, the MAC will not decrease the datarate, in case the LoRaMAC layer did not receive an acknowledgment

```
*/
```

```
uint8_t confirmedNbTrials = 4;
```

```
/* Prepares the payload of the frame */
```

```
static void prepareTxFrame(uint8_t port) {
```

```
/*appData size is LORAWAN_APP_DATA_MAX_SIZE which is defined in
"commissioning.h".
    appDataSize max value is LORAWAN_APP_DATA_MAX_SIZE.
    if enabled AT, don't modify LORAWAN_APP_DATA_MAX_SIZE, it may cause system
hanging or failure.
    if disabled AT, LORAWAN_APP_DATA_MAX_SIZE can be modified, the max value is
reference to lorawan region and SF.
    for example, if use REGION_CN470,
    the max value for different DR can be found in MaxPayloadOfDatarateCN470 refer to
DataratesCN470 and BandwidthsCN470 in "RegionCN470.h".
*/
appDataSize = 4;
```

```
Serial.print("requesting (");
Serial.print(sizeof data);
Serial.print(" bytes)... ");
if (Wire.requestFrom(2, sizeof data)) {
    Wire.readBytes((byte*)&data, sizeof data);
    Serial.println("done");
    delay(1000);
    Serial.print("MonixidoCarbono: ");
    Serial.print(data.s1);
    Serial.println(" ug/m3");

    Serial.print("Ozono: ");
    Serial.print(data.s2);
    Serial.println(" ug/m3");

    Serial.print("DioxidoNitrogeno: ");
    Serial.print(data.s3);
    Serial.println(" ug/m3");
```

```

Serial.print("DioxidoAzufre: ");
Serial.print(data.s4);
Serial.println(" ug/m3");
delay(1000);

int base1 = 100;
/* Condicional para el envio del paquete 1, al gateway y la plataforma de Things Network
*/
if (paquete2 == true && paquete1 == false) {
  /* Envio del paquete 1*/
  appData[0] = int(round(data.s1 * 10)) - base1;
  appData[1] = int(round(data.s2 * 10));
  appData[2] = int(round(data.s3 * 10));
  appData[3] = int(round(data.s4 * 10));
  paquete1 = true;
  paquete2 = false;
  /* Cumplida la analogia del paquete 1, se envia el paquete 2*/
} else {
  /* Envio del paquete 1*/
  appData[0] = 255;
  appData[1] = 0;
  appData[2] = int(round(data.s5 * 10));
  appData[3] = int(round(data.s6 * 10));
  paquete2 = true;
  paquete1 = false;
}
} else { // Caso conteoio se imprime un mensaje que no se pudo conectar
  Serial.println("could not connect");
}
}
}

```

```

void setup() {
  Serial.begin(115200);
  Wire.begin();
  #if (AT_SUPPORT)
    enableAt();
  #endif
  deviceState = DEVICE_STATE_INIT;
  LoRaWAN.ifskipjoin();
}

void loop() {
  switch (deviceState) {
    case DEVICE_STATE_INIT:
      {
        #if (LORAWAN_DEVEUI_AUTO)
          LoRaWAN.generateDeveuiByChipID();
        #endif
        #if (AT_SUPPORT)
          getDevParam();
        #endif
        printDevParam();
        LoRaWAN.init(loraWanClass, loraWanRegion);
        deviceState = DEVICE_STATE_JOIN;
        break;
      }
    case DEVICE_STATE_JOIN:
      {
        LoRaWAN.join();
        break;
      }
  }
}

```

```

case DEVICE_STATE_SEND:
{
    prepareTxFrame(appPort);
    LoRaWAN.send();
    deviceState = DEVICE_STATE_CYCLE;
    break;
}
case DEVICE_STATE_CYCLE:
{
    // Schedule next packet transmission
    txDutyCycleTime = appTxDutyCycle + randr(0, APP_TX_DUTYCYCLE_RND);
    LoRaWAN.cycle(txDutyCycleTime);
    deviceState = DEVICE_STATE_SLEEP;
    break;
}
case DEVICE_STATE_SLEEP:
{
    LoRaWAN.sleep();
    break;
}
default:
{
    deviceState = DEVICE_STATE_INIT;
    break;
}
}
}

String convertDecimalToHex(long n) {

String hexNum;
long remainder;

```

```
while (n > 0) {  
  
    remainder = n % 16;  
    n = n / 16;  
  
    if (remainder < 10) {  
        hexNum = String(remainder) + hexNum;  
    }  
  
    else {  
  
        switch (remainder) {  
  
            case 10:  
                hexNum = "A" + hexNum;  
                break;  
  
            case 11:  
                hexNum = "B" + hexNum;  
                break;  
  
            case 12:  
                hexNum = "C" + hexNum;  
                break;  
  
            case 13:  
                hexNum = "D" + hexNum;  
                break;  
  
            case 14:
```

```
    hexNum = "E" + hexNum;
    break;

    case 15:
        hexNum = "F" + hexNum;
        break;
    }
}

return hexNum;
}
```

Anexo 7. Código para el envío de los datos del módulo 2

```
#include "LoRaWan_APP.h"
#include "Arduino.h"

/*
 * set LoraWan_RGB to Active,the RGB active in loraWan
 * RGB red means sending;
 * RGB purple means joined done;
 * RGB blue means RxWindow1;
 * RGB yellow means RxWindow2;
 * RGB green means received done;
 */

/* OTAA para*/
uint8_t devEui[] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0x84, 0xDB };
uint8_t appEui[] = { 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01 };
uint8_t appKey[] = { 0xAE, 0x6D, 0xE9, 0x95, 0x03, 0xFF, 0x74, 0xA4, 0xA7, 0x01, 0xD8,
0x33, 0x80, 0xAD, 0x53, 0x10 };

struct TransmitData {
    float s1;
    float s2;
    float s3;
    float s4;
    float s5;
    float s6;
};

bool paquete1 = false; // Variable para el envío del paquete 1
bool paquete2 = true; // Variable para el envío del paquete 2

TransmitData data;

/*
    set LoraWan_RGB to Active,the RGB active in loraWan
```

```

    RGB red means sending;
    RGB purple means joined done;
    RGB blue means RxWindow1;
    RGB yellow means RxWindow2;
    RGB green means received done;
*/

/* OTAA para*/
uint8_t devEui[] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0x9C, 0xDA };
uint8_t appEui[] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };
uint8_t appKey[] = { 0x47, 0xCC, 0x3E, 0x4D, 0xE2, 0x5F, 0xBA, 0x93, 0xF6, 0xD6, 0x0A,
0x35, 0x16, 0x17, 0xFD, 0xD6 };

/* ABP para*/
uint8_t nwkSKey[] = { 0x15, 0xb1, 0xd0, 0xef, 0xa4, 0x63, 0xdf, 0xbe, 0x3d, 0x11, 0x18,
0x1e, 0x1e, 0xc7, 0xda, 0x85 };
uint8_t appSKey[] = { 0xd7, 0x2c, 0x78, 0x75, 0x8c, 0xdc, 0xca, 0xbf, 0x55, 0xee, 0x4a,
0x77, 0x8d, 0x16, 0xef, 0x67 };
uint32_t devAddr = (uint32_t)0x007e6ae1;

/*LoraWan channelsmask, default channels 0-7*/
uint16_t userChannelsMask[6] = { 0x00FF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 };

/*LoraWan region, select in arduino IDE tools*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;

/*LoraWan Class, Class A and Class C are supported*/
DeviceClass_t loraWanClass = LORAWAN_CLASS;

/*the application data transmission duty cycle. value in [ms].*/
uint32_t appTxDutyCycle = 300000;

```

```
/*OTAA or ABP*/
```

```
bool overTheAirActivation = LORAWAN_NETMODE;
```

```
/*ADR enable*/
```

```
bool loraWanAdr = LORAWAN_ADR;
```

```
/* set LORAWAN_Net_Reserve ON, the node could save the network info to flash, when  
node reset not need to join again */
```

```
bool keepNet = LORAWAN_NET_RESERVE;
```

```
/* Indicates if the node is sending confirmed or unconfirmed messages */
```

```
bool isTxConfirmed = LORAWAN_UPLINKMODE;
```

```
/* Application port */
```

```
uint8_t appPort = 2;
```

```
/*!
```

Number of trials to transmit the frame, if the LoRaMAC layer did not receive an acknowledgment. The MAC performs a datarate adaptation, according to the LoRaWAN Specification V1.0.2, chapter 18.4, according to the following table:

Transmission nb | Data Rate

Transmission nb	Data Rate
1 (first)	DR
2	DR
3	max(DR-1,0)
4	max(DR-1,0)
5	max(DR-2,0)
6	max(DR-2,0)
7	max(DR-3,0)

8 | max(DR-3,0)

Note, that if NbTrials is set to 1 or 2, the MAC will not decrease the datarate, in case the LoRaMAC layer did not receive an acknowledgment

*/

```
uint8_t confirmedNbTrials = 4;
```

```
/* Prepares the payload of the frame */
```

```
static void prepareTxFrame(uint8_t port) {
```

```
/*appData size is LORAWAN_APP_DATA_MAX_SIZE which is defined in "commissioning.h".
```

```
appDataSize max value is LORAWAN_APP_DATA_MAX_SIZE.
```

```
if enabled AT, don't modify LORAWAN_APP_DATA_MAX_SIZE, it may cause system hanging or failure.
```

```
if disabled AT, LORAWAN_APP_DATA_MAX_SIZE can be modified, the max value is reference to lorawan region and SF.
```

```
for example, if use REGION_CN470,
```

```
the max value for different DR can be found in MaxPayloadOfDatarateCN470 refer to DataratesCN470 and BandwidthsCN470 in "RegionCN470.h".
```

```
*/
```

```
appDataSize = 4;
```

```
Serial.print("requesting (");
```

```
Serial.print(sizeof data);
```

```
Serial.print(" bytes)... ");
```

```
if (Wire.requestFrom(2, sizeof data)) {
```

```
Wire.readBytes((byte*)&data, sizeof data);
```

```
Serial.println("done");
```

```
delay(1000);
```

```
Serial.print("MonixidoCarbono: ");
```

```
Serial.print(data.s1);
```

```
Serial.println(" ug/m3");
```

```
Serial.print("Ozono: ");
```

```
Serial.print(data.s2);
```

```
Serial.println(" ug/m3");
```

```
Serial.print("DioxidoNitrogeno: ");
```

```
Serial.print(data.s3);
```

```
Serial.println(" ug/m3");
```

```
Serial.print("DioxidoAzufre: ");
```

```
Serial.print(data.s4);
```

```
Serial.println(" ug/m3");
```

```
delay(1000);
```

```
int base1 = 100;
```

```
/* Condicional para el envio del paquete 1, al gateway y la plataforma de Things Network
```

```
*/
```

```
if (paquete2 == true && paquete1 == false) {
```

```
/* Envio del paquete 1*/
```

```
appData[0] = int(round(data.s1 * 10)) - base1;
```

```
appData[1] = int(round(data.s2 * 10));
```

```
appData[2] = int(round(data.s3 * 10));
```

```
appData[3] = int(round(data.s4 * 10));
```

```
paquete1 = true;
```

```
paquete2 = false;
```

```
/* Cumplida la analogia del paquete 1, se envia el paquete 2*/
```

```
} else {
```

```
/* Envio del paquete 1*/
```

```
appData[0] = 255;
```

```
appData[1] = 0;
```

```

    appData[2] = int(round(data.s5 * 10));
    appData[3] = int(round(data.s6 * 10));
    paquete2 = true;
    paquete1 = false;
}
} else { // Caso conteo se imprime un mensaje que no se pudo conectar
    Serial.println("could not connect");
}
}
}

```

```

void setup() {
    Serial.begin(115200);
    Wire.begin();
#ifdef (AT_SUPPORT)
    enableAt();
#endif
    deviceState = DEVICE_STATE_INIT;
    LoRaWAN.ifskipjoin();
}

```

```

void loop() {
    switch (deviceState) {
        case DEVICE_STATE_INIT:
            {
#ifdef (LORAWAN_DEVEUI_AUTO)
                LoRaWAN.generateDeveuiByChipID();
#endif
#ifdef (AT_SUPPORT)
                getDevParam();
#endif
                printDevParam();
            }
        }
    }
}

```

```

    LoRaWAN.init(loraWanClass, loraWanRegion);
    deviceState = DEVICE_STATE_JOIN;
    break;
}
case DEVICE_STATE_JOIN:
{
    LoRaWAN.join();
    break;
}
case DEVICE_STATE_SEND:
{
    prepareTxFrame(appPort);
    LoRaWAN.send();
    deviceState = DEVICE_STATE_CYCLE;
    break;
}
case DEVICE_STATE_CYCLE:
{
    // Schedule next packet transmission
    txDutyCycleTime = appTxDutyCycle + randr(0, APP_TX_DUTYCYCLE_RND);
    LoRaWAN.cycle(txDutyCycleTime);
    deviceState = DEVICE_STATE_SLEEP;
    break;
}
case DEVICE_STATE_SLEEP:
{
    LoRaWAN.sleep();
    break;
}
default:
{

```

```
        deviceState = DEVICE_STATE_INIT;
        break;
    }
}
}
String convertDecimalToHex(long n) {
```

```
    String hexNum;
```

```
    long remainder;
```

```
    while (n > 0) {
```

```
        remainder = n % 16;
```

```
        n = n / 16;
```

```
        if (remainder < 10) {
```

```
            hexNum = String(remainder) + hexNum;
```

```
        }
```

```
    else {
```

```
        switch (remainder) {
```

```
            case 10:
```

```
                hexNum = "A" + hexNum;
```

```
                break;
```

```
            case 11:
```

```
                hexNum = "B" + hexNum;
```

```
                break;
```

case 12:

```
hexNum = "C" + hexNum;
```

```
break;
```

case 13:

```
hexNum = "D" + hexNum;
```

```
break;
```

case 14:

```
hexNum = "E" + hexNum;
```

```
break;
```

case 15:

```
hexNum = "F" + hexNum;
```

```
break;
```

```
}
```

```
}
```

```
}
```

```
return hexNum;
```

```
}
```

Anexo 8. Diseño del circuito

En las siguientes figuras 140 y 141, se muestra el desarrollo del circuito digital de los nodos sensores.

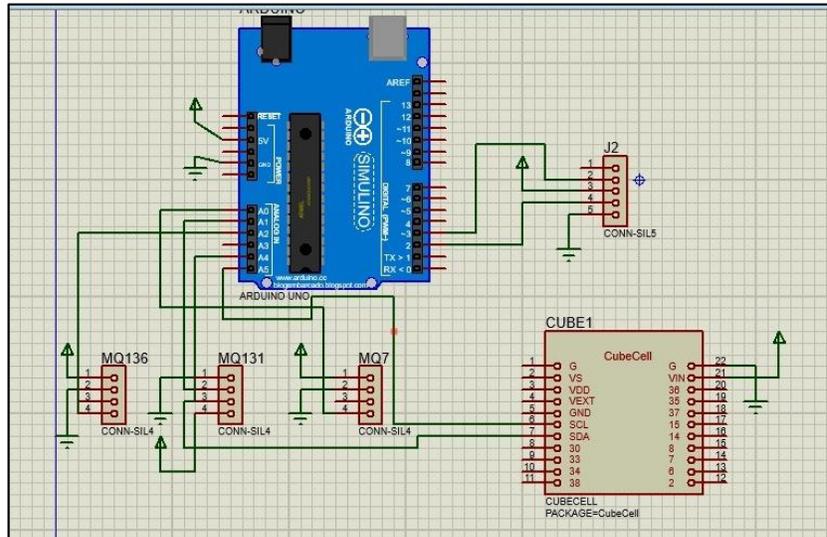


Figura 140. *Diseño del circuito digital en proteus.*

Fuente: *Elaborado por el autor, 2023*

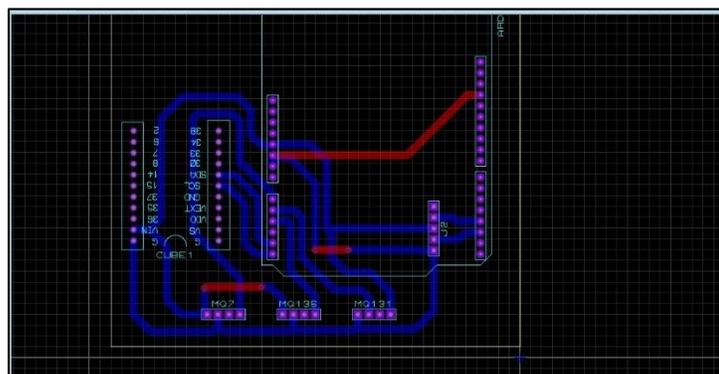


Figura 141. *Impresión del circuito en la vaquelita.*

Fuente: *Elaborado por el autor, 2023*

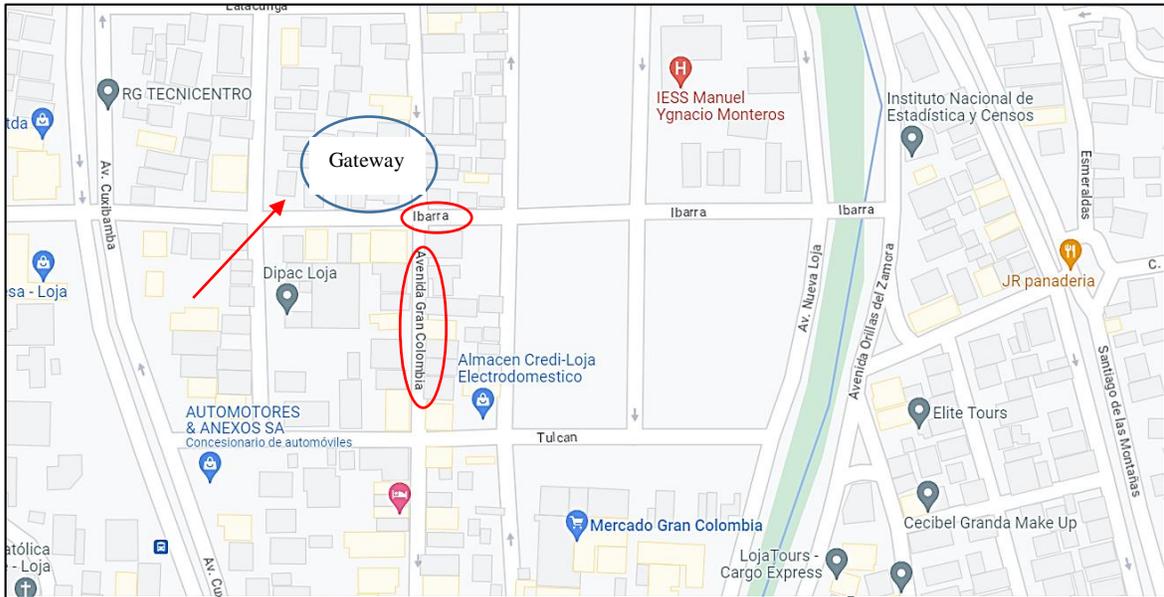
Anexo 9. Construcción de los nodos sensores



Anexo 10. Ubicación del Gateway

El Gateway se encuentra ubicado por el mercado mayorista entre la avenida gran Colombia e Ibarra; a 4 metros de altura sujeto a un tubo de metal como se observa en la imagen, donde su longitud: $3^{\circ} 59' 18''$ S y latitud: $79^{\circ} 12' 25''$ O.





Anexo 11. Certificado Traducción Resumen



Loja, 08 de diciembre de 2023

Lic. Karina Yajaira Martínez Luzuriaga

LICENCIADA EN CIENCIAS DE LA EDUCACIÓN MENCIÓN INGLÉS

CERTIFICO:

Yo, Karina Yajaira Martínez Luzuriaga con cédula de identidad Nro. 1104902679, Licenciada en Ciencias de la Educación Mención Inglés por la Universidad Técnica Particular de Loja, con número de registro 1031-2022-2574017 en la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación, señalo que el presente documento es fiel traducción del idioma español al idioma inglés del resumen del Trabajo de Titulación denominado **“Diseño e implementación de una red inalámbrica de sensores de bajo costo, para la medición de parámetros de la calidad del aire”** elaborado por el Sr. Paul Alexander Aponte Ortiz, con cédula de identidad Nro. 1105456287, estudiante egresado de la carrera de Ingeniería en Telecomunicaciones de la Universidad Nacional de Loja.



Lic. Karina Yajaira Martínez Luzuriaga

C.I. 1104902679

REGISTRO SENESCYT N°: 1031-2022-2574017