



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

**CARRERA DE INGENIERÍA ELECTRÓNICA CON MENCIÓN EN
TELECOMUNICACIONES.**

PROYECTO DE TITULACIÓN

Previa la obtención del Título de:

**INGENIERO ELECTRÓNICO CON MENCIÓN EN
TELECOMUNICACIONES.**

TEMA

“DISEÑO Y DESARROLLO DE UN PROTOTIPO DE RED DE SENSORES IOT
UTILIZANDO TECNOLOGÍA LORAWAN PARA EL MONITOREO DE
PARÁMETROS AMBIENTALES EN INTERIORES Y EXTERIORES.”

AUTORES

Narváez Narváez Karen Stefany

Contreras Pérez Víctor Andrés

DIRECTOR: PhD. Bremnen Veliz

GUAYAQUIL

2020

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

Nosotros, Karen Narváez Narváez y Víctor Contreras Pérez, estudiantes de la universidad Politécnica Salesiana, declaramos que los conceptos, análisis desarrollados y conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Guayaquil, 2020



Karen Narváez N.

C.I. 092368549-9



Víctor Contreras P.

C.I. 093032685-5

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Yo, **NARVÁEZ NARVÁEZ KAREN STEFANY**, con documento de identificación N° **0923685499**, manifiesto mi voluntad y cedo a la **UNIVERSIDAD POLITÉCNICA SALESIANA** la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de grado titulado “**DISEÑO Y DESARROLLO DE UN PROTOTIPO DE RED DE SENSORES IOT UTILIZANDO TECNOLOGÍA LORAWAN PARA EL MONITOREO DE PARÁMETROS AMBIENTALES EN INTERIORES Y EXTERIORES.**” mismo que ha sido desarrollado para optar por el título de **INGENIERO ELECTRÓNICO CON MENCIÓN EN TELECOMUNICACIONES**, en la Universidad Politécnica Salesiana, quedando la universidad facultada para ejercer plenamente los derechos antes cedidos.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscrito este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, agosto de 2020



f) _____

Autor: Karen Narváez Narváez

Cédula: 0923685499

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Yo, **CONTRERAS PÉREZ VÍCTOR ANDRÉS**, con documento de identificación N° **0930326855**, manifiesto mi voluntad y cedo a la **UNIVERSIDAD POLITÉCNICA SALESIANA** la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de grado titulado “**DISEÑO Y DESARROLLO DE UN PROTOTIPO DE RED DE SENSORES IOT UTILIZANDO TECNOLOGÍA LORAWAN PARA EL MONITOREO DE PARÁMETROS AMBIENTALES EN INTERIORES Y EXTERIORES.**” mismo que ha sido desarrollado para optar por el título de **INGENIERO ELECTRÓNICO CON MENCIÓN EN TELECOMUNICACIONES**, en la Universidad Politécnica Salesiana, quedando la universidad facultada para ejercer plenamente los derechos antes cedidos.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscrito este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, agosto de 2020

f)  _____

Autor: Víctor Contreras Pérez

Cédula: 0930326855

**CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN
SUSCRITO POR EL TUTOR**

Yo, **BREMEN MARINO VÉLIZ NOBOA**, director del proyecto de Titulación denominado **“DISEÑO Y DESARROLLO DE UN PROTOTIPO DE RED DE SENSORES IOT UTILIZANDO TECNOLOGÍA LORAWAN PARA EL MONITOREO DE PARÁMETROS AMBIENTALES EN INTERIORES Y EXTERIORES.”** realizado por los estudiantes: Karen Narváz Narváz y Víctor Contreras Pérez, certifico que ha sido orientado y revisado durante su desarrollo, por cuanto se aprueba la presentación de este ante las autoridades pertinentes.

Guayaquil, agosto de 2020



f)

PhD. Bremnen Veliz Noboa.

DEDICATORIA

Este logro se lo dedico en primero lugar a Dios por darme fuerzas para seguir adelante y no desmayar en los problemas que se me presentaban, por ser siempre mi luz y guía.

A mis padres Ángel y Glenda, por ser un pilar fundamental en mi formación profesional, por su apoyo incondicional, consejos, valores y motivación constante que me ha permitido culminar esta etapa de mi vida.

Karen Narváez Narváez

A los que han coincidido en este tiempo y lugar, para hacer de él una extensión del amor de Dios. Mis padres sabios y dispuestos. Mi familia incondicional y afectuosa.

Víctor Contreras Pérez

AGRADECIMIENTOS

Agradezco a Dios por no permitir que renuncie a mis sueños, porque mi fe siempre se mantuvo viva y me ilumino en este arduo camino, aprendiendo de mis errores y formando mi carácter.

A mis padres por depositar toda su confianza en mí y a mi hermana que siempre nos hemos estado apoyando una a la otra.

Al tutor Ing. Bremmen Veliz, por el aporte con sus conocimientos y a todas las personas que, de alguna forma, son parte de la culminación de mi tesis.

Karen Narváez Narváez

Mis más sinceros agradecimientos a mi familia los cuales son las personas que me impulsan a cumplir mis metas, a mis padres que me han apoyado e impulsado en todo momento, a mis amigos los cuales brindaron su respaldo incondicional y mostraron su preocupación en todo momento, a mi compañera y amiga quien tuvo la voluntad de participar en este proyecto, su aporte y dedicación fueron invaluable.

Víctor Contreras Pérez

RESUMEN

AÑO	TÍTULO	ALUMNOS	DIRECTOR DE TESIS	TEMA DE TITULACIÓN
2020	Ingeniería Electrónica	Contreras Pérez Víctor Andrés Narváez Narváez Karen Stefany	PhD. Bremnen Veliz	Diseño y desarrollo de un prototipo de red de sensores iot utilizando tecnología lorawan para el monitoreo de parámetros ambientales en interiores y exteriores.

El presente proyecto de titulación “Diseño y desarrollo de un prototipo de red de sensores IoT utilizando tecnología Lorawan para el monitoreo de parámetros ambientales en interiores y exteriores” tiene como principal enfoque la adquisición a distancia de variables ambientales para darlos a conocer a usuarios en tiempo real a través de una interfaz web. Los variables a considerar son: humedad, temperatura, presión, ruido, radiación solar y calidad del aire en interiores y exteriores.

Para el desarrollo del proyecto técnico se dará uso de la tecnología LoRa-RAK 811, se empleará el entorno de desarrollo integrado Arduino MKR 1300, sensores, y raspberry PI 3, para la correcta adquisición y transferencia de datos, y a la vez siendo visualizados en una plataforma IoT.

Teniendo como resultado final una herramienta útil para ayudar al control del medioambiente.

Palabras claves: LPWAN, LoRa, Internet de las cosas, sensores, raspberry.

ABSTRACT

YEAR	DEGREE	STUDENTS	TESIS DIRECTOR	TITLE TOPIC
2020	Electronic Engineering	Contreras Pérez Víctor Andrés Narváez Narváez Karen Stefany	PhD. Bremnen Veliz	Design and development of a prototype network of IoT sensors using LoRaWAN technology for monitoring indoor and outdoor environmental parameters.

The present degree project "Design and development of a prototype network of IoT sensors using LoRaWAN technology for monitoring indoor and outdoor environmental parameters" has as its main focus is the remote acquisition of environmental variables in order to inform users in real time through an IoT platform. The variables to be considered are: humidity, temperature, noise, solar radiation and quality of the indoor and outdoor air.

For the development of the technical project, the LoRa-RAK 811 technology will be used, the Arduino MKR 1300 integrated development environment, sensors, and Raspberry Pi 3 will be used for the correct acquisition and transfer of data, and at the same time being visualized in an IoT platform.

Having as a final result a useful tool to help control the environment..

Key words: LPWAN, LoRa, Internet of things, sensors, raspberry.

ÍNDICE DE CONTENIDO

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA.....	I
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR	II
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR	III
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN SUSCRITO POR EL TUTOR	IV
DEDICATORIA	V
AGRADECIMIENTOS	VI
RESUMEN.....	VII
ABSTRACT	VIII
ÍNDICE DE CONTENIDO	IX
ÍNDICE DE IMÁGENES	XIII
ÍNDICE DE TABLAS.....	XVI
INTRODUCCIÓN	17
EL PROBLEMA	18
1.1. ANTECEDENTES	18
1.2. IMPORTANCIA Y ALCANCE	18
1.3. DELIMITACIÓN DEL PROBLEMA.	19
1.3.1. ESPACIAL.....	19
1.3.2. TEMPORAL	19
1.3.3. ACADÉMICA	19
1.4. OBJETIVOS.....	20
1.4.1. OBJETIVO GENERAL	20
1.4.2. OBJETIVOS ESPECÍFICOS.....	20
1.5. BENEFICIARIOS	20
1.6. METODOLOGÍA	20

1.6.1.	MÉTODO EXPERIMENTAL.....	21
1.6.2.	MÉTODO DEDUCTIVO.....	21
	MARCO TEÓRICO.....	22
2.1.	CONTAMINACIÓN AMBIENTAL	22
2.1.1.	TIPOS DE CONTAMINACIÓN AMBIENTAL	22
2.2.	CONTAMINACIÓN DEL AIRE	22
2.2.1.	EFFECTOS PRINCIPALES DE PM 2.5 Y 10.....	23
2.2.2.	CALIDAD DEL AIRE.....	23
2.2.3.	CALIDAD DEL AIRE EN EL ECUADOR.....	24
2.3.	CONTAMINACIÓN POR RUIDO O ACÚSTICA.....	25
2.3.1.	CONTAMINACIÓN ACÚSTICA EN EL ECUADOR.....	26
2.4.	CONTAMINACIÓN POR RADIACIÓN SOLAR	26
2.5.	CONTAMINACIÓN POR HUMEDAD.....	27
2.6.	RED DE SENSORES INALÁMBRICOS (WSN).....	27
2.6.1.	APLICACIONES DE LA RED DE SENSORES INALÁMBRICOS (WSN)	28
2.6.2.	ESTRUCTURA DE UNA RED DE SENSORES INALÁMBRICAS	29
2.6.2.1.	TOPOLOGÍA ESTRELLA	29
2.6.3.	RED DE MONITOREO DE VARIABLES AMBIENTALES	30
2.7.	SENSORES PARA EL MONITOREO AMBIENTAL.....	30
2.7.1.	SENSOR DE RAYOS UV.....	30
2.7.2.	SENSOR DE TEMPERATURA, PRESIÓN Y HUMEDAD	31
2.7.3.	SENSOR DE PARTÍCULAS DE AIRE.....	31
2.7.4.	SENSOR DE MEDICIÓN DE GAS, PRESIÓN, HUMEDAD Y TEMPERATURA	32
2.7.5.	SENSOR DE MEDICIÓN DE NIVEL DE RUIDO.....	32
2.8.	RED DE ÁREA AMPLIA DE BAJA POTENCIA (LPWAN)	33
2.8.1.	DEFINICIÓN.....	33

2.8.2.	LPWANS Y BANDAS SIN LICENCIA.....	33
2.9.	PROTOCOLO LoRaWAN.....	35
2.9.1.	MODULACIÓN LoRaWAN.....	36
2.9.2.	ARQUITECTURA LoRaWAN	36
2.9.3.	FORMATO DE LA TRAMA FÍSICA	37
2.9.4.	ELEMENTOS QUE CONFORMAN UNA RED LoRaWAN	38
2.9.5.	BANDAS DE FRECUENCIAS UTILIZADAS EN LORAWAN	38
2.9.6.	SEGURIDAD.....	40
2.9.7.	CLASES DE DISPOSITIVOS LoRaWAN	40
2.9.7.1.	DISPOSITIVO LORA RAK 831.....	41
2.9.7.1.1.	ESPECIFICACIONES	43
2.10.	INTERNET DE LAS COSAS (IoT).....	44
2.10.1.	ESCENARIOS DE INTERNET DE LAS COSAS.....	44
2.10.2.	ARQUITECTURA IOT	45
2.11.	PLATAFORMAS IOT	46
2.11.1.	THE THINGS NETWORK.....	46
2.11.2.	UBIDOTS.....	47
2.12.	ARDUINO	48
2.12.1.	ARDUINO MKR WAN 1300.....	48
2.12.1.1.	ESPECIFICACIONES.....	48
	DISEÑO E IMPLEMENTACIÓN DE LA RED DE SENSORES IOT	50
3.1.	SISTEMATIZACIÓN DEL PROTOTIPO	50
3.2.	DESPLIEGUE DE RED LORA IMPLEMENTADO.....	51
3.2.1.	NODO GATEWAY.....	52
3.2.1.1.	CONFIGURACIÓN DEL NODO GATEWAY	53
3.2.2.	NODO SENSORES.....	57
3.2.2.1.	CONFIGURACIÓN DEL NODO SENSORES	59

3.2.2.2.	REGISTRO DE NODOS SENSORES EN EL TTN	60
3.2.3.	NODO MONITOREO (APLICATIVO WEB)	63
3.2.3.1.	CONFIGURACIÓN DEL NODO MONITOREO	63
3.3.	DISEÑO DE LAS PLACAS	66
3.3.1.	NODO 1	66
3.3.2.	NODO 2, 3 Y 4.....	66
3.4.	PROTOTIPO FINAL DEL NODO 1	67
3.5.	PROTOTIPO FINAL DEL NODO 2,3 Y 4	68
	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.....	69
4.1.	INTERPRETACIÓN DE VARIABLES AMBIENTALES PUBLICADAS EN LA PLATAFORMA UBIDOTS.....	69
4.1.1.	SENSACIÓN TÉRMICA	69
4.1.2.	MATERIAL PARTICULADO	71
4.1.3.	ÍNDICE DE RADIACIÓN SOLAR UV	72
4.1.4.	RUIDO.....	73
4.2.	ANÁLISIS DE DATOS DEL MATERIAL PARTICULADO (PM 2.5 Y PM 10) 75	
4.2.1.	ANÁLISIS COMPARATIVO DE MEDICIONES.....	76
4.2.2.	ALCANCE - ANÁLISIS DE MATERIAL PARTICULADO.....	77
4.3.	DATOS OBTENIDOS EN LOS NODOS SENSORES VS ESTACIÓN METEOROLÓGICA	79
1.	BIBLIOGRAFÍA.	85
ANEXOS	92

ÍNDICE DE IMÁGENES

Imagen 1. Topología estrella [14]	29
Imagen 2. Sensor VEML6075. [16].....	31
Imagen 3. Sensor BME280. [18]	31
Imagen 4. Sensor PMS3003. [19].....	32
Imagen 5. Sensor BME680. [21]	32
Imagen 6. Sparkfun Sound Detector. [22]	33
Imagen 7. Arquitectura LoRaWAN [Autor].	37
Imagen 8. LoRa RAK 831. [32].....	42
Imagen 9. Pines de LoRa RAK 831. [32]	43
Imagen 10. Arquitectura de 3 y 5 niveles [Autor].	46
Imagen 11. Logo TTN. [40].....	47
Imagen 12. Logo Ubidots [42].....	47
Imagen 13. Arduino MKR WAN 1300. [45]	49
Imagen 14. Esquema del prototipo [Autor].	50
Imagen 15. Diagrama de flujo de Nodo Sensor – Nodo Gateway [Autor]......	51
Imagen 16. Arquitectura del hardware [Autor]......	52
Imagen 17. Módulo LoRa RAK831, Raspberry PI 3 [Autor]......	53
Imagen 18. Configuración Raspberry PI 3 para SPI. [46].	54
Imagen 19. Consola de cuenta en TTN [Autor]......	56
Imagen 20. Registro de Gateway en TTN [Autor]......	56
Imagen 21. Nodo gateway regsitrado en TTN [Autor]......	57
Imagen 22. Armado de nodo sensor Outdoor [Autor].	58
Imagen 23. Armado de nodo sensor indoor [Autor].	58
Imagen 24. DevEui del nodo 4 [Autor]......	59
Imagen 25. Programación del arduino MKR WAN 1300 [Autor]......	59
Imagen 26. División de variables por bytes [Autor]......	60
Imagen 27. Creación de aplicación para sensores indoor y outdoor [Autor]......	60
Imagen 28. Vista de la aplicación Outdoor [Autor]......	61
Imagen 29. Registro de dispositivo [Autor]......	61
Imagen 30. Credenciales de acceso del dispositivo [Autor].	62
Imagen 31. Decodificación de datos con JSON [Autor]......	62

Imagen 32. Integración con la plataforma Ubidots [Autor].....	63
Imagen 33. Configuración en TTN para integrar con Ubidots [Autor].	64
Imagen 34. Verificación de integración con Ubidots [Autor].	64
Imagen 35. Panel principal de Ubidots [Autor].	65
Imagen 36. Diseño del dashboard con widgets [Autor].....	65
Imagen 37. Diseño de Placa Nodo 1 [Autor].	66
Imagen 38. Diseño de Placa Nodo 2, 3 y 4 [Autor].	66
Imagen 39. Vista frontal y antena del nodo Gateway [Autor].	67
Imagen 40. Vista frontal y lateral de los Nodo 1 [Autor].	67
Imagen 41. Caja de pase PVC [Autor].....	68
Imagen 42. Vista frontal de los Nodo 2,3 y 4 [Autor].	68
Imagen 43. Vista lateral de los Nodo 2,3 y 4 [Autor].	68
Imagen 44. Humedad, obtenidas de Ubidots [Autor]	69
Imagen 45. Temperatura obtenida de Ubidots [Autor].....	70
Imagen 46. Valores de sensación térmica [48].	70
Imagen 47. Material particulado, nodo 1 y 2 obtenidas de Ubidots [Autor]	71
Imagen 48. Material particulado, nodo 3 y 4 obtenidas de Ubidots [Autor]	71
Imagen 49. Leyenda para el análisis de material particulado y calidad del aire [49].	71
Imagen 50. Índice de radiación solar, nodo 2, 3 obtenidas de Ubidots [Autor]	72
Imagen 51. Índice de radiación solar, nodo 4 obtenidas de Ubidots [Autor].....	72
Imagen 52. Leyenda de índice UV [50].....	73
Imagen 53. Ruido, nodo 1 y 2 obtenidas de Ubidots [Autor]	73
Imagen 54. Ruido, nodo 3 y 4 obtenidas de Ubidots [Autor]	74
Imagen 55. Equivalencias de ruido (dB) [51].	74
Imagen 56. Ubicación satelital de nodos de medición en urbanización privada [52]	75
Imagen 57. Promedio semanal de particulado PM 2.5 y PM 10 [Autor].....	76
Imagen 58. Imagen satelital de la localización del nodo sensor outdoor [52]	77
Imagen 59. Instalación de nodo sensor outdoor [Autor].....	78
Imagen 60. Registro de datos PM 10 $\mu\text{g}/\text{m}^3$ [Autor].....	78
Imagen 61. Registro de datos PM 2.5 $\mu\text{g}/\text{m}^3$ [Autor].....	78
Imagen 62. Temperatura y humedad reportado por EXA [53]	79
Imagen 63. Temperatura nodo sensor vs estación climatológica Guayaquil [Autor].	80
Imagen 64. Humedad nodo sensor vs estación climatológica Guayaquil [Autor].	80

Imagen 65. Ubicacion de nodo 1- pruebas [Autor].	92
Imagen 66. Ubicacion de nodo 2- pruebas [Autor].	92
Imagen 67. Ubicacion de nodo 3- pruebas [Autor].	93
Imagen 68. Compilación de código en nodo exterior [Autor].	93

ÍNDICE DE TABLAS.

Tabla 1 Programas de acción del plan nacional de calidad del aire del Ecuador. [6]	25
Tabla 2 Características principales de las topologías de red. [13]	29
Tabla 3 Principales características de las tecnologías lora, sigfox y iqrf. [24]	34
Tabla 4 Formato de la trama física [27]	37
Tabla 5 Frecuencia por regiones. [29]	39
Tabla 6 Canales de Frecuencias y Sub-Bandas de LoRaWAN. [29]	39
Tabla 7 Descripción de pines de salida	43
Tabla 8 Conexión de pines entre RAK811 y Raspberry PI 3. [46]	53
Tabla 9 Análisis de la sensación térmica [Autor].	70
Tabla 10 Interpretación del material particulado y calidad del aire [Autor].	72
Tabla 11 Interpretación de índice UV [Autor].	73
Tabla 12 Interpretación del nivel de ruido [Autor].	74
Tabla 13 Resumen de mediciones realizadas [Autor].	76
Tabla 14 Resumen medición variable Temperatura [Autor].	81
Tabla 15 Resumen medición variable Humedad [Autor].	82

INTRODUCCIÓN

Sin lugar a dudas los avances en las tecnologías inalámbricas están facilitando la forma de intercambiar información entre personas y por lo tanto se está extendiendo la comunicación máquina a máquina (M2M). Hoy en día la interconexión entre objetos con acceso a Internet sin necesidad de la actuación humana presenta un enorme potencial de crecimiento. Según Gartner, existe una predicción de que habrá más de 20.000 millones de dispositivos IoT para 2020 y un incremento exponencial comparado a los 2,5 millones de había en 2009. [1].

Las redes de comunicaciones inalámbricas han ido evolucionando hacia una nueva industria denominada Internet de las Cosas (IoT del acrónimo inglés), cuya idea es que todas las cosas este conectadas al internet. De momento, esta industria no compete con el sector de la telefonía móvil a nivel comercial, pero si se ha despertado un gran interés por parte de numerosas empresas, debido a las viables aplicaciones que emergen para una amplia gama de sectores, por ejemplo las ciudades inteligentes. Entre las tecnologías más utilizadas para el IoT están: SigFox, LoRa, WiFi, BLE y ZigBee. A diferencia de las tecnologías WiFi, BLE y ZigBee, la tecnología LoRa sobresale de las antes mencionadas puesto que, es utilizada para cubrir distintas necesidades de conectividad de largo alcance, bajo costo, con una amplia capacidad y bajo consumo de energía. [2].

Para el análisis del comportamiento de la tecnología LoRA, se pretende realizar el diseño e implementación de un prototipo de red de sensores IoT con el fin de monitorear en tiempo real desde una interfaz web, parámetros ambientales tales como: humedad, temperatura, ruido, presión, radiación solar y calidad del aire en interiores y exteriores. La contaminación del medioambiente no es un problema que se ha generado de un día al otro y desde las últimas décadas se ha percibido un fuerte inconveniente al tratar de solucionarlo, tanto en países desarrollados y en vías de desarrollo.

Para el desarrollo del presente proyecto se dará uso de la tecnología LoRa-RAK 811, se empleará el entorno de desarrollo integrado Arduino MKR 1300, sensores, y raspberry PI 3, para la correcta adquisición y transferencia de datos, y a la vez siendo visualizados en una plataforma IoT. Basados en el problema antes mencionado, se implementa este proyecto en la ciudad de Guayaquil.

EL PROBLEMA

1.1. ANTECEDENTES

En el Ecuador, son pocas las ciudades que cuentan con una red de monitoreo ambiental y con una interfaz web informativa, como es el caso de las ciudades de Quito y Cuenca; además, en la ciudad de Guayaquil los datos medioambientales generados por la INAMHI [Instituto Nacional de Meteorología e Hidrología] y EXA [Agencia Espacial Civil Ecuatoriana] no son gratuitos, lo que origina un obstáculo para obtener información histórica que ayude a realizar un análisis en tiempo real de los parámetros medioambientales, a fin de evitar afecciones a la salud de la población, al medio ambiente en general y al deterioro del patrimonio urbanístico de la ciudad, con el objetivo de favorecer a la comunidad, entidades estatales y estudiantes afines con la información proporcionada por dichos sensores.

1.2. IMPORTANCIA Y ALCANCE

Con la implementación de este proyecto técnico se pretende innovar e incentivar en el ámbito de la electrónica y telecomunicaciones, mediante la solución de problemas cotidianos que permitan mejorar la calidad de vida de las personas. Por ende, esta solución abarca múltiples áreas de conocimiento de ingeniería, sobre todo encaminándolo a desarrollar un sistema de monitoreo en tiempo real de los parámetros ambientales puesto que, a través de su medición se puede tener un histórico que ayude a estudios ambientales.

Por otra parte, este proyecto tendrá un impacto de índole académico y tecnológico, siendo un aporte significativo para mejorar el desarrollo del estudio y la enseñanza de las tecnologías LPWAN (Low Power Area Network), LoRa (Long Range) y el protocolo de red LoRaWAN, que son ideales para redes de IoT. Además, LoRa es de baja potencia y costo, largo alcance, con buena penetración en paredes, opera en una banda libre y se puede utilizar en lugares con poca cobertura celular o en redes privadas de sensores o actuadores. Adicionalmente, este diseño se podrá aplicar para desarrollar proyectos futuros.

1.3. DELIMITACIÓN DEL PROBLEMA.

1.3.1. ESPACIAL

El proyecto propuesto se desarrollará en diversos puntos en la ciudad de Guayaquil de la provincia del Guayas para la medición de parámetros ambientales, se contará con cuatro nodos sensores. Se destinará un nodo para el monitoreo en el interior de una edificación y tres nodos para exteriores, además de la puerta de enlace que recibirá los datos. Estas mediciones se visualizarán en una aplicación web para su monitorización.

1.3.2. TEMPORAL

La captura y medición de datos que serán considerados para la realización del proyecto propuesto se contempla por un mes dentro del periodo 2020 considerando únicamente la temática de la medición de parámetros ambientales en sectores de la ciudad de Guayaquil, el mismo que se prevé que se realice a partir del mes de marzo. La medición en tiempo real de las variables ambientales tales como, humedad, temperatura, ruido, radiación solar y calidad del aire se efectuará una vez culminado el prototipo funcional de red de sensores para despliegue en campo.

1.3.3. ACADÉMICA

El proyecto técnico planteado cumplirá con lo exigido por la Universidad Politécnica Salesiana entorno al grado social, investigativo y la estructura de presentación para proyectos de tesis; por otra parte abarcará el área de estudio de las Telecomunicaciones, se sustentará en actas de conferencia, libros, documentos de sitios webs e informes que proporcionarán conceptos sobre la tecnología LoRaWAN y sensores inalámbricos.

1.4. OBJETIVOS.

1.4.1. OBJETIVO GENERAL

Diseñar y desarrollar un prototipo de red sensores IoT utilizando tecnología LoRaWAN ubicados en puntos estratégicos de la ciudad de Guayaquil para el monitoreo en tiempo real desde una interfaz web de los parámetros de humedad, temperatura, ruido, radiación solar y calidad del aire en interiores y exteriores.

1.4.2. OBJETIVOS ESPECÍFICOS

- Diseñar e implementar la red de sensores inalámbricos auto sustentable energéticamente, tal como sensores de humedad, temperatura, ruido, radiación solar y calidad del aire.
- Implementar una puerta de enlace para que sean enrutados hacia el servidor los parámetros adquiridos de la red de sensores.
- Diseñar un interfaz web informativo con una estructura metodológica amigable al usuario.

1.5. BENEFICIARIOS

Como beneficiarios principales de este proyecto técnico serán los estudiantes de la carrera de ingeniería electrónica y telecomunicaciones de la sede Guayaquil que estén cursando las materias de redes inalámbricas, antenas, medios de transmisión y redes de computadoras.

Como beneficiarios secundarios tenemos a la comunidad de la ciudad de Guayaquil ya que tendrán acceso a la información obtenida en tiempo real, la misma que contribuirá al desarrollo de futuros proyectos en el control y manejo de los recursos, teniendo así un aporte ambiental a la ciudad.

1.6. METODOLOGÍA

Las metodologías que se utilizarán para el desarrollo de este trabajo de titulación son el método experimental y deductivo.

- Se diseñarán nodos sensores con la capacidad de recolectar parámetros ambientales: humedad, temperatura, ruido, radiación solar y calidad del aire; los cuáles interactuarán por diversos protocolos de comunicación con la plataforma Arduino MKR WAN 1300.
- Los nodos sensores serán programados con protocolos de mensajería, diseñados para IoT. Por consiguiente, los nodos por cada periodo de tiempo enviarán información con los parámetros propuestos vía inalámbrica a través de la puerta de enlace a la nube (cloud).
- La puerta de enlace (gateway) será diseñada utilizando un Raspberry pi 3 B+ y un módulo gateway LoRaWAN RAK 831, el cual realizará operaciones de coordinación y transporte además de administración mediante una plataforma opensource.
- Se escogerá un punto en cada locación que nos permita tener línea de vista para levantar el enlace entre el gateway y los nodos sensores.
- La recopilación y presentación de datos se dará mediante el desarrollo de una interfaz la cual realizará sus procesos dentro del Cloud Computing que nos permitirá el servidor en la nube escogido para operaciones de IoT.

1.6.1. MÉTODO EXPERIMENTAL

Durante la formación académica, el estudiante de ingeniería electrónica obtiene la habilidad y los conocimientos necesarios que se aplican en el campo real para poder desarrollar cualquier aplicación a su rama de estudio y por medio de este método se realizarán pruebas experimentales en las ubicaciones que tendrán los nodos en la ciudad.

1.6.2. MÉTODO DEDUCTIVO

Por medio de este método se puede realizar un proceso sintético-analítico, donde se deducirá la información obtenida por los sensores y la cual podrá ser visualizada mediante un interfaz web.

MARCO TEÓRICO

2.1. CONTAMINACIÓN AMBIENTAL

La contaminación ambiental siempre ha existido y es una consecuencia producida por diferentes actividades que realiza el hombre, por lo que conlleva a muchas repercusiones en la integridad física del ambiente. Las actividades de desarrollo, como la construcción, el transporte y la fabricación, no solo agotan los recursos naturales, sino que también producen una gran cantidad de desechos que conducen a la contaminación del aire, el agua, el suelo y los océanos; calentamiento global y lluvias ácidas. En esta sección, se estudia las principales causas de la contaminación y sus efectos en nuestro medio ambiente. [3].

2.1.1. TIPOS DE CONTAMINACIÓN AMBIENTAL

La contaminación puede surgir de los siguientes tipos:

- Contaminación del aire
- Contaminación del ruido o acústica
- Contaminación térmica
- Contaminación por radiación
- Contaminación por humedad

2.2. CONTAMINACIÓN DEL AIRE

La contaminación atmosférica es el principal riesgo ambiental para la salud en las Américas. De acuerdo con la Organización Mundial de la Salud (OMS), aproximadamente 7 millones de personas por año mueren por enfermedades relacionadas con la contaminación atmosférica.

Los contaminantes atmosféricos más relevantes para la salud son material particulado (PM) con un diámetro de 10 micras o menos, que pueden penetrar profundamente en los pulmones e inducir la reacción de la superficie y las células de defensa. La mayoría de estos contaminantes son el producto de la quema de combustibles fósiles, pero su composición puede variar según sus fuentes. Las directrices de la OMS sobre la calidad del aire recomiendan una exposición máxima de 20 $\mu\text{g}/\text{m}^3$ para las PM10 y una exposición máxima de 10 $\mu\text{g}/\text{m}^3$ para las PM2.5, basado en las evidencias de los efectos sobre la salud de la exposición a la contaminación del aire ambiente. [4].

2.2.1. EFECTOS PRINCIPALES DE PM 2.5 Y 10

- Salud humana: produce irritación en las vías respiratorias. Su acumulación en los pulmones origina enfermedades como la silicosis y la asbestosis. Agravan el asma y las enfermedades cardiovasculares.
- Materiales: deteriora los materiales de construcción y otras superficies.
- Vegetación: interfiere en la fotosíntesis. Medio ambiente: disminuyen la visibilidad y provocan la formación de nubes.

2.2.2. CALIDAD DEL AIRE

Es una táctica para establecer las condiciones del aire desde un grado de pureza hasta una calidad crítica perjudicial para el deterioro de la salud humana. Es decir, hace referencia a la cantidad de contaminantes presentes en el aire, ya sea de una alta calidad, con un nivel bajo de concentración; o una mala calidad, con un nivel elevado de concentración de contaminantes del aire. La contaminación del aire se considera un tema de gran importancia puesto que, es uno de los principales problemas en las zonas urbanas del mundo, tanto en los países desarrollados como en aquellos que están en vías de desarrollo. Es un recurso importante contar con una red de monitoreo de la calidad del aire, para la obtención de información de los principales polutantes atmosféricos a fin de realizar los respectivos análisis.

La OMS reconoce que la contaminación del aire es un factor de riesgo crítico para las enfermedades no transmisibles (ENT), ya que se estima que causa una cuarta parte

(24%) de todas las muertes de adultos por cardiopatías, el 25% de las muertes por accidentes cerebrovasculares, el 43% de las muertes por neumopatía obstructiva crónica y el 29% de las muertes por cáncer de pulmón. [5].

2.2.3. CALIDAD DEL AIRE EN EL ECUADOR

Hoy en día los problemas relacionados con la calidad del aire en el Ecuador se presentan en las más grandes ciudades, en especial en Quito, Guayaquil y Cuenca, donde se evidencia gran concentración de población urbana, la presencia de industrias manufactureras, la utilización de tecnologías obsoletas en las actividades productivas y de transporte, mala calidad de los combustibles, explotaciones mineras a cielo abierto, entre otras. [6].

En el país se cuenta con pocas investigaciones sobre los efectos de la contaminación del aire en la salud de las personas, estos temas no han sido incluidos en los programas de desarrollo urbano y no se han llevado a cabo estudios epidemiológicos relacionados con la contaminación del aire. En los últimos años, la Fundación Natura a través de su proyecto “Calidad del Aire” ha efectuado algunos estudios muy puntuales sobre los efectos a la salud provocados por la contaminación atmosférica solamente en la ciudad de Quito. [6].

De acuerdo con la Organización Mundial de la Salud, el aire limpio es un derecho inalienable de la población para su salud y bienestar, por lo cual es indispensable que Ecuador incorpore las normas de prevención y control de contaminación atmosférica propuestos en el Plan Nacional de la Calidad del Aire.

El Ministerio del Ambiente del estado ecuatoriano, ha desarrollado tres programas en beneficio del medio ambiente, los cuales tienen como finalidad el control y vigilancia de la calidad del aire, mejoramiento y prevención de su deterioro y además las medidas a tomar en cuenta en estado de alerta. En la Tabla 1 se detalla los programas de acción con sus respectivos proyectos que cuenta el Plan Nacional de Calidad del Aire del Ecuador.

Tabla 1 Programas de acción del plan nacional de calidad del aire del Ecuador. [6]

PROGRAMAS	PROYECTOS
<p>PROGRAMA 1: CONTROL Y VIGILANCIA DE LA CALIDAD DEL AIRE.</p>	Proyecto 1: Desarrollo del inventario nacional de emisiones.
	Proyecto 2: Sistema nacional de monitoreo y vigilancia de la calidad de aire.
	Proyecto 3: Elaboración de un sistema de información de la calidad del aire (sica).
	Proyecto 4: Implantación del sistema nacional de vigilancia epidemiológica de las enfermedades respiratorias generadas por contaminación del aire.
	Proyecto 5: Modelos predictivos de calidad del aire.
<p>PROGRAMA 2: MEJORAMIENTO DE LA CALIDAD DEL AIRE Y PREVENCIÓN DE SU DETERIORO.</p>	Proyecto 6: Competencias sobre la gestión del recurso aire.
	Proyecto 7: Formación, capacitación e investigación en gestión de la calidad del aire integrados a las políticas nacionales de ciencia y tecnología.
	Proyecto 8: Programa nacional de reducción de emisiones.
	Proyecto 9: Producción de combustibles de mejor calidad para fuentes móviles y fijas.
	Proyecto 10: Sistema nacional de revisión técnica vehicular.
<p>PROGRAMA 3: MEDIDAS A SER APLICADAS DURANTE LOS ESTADOS DE ALERTA.</p>	Proyecto 11: Sistema de participación ciudadana en tema calidad del aire.
<p>PROGRAMA 3: MEDIDAS A SER APLICADAS DURANTE LOS ESTADOS DE ALERTA.</p>	Proyecto 12: Implementación de planes de contingencia ante episodios críticos de contaminación del aire.

2.3. CONTAMINACIÓN POR RUIDO O ACÚSTICA

Se denomina contaminación acústica al exceso de sonido que altera las condiciones normales del ambiente. Si bien es cierto, el ruido no se acumula como otras contaminaciones. Dicha contaminación es una de las mayores preocupaciones en las áreas urbanas puesto que, puede causar grandes daños en la calidad de vida de las personas si la contaminación por ruido sobrepasa los niveles tolerables de entre 40 a 50 decibeles (dB), según la Organización Mundial de la Salud (OMS). Desde 60 dB se pueden generar malestares. [7].

2.3.1. CONTAMINACIÓN ACÚSTICA EN EL ECUADOR

Ministerio de Ambiente del Ecuador (MAE) realiza el control y seguimiento del ruido a partir de la verificación del cumplimiento del Plan de Manejo Ambiental con el que cuentan los distintos proyectos y actividades. Esto se lo realiza a través de auditorías ambientales y reportes de monitoreo ambiental, que son ejecutados con metodología estandarizada y mediante laboratorios acreditados.

Los sonidos con mayor intensidad provienen de diversas fuentes como: industrias, tráfico de vehículos, construcciones y eventos deportivos/recreativos, entre otras. En ese sentido, el MAE ha expedido la normativa sobre ‘Límites permisibles de niveles de ruido ambiente para fuentes fijas, fuentes móviles, y vibraciones’ y una ‘Norma de Ruido de Aeropuertos’, que se presenta en el Anexo 5 y Anexo 9 del Libro VI del Texto Unificado de Legislación Secundaria del Ministerio del Ambiente (TULSMA). En el documento se establecen los límites máximos permisibles de ruido para fuentes fijas como por ejemplo industrias y recintos aeroportuarios, y fuentes móviles como vehículos.

Asimismo, el MAE trabaja en proyectos de Educación Ambiental dirigidos a toda la población ecuatoriana, con el objetivo de formar una nueva Generación Verde que contribuya en el trabajo de conservar los ecosistemas. [8]

2.4. CONTAMINACIÓN POR RADIACIÓN SOLAR

La contaminación por radiación es el aumento de la radiación de fondo natural. Hay muchas fuentes de contaminación por radiación, como los desechos nucleares de las centrales nucleares, la minería y el procesamiento de material nuclear, etc. El peor caso de la contaminación nuclear fue el desastre de Chernóbil en Rusia que ocurrió en 1986, pero los efectos aún más en la actualidad. [9].

2.5. CONTAMINACIÓN POR HUMEDAD

La Organización Mundial de la Salud (OMS) para la protección de la salud pública frente a los riesgos debido a la humedad en interiores, hacen referencia a que los problemas de calidad del aire en el interior son reconocidos como importantes factores de riesgo para la salud, además de que las personas pasan una parte importante del tiempo dentro de los edificios. La contaminación microbiana que surge por la humedad incluye cientos de especies de bacterias y hongos que crecen en el interior cuando se dispone de suficiente humedad. La exposición a los contaminantes microbianos se asocia clínicamente con síntomas respiratorios, alergias, asma y reacciones inmunológicas. Por ende, se concluye que la presencia de muchos agentes biológicos en el ambiente interior se debe a la humedad y a una ventilación inadecuada. [10].

2.6. RED DE SENSORES INALÁMBRICOS (WSN)

Las redes de sensores inalámbricas (Wireless Sensor Networks, o simplemente WSN), constituyen actualmente una herramienta tecnológica muy valiosa para recolectar información del entorno. Estas redes están formadas por plataformas de hardware con recursos limitados de memoria, cómputo y energía, motivo por el cual las WSN trabajan con protocolos especialmente diseñados para funcionar sobre estas plataformas (LoRa, ZigBee, etc.). Estos protocolos de WSN, generalmente, no son compatibles con los protocolos de la pila TCP/IP. Esta incompatibilidad implica que para presentar los datos recolectados por las WSN al usuario mediante redes TCP/IP (p. ej. Internet), deban implementarse distintas técnicas de hardware y software. [11].

Las WSN, permiten obtener y transmitir información del entorno donde están desplegadas. Estas redes pueden ser utilizadas en aplicaciones tales como agricultura de precisión, meteorología y monitoreo del medio ambiente, cuidado y monitoreo de pacientes, prevención de inundaciones, etc. Una red de sensores inalámbrica está formada por nodos, routers, y una estación base. Los datos recolectados por las WSN pueden ser luego puestos a disposición de usuarios a través de alguna red. [11].

2.6.1. APLICACIONES DE LA RED DE SENSORES INALÁMBRICOS (WSN)

Las WSN han tenido un rápido desarrollo en los últimos años y tiene gran popularidad en el mercado, se han aplicado con éxito en varios dominios de aplicación, tales como [3], [12]:

✓ **Aplicaciones militares:**

- Monitorización de fuerzas y equipos enemigos
- Vigilancia en el campo de batalla
- Reconocimiento del terreno
- Detección de ataques biológicos, químicos o nucleares, etc.

✓ **Aplicaciones medioambientales:**

- Calidad del aire
- Calidad del agua
- Monitorización de las condiciones ambientales en cultivos
- Agricultura de precisión
- Detección de incendios forestales
- Detección de inundaciones
- Estudios de contaminación
- Prevención de desastres
- Monitorización de áreas afectadas por desastres, etc.
- Estudios sísmicos
- Seguridad de estructuras

✓ **Aplicaciones médicas:**

- Telemonitorización de datos fisiológicos en pacientes
- Diagnóstico
- Administración de medicamentos

- Seguimiento de médicos y pacientes en hospitales, etc.
- Aplicaciones en el hogar/edificios

2.6.2. ESTRUCTURA DE UNA RED DE SENSORES INALÁMBRICAS

Dentro de la estructura de una red de sensores inalámbricos se analizan diferentes topologías para redes de comunicaciones de radio frecuencia.

Tabla 2 Características principales de las topologías de red. [13]

Parámetros a tener en cuenta	Malla	Árbol	Estrella
Escalabilidad	Sí	Sí	No
Eficiencia energética	No	Sí	Sí
Redundancia de rutas	Sí	No	No
Movilidad de nodos	Sí	Parcial	Parcial

2.6.2.1. TOPOLOGÍA ESTRELLA

En una topología estrella activa cada nodo se conecta a un nodo central que normalmente tiene los medios para prevenir problemas relacionados con el eco y es el encargado del control de acceso a la red por el resto de los nodos. En esta topología adquiere una importancia decisiva el nodo central que se encarga de controlar toda la comunicación, pues cualquier perturbación en el mismo conduce, generalmente, al fallo de la red completa. [13]

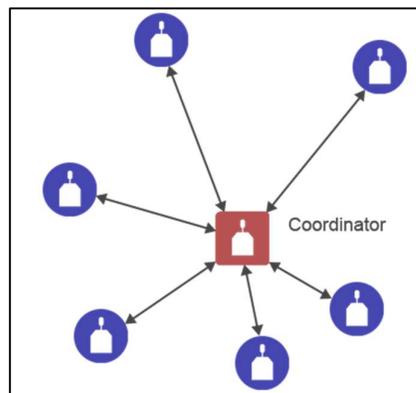


Imagen 1. Topología estrella [14]

2.6.3. RED DE MONITOREO DE VARIABLES AMBIENTALES

Para el control ambiental es importante contar con sensores aptos para identificar contaminantes ambientales y a su vez, el control de parámetros tales como: humedad, temperatura, ruido, radiación solar, calidad del aire, etc.

Tomando como ejemplo al monitoreo de la calidad del aire como una acción de gran importancia, para restringir la suspensión de compuestos tóxicos en el ambiente cuya presencia contribuya a enfermedades. Sobre todo, calidad del aire es un factor que mide el contenido de una concentración de componentes presentes en el aire, el cual debe cumplir con normas que beneficien la salud y el bienestar de la población y mantener el equilibrio ecológico. [9].

2.7. SENSORES PARA EL MONITOREO AMBIENTAL

Las tareas de los sensores para el monitoreo ambiental abarcan una amplia gama de actividades, ya que la contaminación del medio ambiente ocurre no solo por la descarga de desperdicios de agua, la tierra y el aire, sino también por la generación de ruidos en el audio. En los últimos años, se han desarrollado diversos sistemas sensores que cubren todas estas aplicaciones, los cuales tienen como objetivo supervisar, registrar y organizar los datos recogidos enviados a una ubicación central de forma autónoma. [15]

2.7.1. SENSOR DE RAYOS UV

Es un pequeño sensor que tiene la posibilidad de detectar luz ultravioleta (UV) en cualquier proyecto con microcontrolador. El VEML6075 de Vishay tiene un sensor de luz UV y un ADC I2C integrado que se encarga de realizar lecturas e integrarlas en aprox. 60 ms a 500 ms.

Este sensor UV funciona muy bien con alimentación de 3 o 5V. Es compacto y es fácil de usar con cualquier microcontrolador compatible con interface I2C. Tiene un rango de medición UVA: $365\pm 10\text{nm}$ y UVB: $330\pm 10\text{nm}$



Imagen 2. Sensor VEML6075. [16]

2.7.2. SENSOR DE TEMPERATURA, PRESIÓN Y HUMEDAD

El sensor BME280 en un solo encapsulado es capaz de medir temperatura, humedad y presión atmosférica. Se puede comunicar mediante el bus I2C o SPI, así que es compatible con prácticamente cualquier plataforma. Las características en cuanto a medición de temperatura y presión atmosférica tienen un rango de temperatura de -40 a $+85$ °C con una precisión de ± 1 °C y resolución $0,01$ °C, y para presión 300 - 1100 hPa, precisión de ± 1 Pa, y resolución de $0,18$ Pa.

Respecto a su comportamiento como higrómetro, el BME280 tiene un rango de medición de humedad relativa de 0 a 100% , con una precisión de $\pm 3\%$ Pa y una resolución de 0.008% . El sensor puede alimentarse tanto con $5V$ como $3.3V$. [17]



Imagen 3. Sensor BME280. [18]

2.7.3. SENSOR DE PARTÍCULAS DE AIRE

PMS3003 es un tipo de sensor de concentración de partículas digital y universal, que se puede utilizar para obtener el número de partículas suspendidas en el aire entre los $0,3$ y 10 μm , es decir, la concentración de partículas, y salir en forma de interfaz digital. Este sensor se puede insertar en instrumentos variables relacionados con la concentración de partículas suspendidas en el aire u otros equipos de mejora ambiental para proporcionar los datos de concentración correctos a tiempo. Tiene un tiempo de respuesta menos de 10 segundos.



Imagen 4. Sensor PMS3003. [19]

2.7.4. SENSOR DE MEDICIÓN DE GAS, PRESIÓN, HUMEDAD Y TEMPERATURA

El sensor BME680 puede medir la humedad con una precisión de $\pm 3\%$, la presión barométrica con una precisión absoluta de ± 1 hPa y la temperatura con una precisión de $\pm 1,0$ ° C. Además, el BME680 contiene un sensor MOX. El óxido metálico calentado cambia de resistencia en base a los VOC del aire, por lo que se puede utilizar para detectar gases y alcoholes como etanol, alcohol y monóxido de carbono, y realizar mediciones de calidad del aire. El BME680 minimiza el consumo medio de energía al combinar un modo activo y eficiente, denominado modo forzado, y un modo de reposo de bajo consumo, que solo utiliza $0,15$ μ A. En modo forzado, el sensor realiza un ciclo en el que mide secuencialmente temperatura, presión, humedad y gas antes de volver al modo de reposo automáticamente.

Por esto, el rápido tiempo de respuesta del sensor reduce el tiempo de ciclo en modo forzado, las mediciones pueden activarse o desactivarse individualmente para ahorrar energía. [20]



Imagen 5. Sensor BME680. [21]

2.7.5. SENSOR DE MEDICIÓN DE NIVEL DE RUIDO

El sensor SparkFun Sound Detector, es una placa de detección de audio pequeña y muy fácil de usar con tres salidas diferentes. El detector de sonido no solo proporciona

una salida de audio, sino también una indicación binaria de la presencia de sonido y una representación analógica de su amplitud. El rango de medición en cuanto a la ganancia es de 50 a 80 dB y un tiempo de respuesta de 0.1 segundo

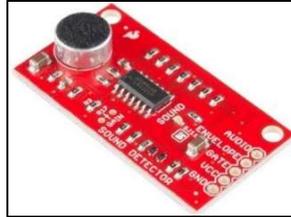


Imagen 6. Sparkfun Sound Detector. [22]

2.8. RED DE ÁREA AMPLIA DE BAJA POTENCIA (LPWAN)

2.8.1. DEFINICIÓN

Las siglas en inglés LPWAN está formado a partir de dos fases: Low Power (consumo de baja potencia) y Wide Area Network (comunicación de largo alcance), el mismo que describen características intrínsecas, que incluyen por ejemplo bajo costo y baja velocidad de datos, además de que puede actuar en el espectro con licencia o sin licencia. LPWAN se convierte así, en una tecnología para comunicaciones inalámbricas y abarca una gama de tecnología que se utilizan para conectar dispositivos de baja potencia, como sensores y controladores en comunicaciones IoT y máquina a máquina (M2M).

Los arquitectos de redes han utilizado tecnologías LPWAN para diseñar redes de sensores inalámbricos a medida y personalizar el costo, la cobertura y el consumo de energía de la transmisión para aplicaciones específicas de IoT y M2M. [23].

2.8.2. LPWANS Y BANDAS SIN LICENCIA

Las principales tecnologías mencionadas a continuación funcionan en bandas sin licencia definidas por el Instituto Europeo de Normas de Telecomunicaciones (ETSI) y la Unión Internacional de Telecomunicaciones (UIT), que están representadas, por

ejemplo, por bandas industriales, científicas y médicas (ISM). No hay pago por el uso de la banda ISM. [24].

Las especificaciones de las tecnologías con sus principales características están resumidas en la siguiente Tabla 3.

Tabla 3 Principales características de las tecnologías lora, sigfox y iqrf. [24]

	LoRa	Sigfox	IQRF
Bandas de frecuencia	868 MHz (Europe) 915 MHz (N. America) 433 MHz (Asia)	868 MHz (Europe) 915 MHz (USA)	868 MHz (Europe) 915 MHz (N. America) 433 MHz (Asia)
Rango máximo	15 - 20 km	30 - 50 km (countryside) 3 - 10 km (urban)	Hundreds meters
Tasa de datos	0.25 - 50 kbit/s	100 bit/s	1.2 - 115 kbit/s Typical 20 kbit/s
Topología	STAR	STAR	MESH
Modulación	FSK s FEC	UNB (DBPSK)	FSK
Carga útil	2 - 255 B	Max 12 B	Max 128 B
Potencia de transmitida	10 – 18 dBm	0 – 14 dBm	Optional Max 11 dBm

➤ **Tecnología LoRa**

Fue creado e impulsado por la asociación LoRa® Alliance, que se ocupa de la comunicación IoT, M2M, Smart cities, etc. Los principales productores de dispositivos LoRa son las empresas Semtech y Microchip. Es una red inalámbrica que utiliza técnicas de capa física de secuencia directa de espectro ensanchado (DSSS). Permite comunicarse en señales con un valor bajo de potencia y utiliza modulación por desplazamiento de frecuencia (FSK).

En el caso de este proyecto, es la tecnología que se estudiará y profundizará a mayor detalle.

➤ **Tecnología Sigfox**

La tecnología Sigfox es desarrollada por un grupo de Francia con el mismo nombre. Sigfox utiliza en la capa física una técnica de Ultra Narrow Band (UNB), además del sistema LoRa y la técnica DSSS. Los dispositivos Sigfox también permiten demodular la señal con una pequeña potencia recibida, la sensibilidad es de hasta -142 dBm. La red Sigfox se está configurando en la topología en estrella, lo que reduce los gastos de construcción y otros.

La red funciona en las bandas de frecuencia globales sin licencia, como 868 MHz para Europa, 902 MHz para EE. UU. El espectro de Sigfox en 868 MHz se divide en 400 canales desde 868.18 hasta 868.22 MHz.

➤ **Tecnología IQRF**

La tecnología IQRF funciona con transceptores de radio (TR) con un microcontrolador (MCU), que tiene su propio sistema operativo. Todo el sistema implementa la topología de malla en lugar de la topología en estrella, que es la mayor diferencia con otras tecnologías LPWAN como LoRa y Sigfox. Utiliza el propio protocolo IQMESH para proporcionar más fácilmente todas las ventajas de la topología de malla.

La tecnología trabaja en las bandas de frecuencia sin licencia de 868, 916 y 433 MHz. Utiliza una cantidad específica de canales en estas bandas, que es diferente para cada banda. En la banda de 868 MHz se definen 62 canales fijos, que están en un rango de 863.15 a 869.25 MHz. Cada canal tiene un ancho de banda de 100 kHz.

2.9. PROTOCOLO LoRaWAN

LoRaWAN es la abreviación de LoRa Wide Area Network, la misma que proporciona un mecanismo de control de acceso al medio a la red LoRa y permite que muchos dispositivos finales se comuniquen por medio de una puerta de enlace (Gateway) utilizando la modulación LoRa. Si bien se conoce que la modulación LoRa es patentada, LoRaWAN es un estándar abierto impulsado por LoRa Alliance. [25].

El protocolo LoRaWAN está diseñado para conectar de forma inalámbrica cosas que funcionan con baterías a Internet en redes regionales, nacionales o globales. Es una de las pocas redes IoT que implementan cifrado de extremo a extremo. La seguridad LoRaWAN utiliza la primitiva criptográfica AES combinada con varios modos de operación: CMAC para protección de integridad y CTR para cifrado. Cada dispositivo LoRaWAN está personalizado con una clave AES única de 128 bits (llamada AppKey) y un identificador único global (DevEUI basado en EUI-64), que se utilizan durante el proceso de autenticación del dispositivo

2.9.1. MODULACIÓN LoRaWAN

La modulación de LoRaWan es LoRa la misma que utiliza como esquema de modulación de señal, una variación de modulación DSSS de tipo Spread Spectrum denominada Chirp Spread Spectrum (CSS). Permite conexiones de bajo costo, bajo consumo (25 mA en transmisión y 10 mA en recepción) robustas frente a interferencias y resistentes al efecto Doppler. Además de alta sensibilidad para recibir datos (-168dB), largo alcance de hasta 10 km máximo y baja transferencia de datos (hasta 255 bytes). [26].

2.9.2. ARQUITECTURA LoRaWAN

La arquitectura básica de una red LoRaWAN es la siguiente: los dispositivos finales se comunican con las puertas de enlace (Gateway) usando tecnología LoRa a través de la red LoRaWAN. Las puertas de enlace envían tramas LoRaWAN sin procesar de los dispositivos a un servidor de red a través de una interfaz de retorno con un rendimiento superior, generalmente Ethernet o 3G.

En consecuencia, las puertas de enlace son solo relés bidireccionales o conversores de protocolos, siendo el servidor de red responsable de decodificar los paquetes enviados por los dispositivos y generar los paquetes que deben enviarse de vuelta a los dispositivos. Hay tres clases de dispositivos finales LoRa, que difieren solo con respecto a la programación del enlace descendente. [25].

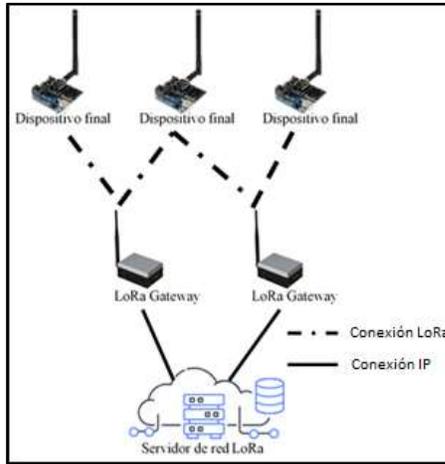


Imagen 7. Arquitectura LoRaWAN [Autor].

2.9.3. FORMATO DE LA TRAMA FÍSICA

Tabla 4 Formato de la trama física [27]

Preámbulo	FHDR	PHDR_CRC	PHYPayload	CRC
-----------	------	----------	------------	-----

En la Tabla 4 se visualiza el formato de la trama física de LoRa, la misma que comienza con un preámbulo de sincronización, el cual define el esquema de modulación del paquete y define parámetros de modulación de forma individualizada para cada paquete. Esto permite indicar el esquema de transmisión para cada paquete en función de la distancia a la pasarela (el objeto puede estar en movimiento) y por lo tanto ajustar el rendimiento y consumo energético según las circunstancias. El preámbulo finaliza con un byte de sincronización que permite diferenciar entre las redes LoRa que están emitiendo en la misma banda de frecuencias de forma que un dispositivo sólo escucha las emisiones que tienen su mismo byte de sincronización, por consiguiente, permite la concurrencia de distintas redes LoRa en un mismo espacio.

El tamaño máximo del payload es de 255 bytes (dicho tamaño está definido por un solo byte en la cabecera de la trama) suficiente para la transmisión de datos de geolocalización, estado de dispositivo, información de sensores, etc. El campo PHYpayload contiene la trama de la capa de acceso al medio LoRaWAN. [26].

2.9.4. ELEMENTOS QUE CONFORMAN UNA RED LoRaWAN

- Dispositivo o mota: Instrumento de comunicación utilizado para realizar la medición de información en el lugar donde se implante, destaca por un uso muy bajo de potencia.
- Red: Camino de comunicación que permite el envío de mensajes que son recibidos por los nodos y enviados a la aplicación.
- Gateway: Dispositivo capaz de enviar y recibir información a los diferentes nodos. Puerta de enlace que establece entre nodos y dispositivos, para su posterior retransmisión a Internet y poder visualización de información desde cualquier lugar.
- Aplicación: Dispositivo capaz de gestionar un software que pueda interpretar los datos y pueda hacer lectura información recibida por la Gateway.
- Mensaje de enlace ascendente: Aquel envío de información desde un determinado dispositivo a la aplicación.
- Mensaje de enlace descendente: Es aquel envío de información desde una determinada aplicación a un dispositivo. [28]

2.9.5. BANDAS DE FRECUENCIAS UTILIZADAS EN LORAWAN

LoRaWAN utiliza un espectro de frecuencia de radio de libre utilización, es decir que cualquiera puede utilizar este espectro de frecuencia sin tener que pagar por su utilización. Este tipo de frecuencias es muy similar a las frecuencias 2.4 GHz y 5 GHz en todo el mundo, donde el uso de estas licencias es de libre utilización sin necesidad de licencia ni de permiso para la transmisión de información.

El uso de frecuencias de bajo espectro hace que el alcance de radio sea mucho mayor, pero hay que prestar atención al hecho del uso de frecuencias libres en función del país

donde nos encontremos, ya que, existen normativas de uso de espectro de frecuencias exclusivas en cada continente en todo el mundo. [28].

LoRaWAN utiliza 9 canales de frecuencia para la transmisión de datos, estos están distribuidos para 7 regiones. Tomando en cuenta en el lugar donde se realizó esta investigación, utilizamos la frecuencia US915 que corresponde a la región de Norte América Tabla 5, consta de 64 canales de enlace ascendente disponibles (125 kHz cada uno) (canales 0-63) a partir de 902.3 MHz que se incrementan cada 200 kHz hasta 914.9 MHz, hay 8 canales de enlace ascendente adicionales (500 kHz cada uno) (canales 64-71) desde 903 MHz que aumentan cada 1,6 MHz hasta 914,2 MHz y para la comunicación de puerta de enlace a dispositivo del nodo final hay 8 canales de enlace descendente (500 kHz cada uno) (canales 0-7) desde 923.3 MHz que se incrementan cada 600 kHz hasta 927.5 MHz. [29].

Tabla 5 Frecuencia por regiones. [29]

Región	Frecuencia (Mhz)
Europa	EU433, EU868
China	CN470
Norte América	US915
Asia	AS920, AS923
Australia	AU915
Korea	KR920
India	IN865

Tabla 6 Canales de Frecuencias y Sub-Bandas de LoRaWAN. [29]

Canales, frecuencias y sub-bandas de LoRaWAN		
Bandas de frecuencia	Rango de frecuencia (MHz)	Canales
Banda total de EE. UU.	902.3 - 914.9	0-63
Sub-bandas de enlace ascendente	Rango de frecuencia (MHz)	Canales
Sub-banda 1	902.3 - 903.7	0-7
Sub-banda 2	903.9 - 905.3	8-15
Sub-banda 3	905.5 - 906.9	16-23
Sub-banda 4	907.1 - 908.5	24-31
Sub-banda 5	908.7 - 910.1	32-39
Sub-banda 6	910.3 - 911.7	40-47
Sub-banda 7	911.9 - 913.3	48-55
Sub-banda 8	915.5 - 914.9	56-63
Sub-bandas de enlace descendente	Rango de frecuencia (MHz)	Canales
Sub-banda de enlace descendente	903 - 914.2	64-71

2.9.6. SEGURIDAD

La seguridad de LoRaWAN facilita el cifrado y firma de paquetes enviados a través de la red. Para el cifrado, se utilizan claves simétricas conocidas por los nodos finales, los servidores de red y de aplicaciones, que se distribuyen de dos maneras diferentes, dependiendo del método de activación empleado, a saber, la activación por aire (OTAA) y la activación por personalización (ABP).

Ambos métodos de activación proporcionan un alto nivel de seguridad mediante el cifrado simétrico en el intercambio de mensajes entre los nodos finales y los servidores. Los mecanismos proporcionados eliminan cualquier posibilidad de que un atacante inyecte nodos finales maliciosos o aproveche la red en el proceso de activación sin conocer las claves del nodo final. LoRaWAN proporciona varias capas de cifrado, que hacen uso del algoritmo de cifrado AES-128 para proteger las comunicaciones de datos:

- **Network Session Key:** Clave de 128 bits que garantiza seguridad a nivel de red.
- **Application Session Key:** Clave de 128 bits que garantiza seguridad extremo-extremo.
- **Application Key:** Clave de 128 bits que se utiliza para despliegues OTAA. [30].

2.9.7. CLASES DE DISPOSITIVOS LoRaWAN

Aunque la capa física LoRa está dedicada, las otras capas en LoRaWAN están abiertas y han sido descritas por la alianza LoRa. La topología de LoRaWAN es estrella, incluidas las puertas de enlace y los dispositivos finales. Los dispositivos finales comunican datos con las puertas de enlace mediante un enlace inalámbrico basado en la modulación LoRa y las puertas de enlace envían, reciben datos desde servidores a través de un enlace de banda ancha confiable y de alto rendimiento. Los dispositivos LoRa pueden actuar en una de las siguientes tres clases:

- **Dispositivos finales bidireccionales (clase A):** Permiten una comunicación bidireccional con rigurosos requerimientos de uso y eficiencia de energía, es decir que reciben paquetes de datos siempre y cuando se haya enviado un paquete antes. La razón de este proceso es que cada vez que un dispositivo envía un dato, se abren dos puertas de recepción, la primera puerta un segundo después de la transmisión de un dato y la segunda un segundo después de la primera puerta, con la oportunidad de recibir un paquete de vuelta.
Los dispositivos de Clase A pueden utilizarse de dos formas:
 - Enviar datos en un intervalo de tiempo (por ejemplo, cada 10 minutos)
 - O enviar datos con eventos (p. ej. cuando la temperatura supera los 16° o por debajo de 19°).

- **Dispositivos finales bidireccionales con ranuras de recepción programadas (Clase B):** Estos dispositivos añaden la capacidad de recibir datos sin necesidad de enviar un paquete antes. Los dispositivos de Clase B abren puertas de recepción adicionales en los horarios programados. Para que el dispositivo final abra su puerta de recepción a la hora programada. Esto permite que el servidor sepa cuándo está escuchando el dispositivo final.

- **Dispositivos finales bidireccionales con ranuras de recepción máximas (Clase C):** los dispositivos finales de Clase C están en modo escucha de recibir paquetes casi continuamente, siempre y cuando no estén transmitiendo. Los dispositivos de clase C entregan mejor latencia de conexión a costa de mayor consumo energético con respecto a las clases A y B.

2.9.7.1. DISPOSITIVO LORA RAK 831

Es un módulo transceptor multicanal de alto rendimiento diseñado para recibir varios paquetes LoRa simultáneamente utilizando diferentes factores de difusión en múltiples canales. Brinda la posibilidad de permitir una comunicación sólida entre una puerta de enlace y una gran cantidad de nodos finales distribuidos en un amplio rango de distancia. [31]



Imagen 8. LoRa RAK 831. [32]

Características:

- Tamaño compacto 80 mm 50 mm x 5 mm
- Tecnología de modulación LoRaWAN®
- Bandas de frecuencia: 433 MHz, 470 MHz, 868 MHz, 915 MHz
- Factores de diseminación ortogonales
- Sensibilidad mínima: -142.5 dBm
- Asignación máxima de enlace: 162 dB
- Interfaz SPI
- Procesador de banda base SX1301
- Emula hasta cuarenta y nueve (49) demoduladores LoRa®
- Doce caminos de demodulación paralelos
- Un demodulador FSK
- Dos frontales SX1257 Tx / Rx Alta frecuencia
- Dos x SX1255 Tx / Rx front-end de baja frecuencia
- Tensión de alimentación: 5 voltios DC
- Interfaz RF optimizada 50
- Potencia de salida de hasta 23 dBm
- Receptor GPS opcional

2.9.7.1.1. ESPECIFICACIONES

El RAK831 proporciona encabezados en la parte inferior, que tienen una separación de 2.54 mm. La descripción de los pines se proporciona en la Tabla a continuación.

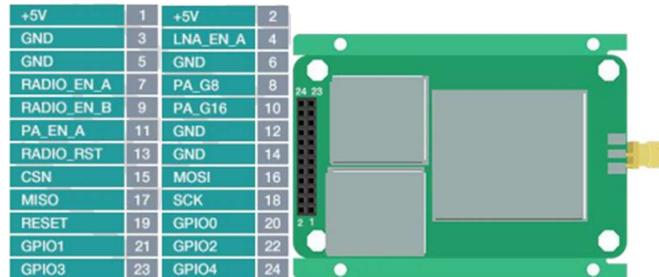


Imagen 9. Pines de LoRa RAK 831. [32]

Tabla 7 Descripción de pines de salida

Pin	Name	Type	Description
1	+5V	POWER	+5V Supply Voltage
2			
3	GND	GND	GND
4	LNA_EN_A	Input	SX 1301 Radio C Sample Valid
5	GND	GND	GPS Module LDO:Enable Pin
6	GND	GND	GND
7	RADIO_EN_A	Input	SX1257_A_EN
8	PA_G8	Input	PA GAIN 0
9	RADIO_EN_B	Input	SX1257_B_EN
10	PA_G16	Input	PA GAIN 1
11	PA_EN_A	Input	PA EN
12	GND	GND	GND
13	RADIO_RST	RST	SX1257_A_B RESET
14	GND	GND	GND
15	CSN	SPI	SX 1301 SPI NSS
16	MOSI	SPI	SX1301 SPI MOSI
17	MISO	SPI	SX1301 SPI MISO
18	SCK	SPI	SX1301 SPI CLK
19	RESET	RST	SX1301 RESET
20	GPIO0	GPIO	SX1301 GPIO
21	GPIO1	GPIO	SX1301 GPIO
22	GPIO2	GPIO	SX1301 GPIO
23	GPIO3	GPIO	SX1301 GPIO
24	GPIO4	GPIO	SX1301 GPIO

2.10. INTERNET DE LAS COSAS (IoT)

Con el Internet de las cosas (IoT) convirtiéndose en un concepto esencial en la sociedad moderna, en la actualidad existen varios tipos de dispositivos que están conectados entre sí, desde microsensores hasta electrodomésticos y automóviles. Hoy por hoy, podemos acceder de forma remota a la mayoría de las cosas del mundo. Las interacciones entre estos dispositivos y los humanos son el núcleo de IoT, y lo que hace posible estas conexiones es la tecnología de comunicación. [33].

El Internet de las cosas se fundamenta en la convergencia de las tecnologías de la información y se puede dar en diversos niveles: la nube, redes de datos, la comunicación y dispositivos de uso diario [34]. Por lo tanto, IoT no es más que lo que obtenemos cuando conectamos las cosas a Internet que no son operadas por los humanos. [35].

2.10.1. ESCENARIOS DE INTERNET DE LAS COSAS

Internet de las cosas está jugando cada vez más un papel clave en varios escenarios como [36]:

- Asistencia sanitaria y bienestar,
- Domótica y automatización de edificios,
- Eficiencia energética mejorada,
- Automatización industrial,
- Medición inteligente e infraestructuras de red inteligentes,
- Monitoreo y pronóstico ambiental,
- Infraestructuras RFID más flexibles,
- Gestión de activos y logística,
- Automatización vehicular y transporte inteligente,
- Agricultura y,
- Compra inteligente

2.10.2. ARQUITECTURA IOT

Diferentes investigadores han propuesto arquitecturas para IoT que acumulan con mayor o menor detalle los aspectos de IoT. A continuación, se muestran algunas de las arquitecturas más destacadas.

Una de las arquitecturas más básicas es la denominada de tres niveles mostrada en la Imagen 2, que incluye los niveles:

- Nivel de percepción: formado por un conjunto de sensores, los cuales captan datos en cualquier medio en el que se encuentren.
- Nivel de red: responsable de conectar los dispositivos (sensores y servidores) entre si para transmitir y recolectar datos.
- Nivel de aplicación: se encarga de brindar y visualizar los servicios requeridos por el usuario. En el caso específico de este proyecto, tiene la función de mostrar los parámetros ambientales.

La arquitectura de cinco niveles se encuentra formada por los niveles de percepción, transporte, proceso, aplicación y nivel de negocio. En esta arquitectura los niveles de percepción y aplicación tienen las mismas características que los de arquitectura de tres niveles.

El nivel de transporte es aquel que transfiere los datos de los sensores mediante redes 3G, LAN, Bluetooth, Identificación por radiofrecuencia (RFID) y Comunicaciones de campo cercano (NFC), desde el nivel de percepción al nivel de proceso y viceversa.

El nivel de proceso almacena, analiza y procesa grandes cantidades de información y datos que proceden del nivel de percepción. Facilita y gestiona servicios, los niveles bajos utilizan tecnológicas de base de datos Cloud Computing y Big data. El nivel de negocio gestiona las aplicaciones, el modelo de negocio y la privacidad.

Existen otras arquitecturas que se adaptaran a los aspectos más concretos de IoT, y todo dependerá del tipo de diseño, su finalidad y el servicio que brindaran. [37] [38].

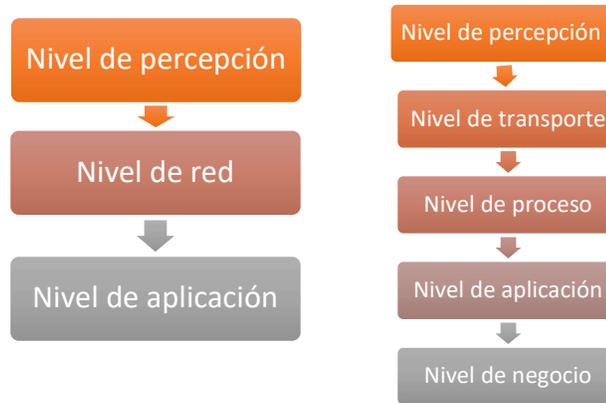


Imagen 10. Arquitectura de 3 y 5 niveles [Autor].

2.11. PLATAFORMAS IOT

Actualmente existen muchas plataformas para interactuar dispositivos independientemente de la complejidad que integra. Una plataforma IoT puede ser desde un simple repositorio de datos y eventos con interfaces estándar, hasta un complejo sistema inteligente que incluya capacidades de predicción, aprendizaje, inteligencia de negocio y otras algoritmas.

El objetivo final de la plataforma abierta de servicios IoT es proporcionar a la aplicación los siguientes servicios;

- Fácil acceso y uso del recurso global de IoT y datos detectados
- fácil conexión de recursos de IoT
- Fácil desarrollo y distribución de varias aplicaciones.

2.11.1. THE THINGS NETWORK

The Things Network (TTN) es una comunidad que pretende construir una red de sensores inalámbrica global y descentralizada de IoT a bajo costo. Se basa en la tecnología LoRaWAN, ya que son miembros de LoRa Alliance. Por lo tanto, se puede considerar The Things Network como un proveedor de red LoRaWAN. [39].

Es una plataforma de código abierto y permite conectar una amplia variedad de nodos, ofrece una función de interfaz web completa y útil para ver los datos que hayan sido

enviados. TTN es responsable de decodificar los datos de los Gateway y transmitirlos a la aplicación de terceros.

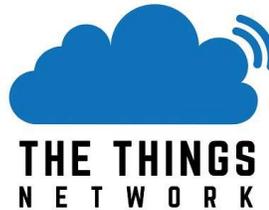


Imagen 11. Logo TTN. [40]

Además, ofrece capacidades de integración por HTTP y MQTT, asimismo de una serie de APIs en distintos lenguajes como: Go, Java, Node-RED y Node.js, con las cuales construir una aplicación end-to-end, mediante la integración de nodos, gateways y server de TTN. [41].

2.11.2. UBIDOTS

Es una plataforma de IoT que capacita a innovadores e industrias para crear prototipos y escalar proyectos de IoT a producción. El uso de la plataforma Ubidots es para enviar datos a la nube desde cualquier dispositivo con acceso a Internet, además de configurar acciones y alertas basadas en sus datos en tiempo real y desbloquear el valor de sus datos a través de herramientas visuales.



Imagen 12. Logo Ubidots [42]

Ubidots ofrece una API REST que le permite leer y escribir datos en los recursos disponibles: fuentes de datos, variables, valores, eventos e información. La API admite HTTP y HTTPS y se requiere una clave API.

Los datos estarán protegidos con dos réplicas más, almacenamiento encriptado y soporte opcional de datos TLS / SSL. También puede personalizar los grupos de permisos para cada módulo de la plataforma, asegurándose de que se muestre la información correcta al usuario correcto.

2.12. ARDUINO

Arduino es una plataforma electrónica de código abierto basada en hardware y software fácil de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida: activar un motor, encender un LED, publicar algo en línea. Puede decirle a su placa qué hacer enviando un conjunto de instrucciones al microcontrolador de la placa. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en el cableado) y el software Arduino (IDE), basado en el procesamiento. [43].

2.12.1. ARDUINO MKR WAN 1300

Arduino MKR WAN 1300 (LoRa Connectivity) combina la funcionalidad de la conectividad MKR Zero y LoRa. La placa es una solución rentable para los fabricantes que buscan agregar conectividad Lo-Ra a proyectos de IoT con experiencia mínima en redes. El MKR WAN 1300 se basa en el Atmel SAMD21 y un módulo Murata CMWX1ZZABZ Lo-Ra. La placa puede alimentarse con dos baterías "AA", "AAA" de 1.5V o 5V externos. [44].

2.12.1.1. ESPECIFICACIONES

- Microcontrolador: SAMD21 Cortex-M0 + 32bit ARM de baja potencia.
- Fuente de alimentación de la placa: (USB / VIN) 5V
- Baterías compatibles: 2x AA o AAA.
- Voltaje de funcionamiento del circuito: 3.3V.
- Pines de E / S digital: 8.

- Pines PWM: 12 (0, 1, 2, 3, 4, 5, 6, 7, 8, 10, A3 o 18, A4 o 19).
- UART: 1
- SPI: 1
- I2C: 1
- Pines de entrada analógica: 7 (ADC 8/10/12 bit).
- Pines de salida analógica: 1 (DAC 10 bit)
- Interrupciones externas: 8 (0, 1, 4, 5, 6, 7, 8, A1 -o 16-, A2- o 17).
- Corriente CC por pin de E / S: 7 mA
- Memoria Flash: 256 KB
- SRAM: 32 KB
- EEPROM: no
- Velocidad de reloj: 32.768kHz (RTC), 48MHz
- LED_BUILTIN: 6
- Dispositivo USB de velocidad completa y host integrado
- Potencia de antena: 2dB
- Frecuencia portadora: 433/868 / 915MHz
- Región de trabajo: UE / EE. UU.
- Longitud: 67.64mm
- Ancho: 25 mm
- Peso: 32g

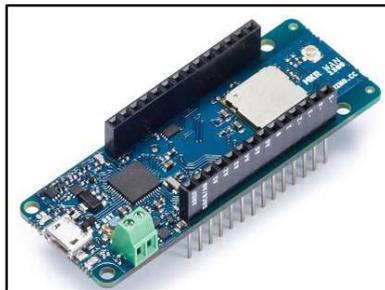


Imagen 13. Arduino MKR WAN 1300. [45]

DISEÑO E IMPLEMENTACIÓN DE LA RED DE SENSORES IOT

En este capítulo se explicará el procedimiento seguido para desplegar una red IoT usando tecnología LoRa y se describe el desarrollo del hardware y software del proyecto. Se muestra el desarrollo para implementar una red IoT con la tecnología LoRa.

3.1. SISTEMATIZACIÓN DEL PROTOTIPO

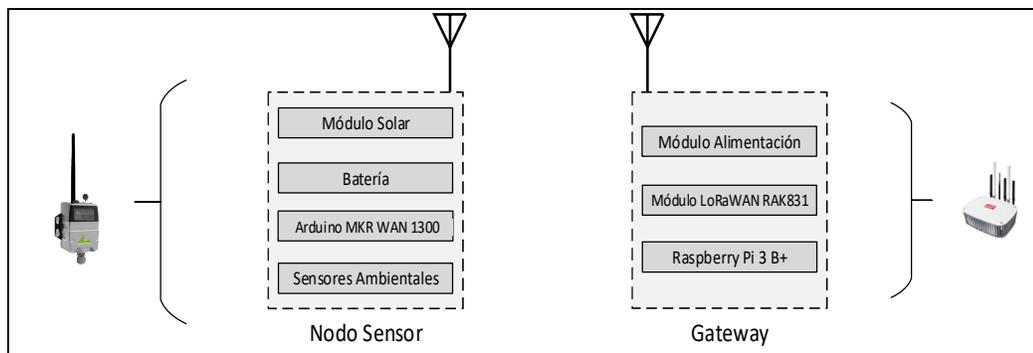


Imagen 14. Esquema del prototipo [Autor].

En esta etapa se procedió a definir la construcción e implementación de los nodos móviles y el nodo Gateway, teniendo en cuenta que se debe usar el método más adecuado (serial, SPI, I2C) para la captura y envío de datos de forma rápida y eficiente, de tal manera que el tamaño del paquete enviado sea lo más pequeño posible y se pueda ser visualizado en una plataforma web.

Entre los parámetros que serán medidos por los nodos móviles son: temperatura, presión, humedad, rayos UV, partículas en el aire y ruido, dichas variables serán tomados de manera simultánea en cada cierto intervalo de tiempo.

Para el monitoreo los parámetros ambientales, se contará con cuatro nodos sensores. Se destinará un nodo para el monitoreo en el interior de una edificación y tres nodos para exteriores, además una puerta de enlace que recibirá los datos. Dichos datos serán procesados de forma inalámbrica con la tecnología LoRaWAN que permite comunicación a largas distancias.

Este proyecto de titulación se basa en la tecnología LoRaWAN ya que posee mejores parámetros con referencia a las otras tecnologías de comunicación inalámbrica, su rango de cobertura es de largo alcance, por esto es ideal la aplicación en áreas extensas con irregularidades geográficas.

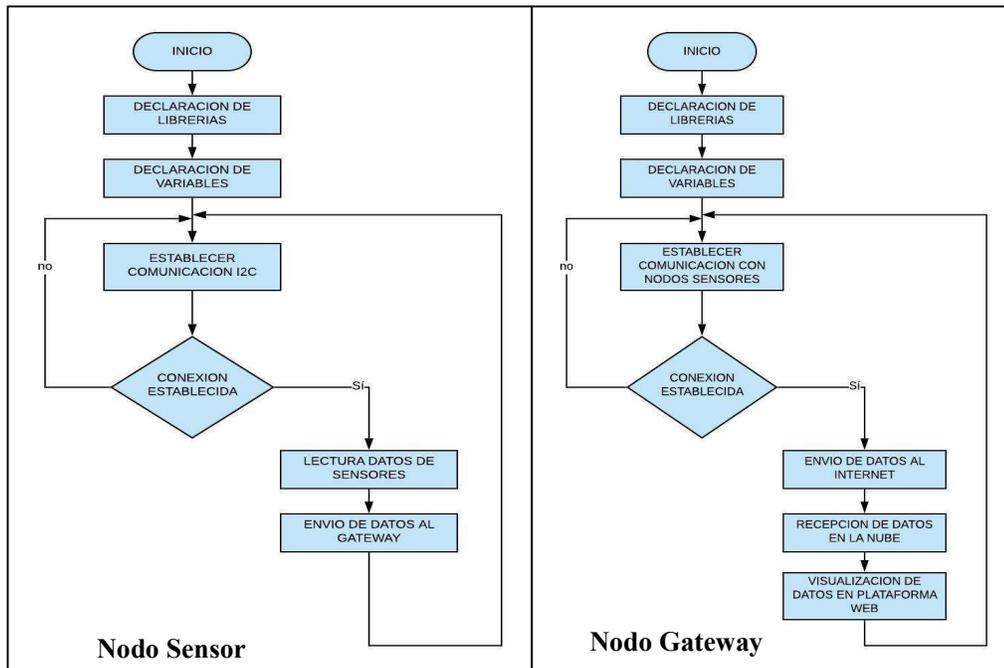


Imagen 15. Diagrama de flujo de Nodo Sensor – Nodo Gateway [Autor].

3.2. DESPLIEGUE DE RED LORA IMPLEMENTADO

Una vez establecido el esquema base del proyecto de titulación, identificamos 4 nodos a usar los cuales son:

- Nodo Gateway
- Nodo de sensor (recolección de datos)
- Nodo de monitoreo (Plataforma Ubidots)

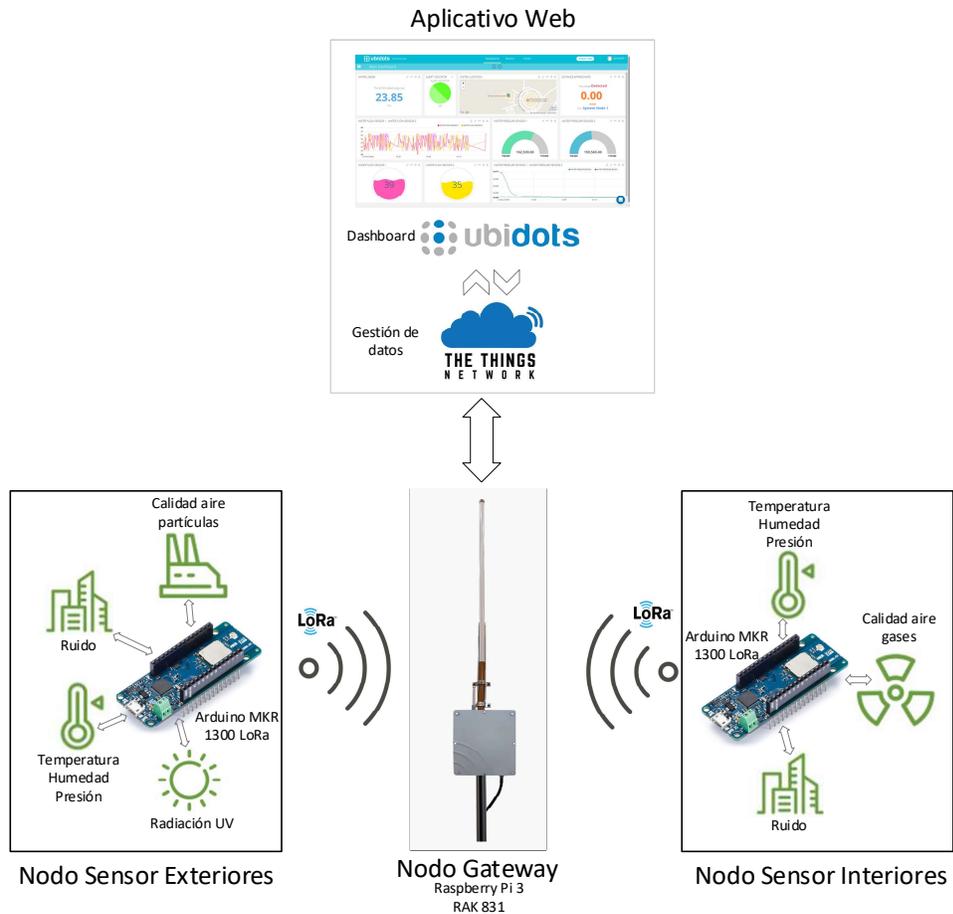


Imagen 16. Arquitectura del hardware [Autor].

3.2.1. NODO GATEWAY

Este nodo será el encargado de recibir los datos de los distintos nodos y enviarlos a la Internet para ser monitoreados en tiempo real a través de una interfaz web. Para lograr esto, se utiliza un dispositivo LoRa RAK811, un Raspberry PI 3 y una antena 915 MHz.

Cada uno de los dispositivos es energizado con 5 voltios y transmiten los datos por SPI del dispositivo LoRa al Raspberry el cual enviará el paquete de dato al nodo de almacenamiento.

3.2.1.1. CONFIGURACIÓN DEL NODO GATEWAY

Para habilitar el nodo Gateway se utilizó un módulo LoRa RAK811 y un raspberry PI 3.

Tabla 8 Conexión de pines entre RAK811 y Raspberry PI 3. [46]

RAK 811 Pin	Description on silk screen	RPI physical pin
1	+5 (V)	2
3	GND	6
19	RST (Reset Pin)	22
18	SCK (SPI Clock)	23
17	MISO	21
16	MOSI	19
15	CSN (Chip Select)	24

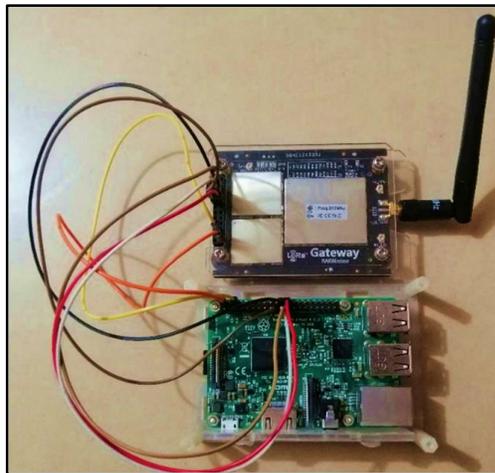


Imagen 17. Módulo LoRa RAK831, Raspberry PI 3 [Autor].

- ✓ Antes de empezar la configuración del software del Raspberry PI 3, se necesitó una tarjeta microSD de 8 Gb con el software Raspbian.
- ✓ Antes de encender la placa RAK811, se colocaron la antena que vienen en su kit en el terminal de tornillo de las antenas.
- ✓ Se realiza las respectivas conexiones entre el módulo LoRa RAK811 y el Raspberry PI 3.
- ✓ El periférico SPI no está activado de manera predeterminada. Para habilitarlo se procedió lo siguiente:

- Ejecutar **sudo raspi-config**.
- Use la flecha hacia abajo para seleccionar **9 Advanced Options**.
- Flecha hacia abajo para **A6 SPI**.
- Seleccione **yes** cuando le pide que habilite SPI,
- También seleccione “**yes**” cuando pregunte acerca de cargar automáticamente el módulo del kernel.
- Use la flecha derecha para seleccionar el **<Finish>** botón.
- Seleccione “**yes**” cuando se solicita reiniciar.

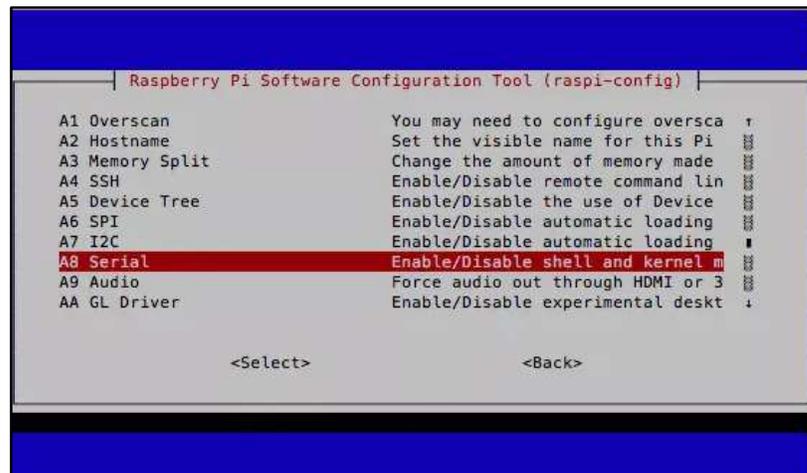


Imagen 18. Configuración Raspberry PI 3 para SPI. [46].

- El sistema se reiniciará y cuando vuelva a funcionar, inicie sesión e ingrese el siguiente comando **>ls /dev/*spi***
- ✓ Una vez reiniciada nuestra Raspberry, realizamos la respectiva configuración de la red inalámbrica para poder tener acceso a internet, abrimos una terminal y procedemos a escribir los siguientes comandos que actualizaran nuestro sistema operativo y se instalará un aplicativo llamado GIT:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install git
```

- ✓ Acto seguido procedemos a copiar el repositorio desde la página Github con los siguientes comandos, los cuales también lo instalarán en la Raspberry:

```
$ git clone -b spi https://github.com/ttn-zh/ic880a-gateway.git~/ic880a-gateway
$ cd ~/ic880a-gateway
$ sudo ./install.sh spi
```

- ✓ Al terminar la instalación se nos entregara una ID que dependerá del MAC Address de la interfase física de salida que estemos usando para la conexión a Internet además nos preguntara si deseamos crear un archivo con formato JSON que podríamos subir al repositorio para una configuración remota; esta ID nos servirá para realizar en registro en la plataforma de The Things Network.

3.2.1.2. REGISTRO DEL GATEWAY EN EL TTN

Una vez realizado los pasos anteriores se procede con el registro del Gateway a la red The Things Network (TTN) para que, los nodos envíen datos a la nube y así poder ser visualizado.

Desde TTN existen dos tipos de reenviadores de paquetes que se pueden ejecutar en el Gateway, según el tipo de protocolo de red que utilicen:

- Reenviadores de paquetes que se conectan utilizando el protocolo Semtech UDP (como el reenviador de paquetes Semtech UDP). Este protocolo no está encriptado, es menos confiable (basado en UDP), pero su simplicidad hace que funcione en una gran cantidad de gateways.
- Reenviadores de paquetes que se conectan utilizando el nuevo protocolo TTN Gateway Connector (como el Reenviador de paquetes TTN). Este protocolo es más seguro, confiable y seguro. [47].

El primer paso es crear una cuenta en TTN en donde se van a registrar los nodos y la aplicación para poder trabajar con los paquetes de datos recibidos.

Al acceder a la cuenta, nos dirigimos a la pestaña consola en donde nos presenta dos opciones: Applications y Gateway, seleccionamos Gateway para proceder a registrarlo.

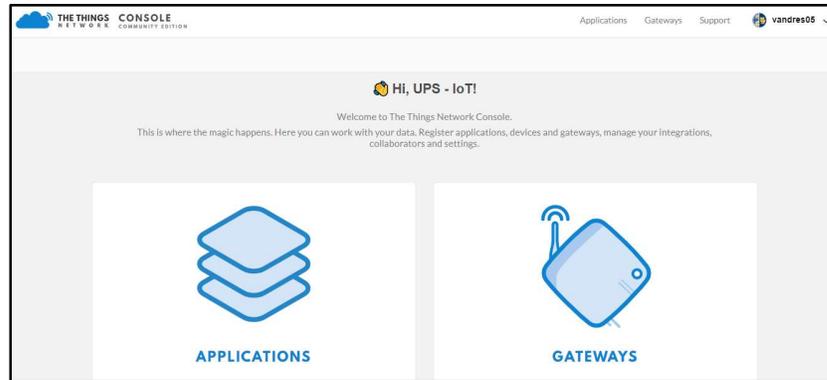


Imagen 19. Consola de cuenta en TTN [Autor].

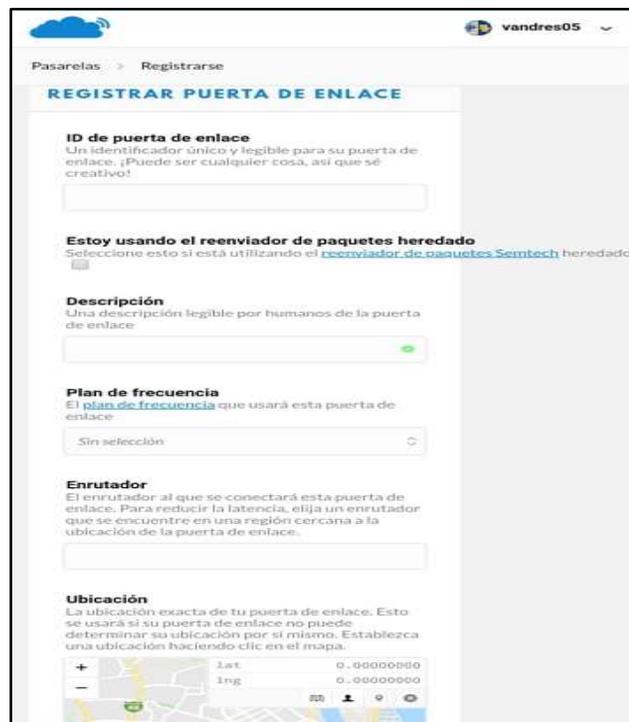


Imagen 20. Registro de Gateway en TTN [Autor].

- Para **ID de puerta de enlace**, elija una ID única de minúsculas, caracteres alfanuméricos y no consecutivos.
- Seleccione el plan de frecuencia (determinado por su región) que usa la puerta de enlace.

- Haga clic para colocar el marcador en la ubicación exacta (desplazar y acercar antes de soltar).
- Haga clic en **Registrar puerta de enlace** para finalizar. [47].

Al finalizar el registro, aparecerá en la pestaña “Gateways”.



Imagen 21. Nodo gateway registrado en TTN [Autor].

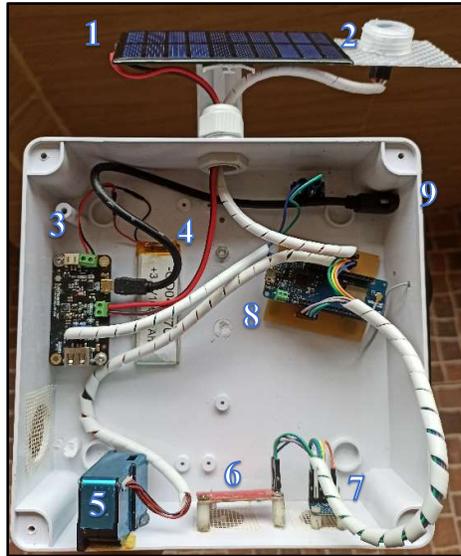
3.2.2. NODO SENSORES.

Son 4 nodos sensores que medirán parámetros ambientales tales como: rayos UV, temperatura, calidad del aire, humedad y ruido, con el fin de determinar las condiciones ambientales de la zona.

Se dispone de tres nodos outdoors y un indoor, los cuales se encargarán de adquirir los datos en tiempo real de la zona en la que fueron instalados y los enviará al Gateway para su procesamiento.

El nodo sensor outdoor estará conformado de los siguientes hardware:

- Sensor de rayos UV (Sensor VEML6075)
- Sensor de temperatura, presión, humedad (Sensor BME280)
- Sensor de partículas en el aire PM 2.5 y 10 (Sensor PMS3003)
- Sensor de ruido (Sparkfun Sound Detector)
- Arduino MKR 1300
- Módulo de gestión energía solar DRF559
- Panel solar
- Batería Lipo 900mAh 3.7 V
- Antena con frecuencia 915 MHZ

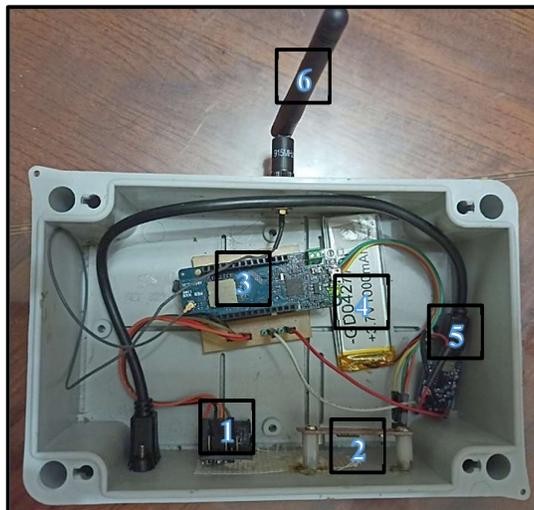


- 1 Panel Solar
- 2 Sensor VEML6075
- 3 Módulo DRF559
- 4 Batería LiPo
- 5 Sensor de partículas
- 6 Sensor de ruido
- 7 Sensor BME280
- 8 Arduino
- 9 Antena

Imagen 22. Armado de nodo sensor Outdoor [Autor].

El nodo sensor indoor estará conformado de los siguientes hardwares:

- Sensor de temperatura, presión, humedad, calidad del aire (Sensor BME680)
- Sensor de ruido (Sparkfun Sound Detector)
- Arduino MKR 1300
- Batería Lipo 900mAh 3.7 V
- Módulo de carga micro Usb para batería (TP4056)
- Antena con frecuencia 915 MHZ



- 1 Sensor BME680
- 2 Sparkfun Sound Detector
- 3 Arduino MKR 1300
- 4 Batería Lipo
- 5 Módulo de carga Micro USB
- 6 Antena

Imagen 23. Armado de nodo sensor indoor [Autor].

3.2.2.1. CONFIGURACIÓN DEL NODO SENSORES

Usando el gestor de librerías del programa Arduino IDE buscamos la librería MKRWAN, esta es dedicada al transceptor LoRa del Arduino y con la cual obtendremos el devEui que es un identificador único de 64 bits de cada dispositivo y nos servirá junto con las credenciales appEui, appKey que obtendremos luego de la plataforma The Things Networks, necesarias para una conexión exitosa.



Imagen 24. DevEui del nodo 4 [Autor].

Para la programación de los Arduinos es necesaria la instalación de librerías adicionales que pertenecen a los distintos sensores a usar en el proyecto, además de la librería Lora usada para la transmisión de datos teniendo en cuenta que la banda a usar es 915 MHz (dependiendo de la región se deberá elegir la banda correcta).

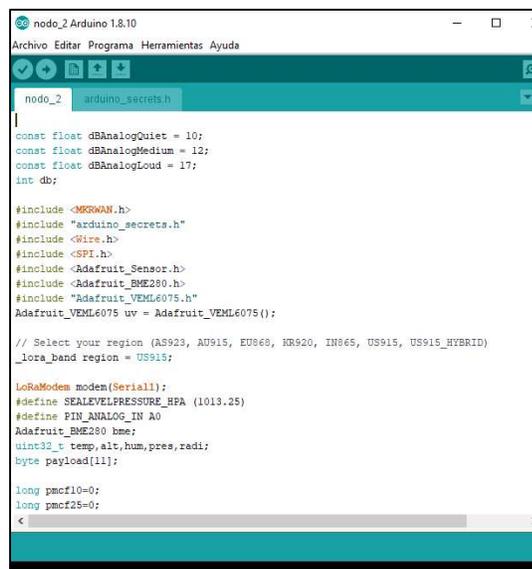


Imagen 25. Programación del arduino MKR WAN 1300 [Autor].

El envío de datos de los diversos sensores se agrupa en un solo paquete (payload) que contendrá también las credenciales previamente mencionadas, para una mejor transferencia de los datos hacia la nube las variables serán enviadas como bytes, de esta manera se evita un peso excesivo del payload. Una vez que este paquete sea recibido se debe realizar la respectiva decodificación para la interpretación de los datos.



```
nodo_2 | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
nodo_2 arduino_secrets.h
modem.beginPacket();
payload[0]=highByte(temp);
payload[1]=lowByte(temp);
payload[2]=highByte(hum);
payload[3]=lowByte(hum);
payload[4]=highByte(radi);
payload[5]=lowByte(radi);
payload[6]=highByte(pres);
payload[7]=lowByte(pres);
payload[8]=pmcf25;
payload[9]=pmcf100;
payload[10]=dB;
```

Imagen 26. División de variables por bytes [Autor].

3.2.2.2. REGISTRO DE NODOS SENSORES EN EL TTN

En la plataforma IoT, The Things Network (TTN) se realiza el registro de los equipos a usar (arduinos). Para este proyecto se usarán sensores indoor y outdoor, por lo que se creó dos aplicaciones en la plataforma.



Imagen 27. Creación de aplicación para sensores indoor y outdoor [Autor].

En la aplicación sensores outdoor se registraron 3 dispositivos y en la aplicación indoor se registró 1 dispositivo, cada aplicación registrará parámetros distintos según se explicó al inicio del documento.

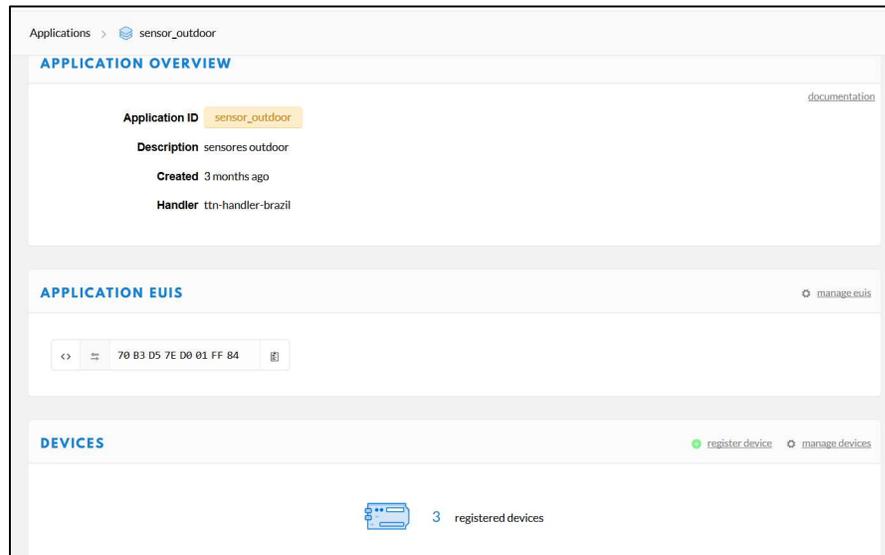


Imagen 28. Vista de la aplicación Outdoor [Autor].

Procedemos a registrar cada dispositivo y asignar un nombre único el cual no podrá ser cambiado posteriormente y colocamos el DevEui obtenido antes; la plataforma nos proporcionará automáticamente el appEui y appKey, siendo el primero el identificativo de la aplicación y el segundo una clave de ingreso de 128 bits, estas credenciales son únicas

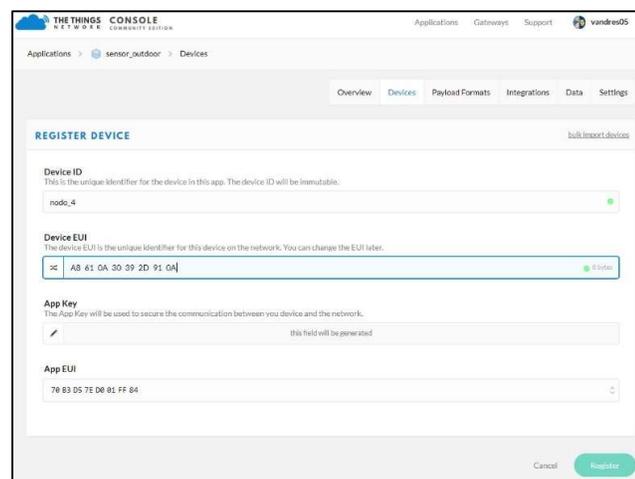


Imagen 29. Registro de dispositivo [Autor].

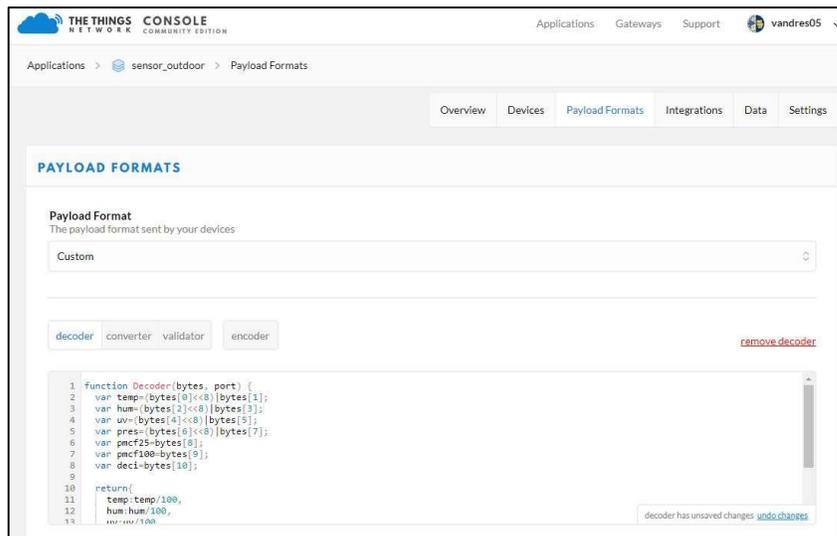
Posterior a esto, regresamos a completar el programa de nuestro Arduino, la librería MKRWAN creará un archivo adicional llamado `arduino_secrets.h`, en el cual se copiarán las credenciales recientemente obtenidas (`appEui`, `appKey`), garantizando la conexión cifrada hacia el gateway y la llegada de datos hacia la plataforma.



```
nodo_4 - arduino_secrets.h | Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda
nodo_4 arduino_secrets.h $
// Copy these keys from The Things Network console and paste them here:
const char *appEui = "70B3D57ED001FF04";
const char *appKey = "704C93B773700423FF9B1C3A92B43CB4";
```

Imagen 30. Credenciales de acceso del dispositivo [Autor].

Como el payload se encuentra cifrado de extremo a extremo la parte lógica de la plataforma se encargará de descifrarlo, pero el creador de la aplicación también podrá manejar su propio entorno privado para lograr esto. En nuestro caso hemos creado un decodificador para la carga útil (variables enviadas) a un formato que permita ser leído fácilmente por el aplicativo, estas funciones son realizadas en JSON. La plataforma nos permitirá descifrar, convertir y validar los datos que recibe.



```
THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications Gateways Support vandres05
Applications > sensor_outdoor > Payload Formats
Overview Devices Payload Formats Integrations Data Settings
PAYLOAD FORMATS
Payload Format
The payload format sent by your devices
Custom
decoder converter validator encoder remove decoder
1 function Decoder(bytes, port) {
2   var temp=(bytes[0]<<8)|bytes[1];
3   var hum=(bytes[2]<<8)|bytes[3];
4   var uv=(bytes[4]<<8)|bytes[5];
5   var pres=(bytes[6]<<8)|bytes[7];
6   var pmcf25=bytes[8];
7   var pmcf100=bytes[9];
8   var deci=bytes[10];
9
10  return {
11    temp: temp/100,
12    hum: hum/100,
13    uv: uv/100,
```

Imagen 31. Decodificación de datos con JSON [Autor].

Esta programación quedará registrada en la plataforma y los únicos cambios que diferirán de las dos aplicaciones serán los nombres de las variables que son enviadas por cada nodo.

3.2.3. NODO MONITOREO (APLICATIVO WEB)

Este nodo se encarga de la visualización de cada uno de los parámetros medidos por los sensores, para tal motivo este aplicativo debe ser compatible con una amplia gama de plataformas IoT, de tal forma que garantice la compatibilidad entre todos los dispositivos.

Para esto se debe analizar de las herramientas existentes en el mercado, entre las más destacables, ThingSpeak, Ubidots entre otras. Para el desarrollo de este proyecto se usará Ubidots el cual permite varias funciones además de tener una interfaz amigable para el usuario final.

3.2.3.1. CONFIGURACIÓN DEL NODO MONITOREO

Para la visualización de datos se usará la plataforma Ubidots la misma que se configurará en base a nuestra necesidad.

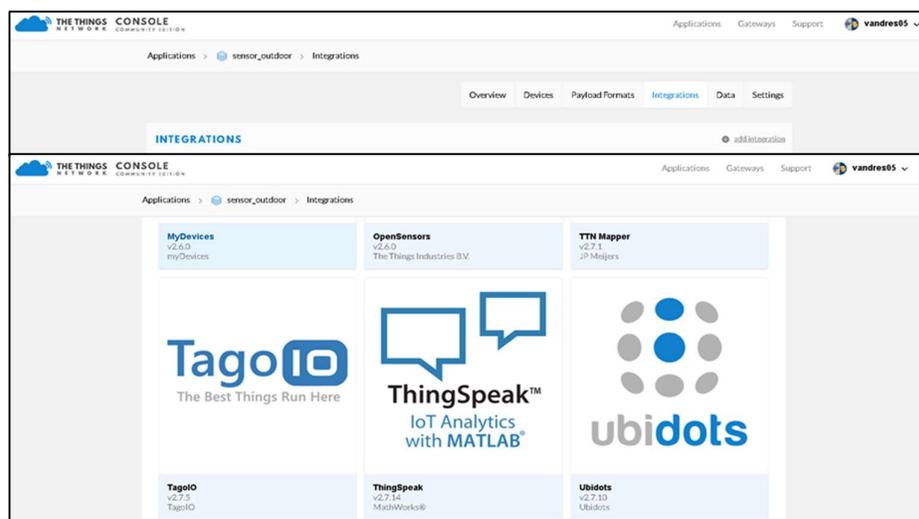


Imagen 32. Integración con la plataforma Ubidots [Autor].

Para que se lleve a cabo la integración, se necesitan 3 datos: Process ID, que es solo un identificador único que asignamos nosotros y el cuál se encuentra directamente relacionado a los nodos y las aplicaciones creadas: sensor_outdoor y sensor_indoor; el Access key, que siempre se elige la configuración por defecto o default, y el token, este es otorgado por la cuenta de Ubidots al momento de ser creada y permite el envío de datos desde TTN. Cabe recalcar que la integración solo funciona con el endpoint industrial.ubidots.com.

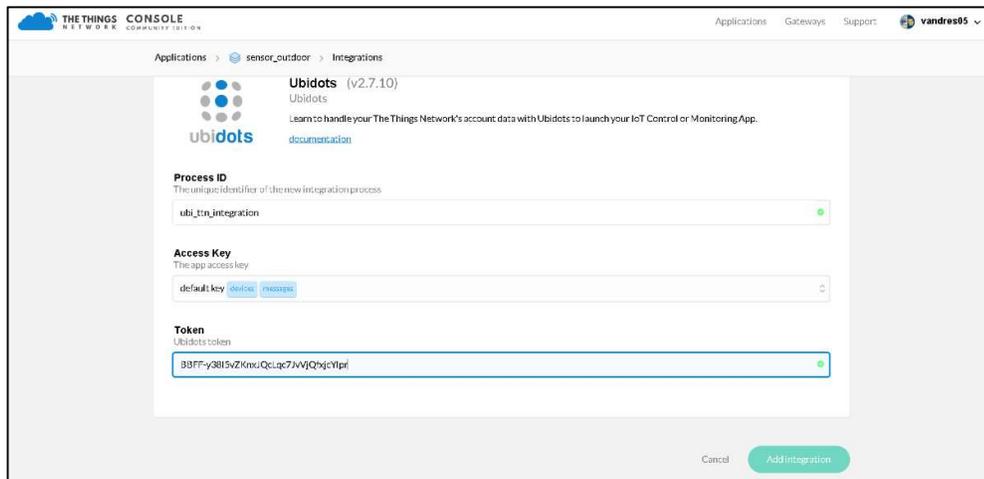


Imagen 33. Configuración en TTN para integrar con Ubidots [Autor].

Una vez creada la integración, se podrá verificar su estado (Running)

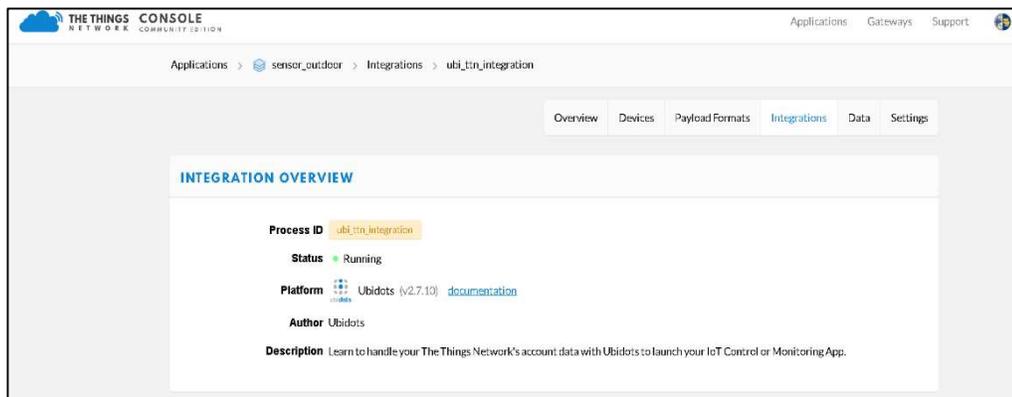


Imagen 34. Verificación de integración con Ubidots [Autor].

Una vez hecho esto del lado de The Things Network, podemos proceder a preparar nuestro dashboard (representación gráfica) en Ubidots. Una vez que se enciendan los equipos asociados con el Gateway en nuestra cuenta de TTN y comiencen a enviar datos al mismo, veremos que en nuestra cuenta de Ubidots bajo el menú Devices, opción Devices, aparecerán nuestros dispositivos, el nombre que se les asigna por defecto es el MAC asignado en la aplicación de TTN al crearlos.

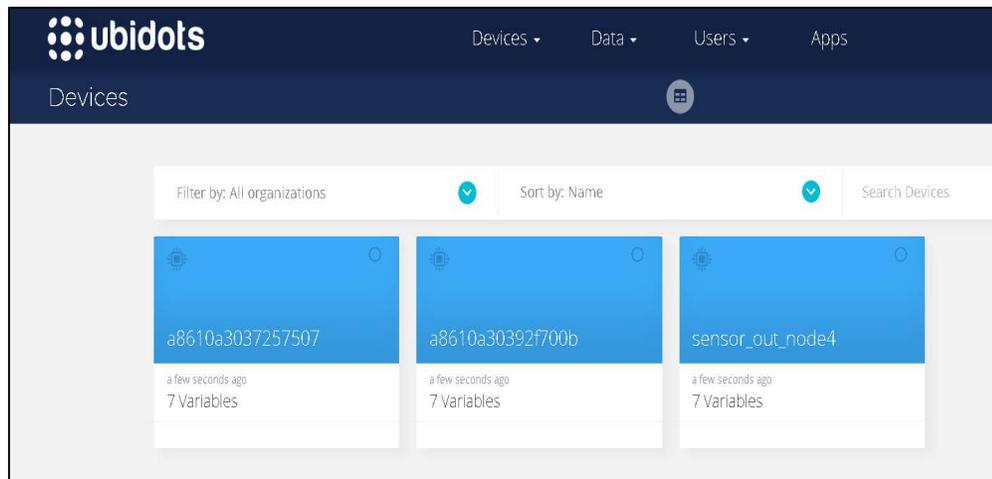


Imagen 35. Panel principal de Ubidots [Autor].

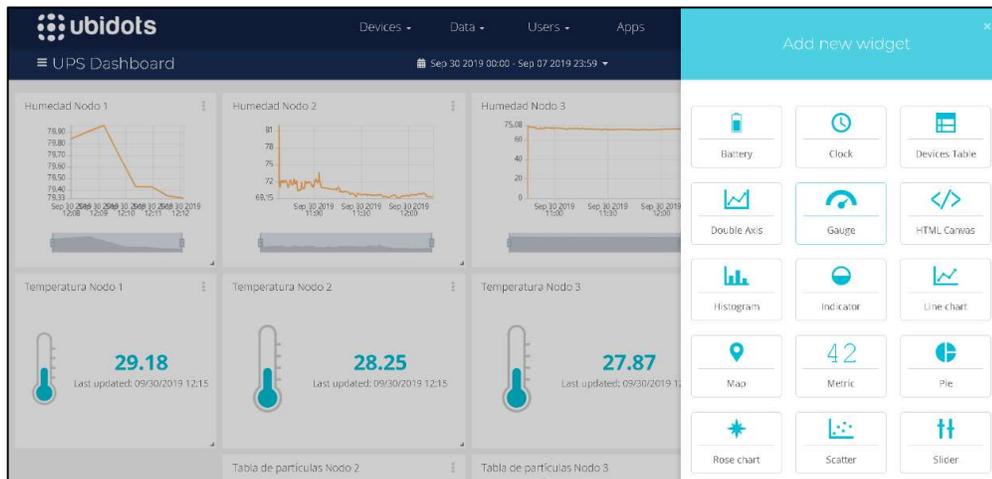


Imagen 36. Diseño del dashboard con widgets [Autor].

Una vez que vayamos añadiendo los diferentes widgets para mostrar nuestra información podemos proceder a acomodarlos en nuestro dashboard, cambiando su ubicación y tamaño inicial como se muestra en la figura a continuación.

3.3. DISEÑO DE LAS PLACAS

3.3.1. NODO 1

La Imagen consta del diseño de la placa para los nodos, como su principal componente se encuentra el Arduino MKR 1300, sensores de ruido, humedad, presión, temperatura y calidad del aire.

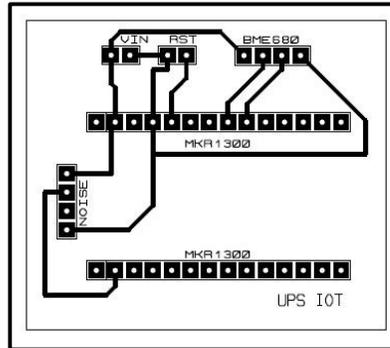


Imagen 37. Diseño de Placa Nodo 1 [Autor].

3.3.2. NODO 2, 3 Y 4

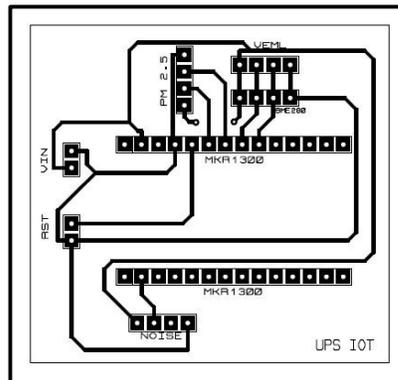


Imagen 38. Diseño de Placa Nodo 2, 3 y 4 [Autor].

La Imagen 38 consta del diseño de la placa para los nodos 2,3 y 4. Tiene como su principal componente el Arduino MKR 1300, y de sensores: ruido, humedad, presión, temperatura, radiación solar y partículas PM 2.5 y PM 10.

3.4. PROTOTIPO FINAL DEL NODO GATEWAY

Se va a usar una caja de pase PVC, cuyas dimensiones son 15x15 y una antena 915 MHz.



Imagen 39. Vista frontal y antena del nodo Gateway [Autor].

3.5. PROTOTIPO FINAL DEL NODO 1

Se va a usar una caja de pase PVC, cuyas dimensiones son 8x14.



Imagen 40. Vista frontal y lateral de los Nodo 1 [Autor].

3.6. PROTOTIPO FINAL DEL NODO 2,3 Y 4

Se va a usar una caja de pase PVC, cuyas dimensiones son 15x20.



Imagen 41. Caja de pase PVC [Autor].



Imagen 42. Vista frontal de los Nodo 2,3 y 4 [Autor].



Imagen 43. Vista lateral de los Nodo 2,3 y 4 [Autor].

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

Teniendo como antecedente las medidas de distanciamiento social implementadas por la emergencia sanitaria, en el presente capítulo se planteará el análisis de resultados adquiridos en las pruebas realizadas considerando una distancia de 600 metros entre el Gateway y dispositivos finales, procurando que exista línea de vista con la antena del nodo Gateway; no obstante, la tecnología LoRaWAN se caracteriza por su comunicación punto a punto de largo alcance.

4.1. INTERPRETACIÓN DE VARIABLES AMBIENTALES PUBLICADAS EN LA PLATAFORMA UBIDOTS

Para efectos de interpretación de datos se tomará en consideración los valores máximos obtenidos en el siguiente periodo de tiempo: 18/07/2020 de 12:30 a 16:00.

Se eligió dicho horario debido a que es un período idóneo para la toma de lecturas por las variables humedad, material particulado, temperatura, radiación solar y ruido; además, al ser un horario pico se obtendrían valores máximos.

4.1.1. SENSACIÓN TÉRMICA

La sensación térmica es la temperatura que el cuerpo humano percibe ante la combinación de parámetros ambientales, humedad y temperatura.

Para obtener la sensación térmica se seleccionará los parámetros indicados previamente y se los relacionará utilizando la Imagen 49.

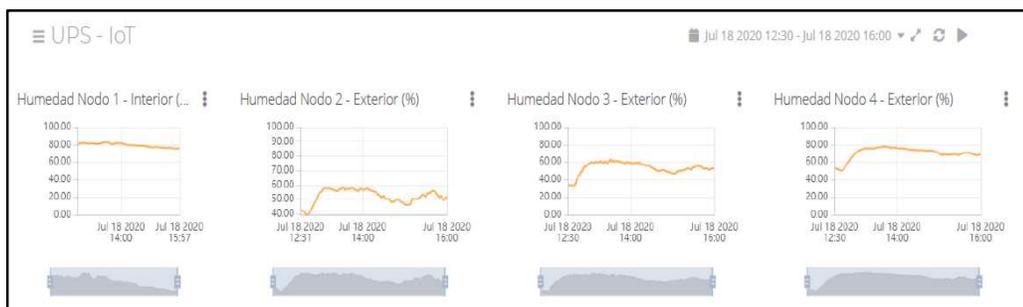


Imagen 44. Humedad, obtenidas de Ubidots [Autor]



Imagen 45. Temperatura obtenida de Ubidots [Autor]

TABLA DE VALORES DE SENSACIÓN TÉRMICA POR CALOR (HEAT INDEX)

		TEMPERATURA DEL AIRE EN GRADOS CELSIUS (C)																		
		27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	
HUMEDAD RELATIVA (%)	45	27	28	29	30	32	33	35	37	39	41	43	46	49	51	54	57	61	64	
	50	27	28	30	31	33	34	36	38	41	43	46	49	52	55	58	62			
	55	28	29	30	32	34	36	38	40	43	46	48	52	55	59	62				
	60	28	29	31	33	35	37	40	42	45	48	51	55	59	63					
	65	28	30	32	34	36	39	41	44	48	51	55	59	63						
	70	29	31	33	35	38	40	43	47	50	54	58	63							
	75	29	31	34	36	39	42	46	49	53	58	62								
	80	30	32	35	38	41	44	48	52	57	61									
	85	30	33	36	39	43	47	51	55	60	65									
	90	31	34	37	41	45	49	54	58	64										
95	31	35	38	42	47	51	57	62												
100	32	36	40	44	49	54	60													

Precaución 27 a 32 Posible fatiga por exposición prolongada o actividad física.
 Precaución extrema 33 a 40 Insolación, golpe de calor, calambres. Posibles por exposición prolongada o actividad física.
 Peligro 41 a 53 Insolación, golpe de calor, calambres. Muy posibles por exposición prolongada o actividad física.
 Peligro extremo 54 o más Golpe de calor, insolación inminente.

Imagen 46. Valores de sensación térmica [48].

Tabla 9 Análisis de la sensación térmica [Autor].

Nodo	Temperatura [°C]	Humedad [%]	Sensación térmica [°C]	Afectaciones
1 - Indoor	31.13	81.86	41	Insolación, golpe de calor, calambres. Muy posibles por exposición prolongada o actividad física.
2 - Outdoor	33.71	58.00	38	Insolación, golpe de calor, calambres. Posibles por exposición prolongada o actividad física.
3 - Outdoor	33.41	63.79	41	Insolación, golpe de calor, calambres. Muy posibles por exposición prolongada o actividad física.
4 - Outdoor	33.84	77.65	46	Insolación, golpe de calor, calambres. Muy posibles por exposición prolongada o actividad física.

4.1.2. MATERIAL PARTICULADO

Para la interpretación de este parámetro ambiental se utilizará la Imagen 52



Imagen 47. Material particulado, nodo 1 y 2 obtenidas de Ubidots [Autor]



Imagen 48. Material particulado, nodo 3 y 4 obtenidas de Ubidots [Autor]

ICCA	Calidad del aire Material particulado (PM)	Contaminante ($\mu\text{g}/\text{m}^3$)		Indicaciones para su salud
		PM2.5	PM10	
0 - 50	Buena	0- 15.3	0- 54	No se anticipan impactos a la salud cuando la calidad del aire se encuentra en este rango.
51 - 100	Moderada	15.5-40.2	56-154	No se anticipan impactos a la salud cuando la calidad del aire se encuentra en este rango.
101 - 150	Dañina a la Salud de los Grupos Sensibles	40.5-65.4	155-254	Los niños y adultos activos, y personas con enfermedades respiratorias tales como el asma, deben evitar los esfuerzos físicos excesivos y prolongados al aire libre.
151 - 200	Dañina a la Salud	66-159	255-354	La gente con la enfermedades respiratorias tal como asma, debe evitar el esfuerzo al aire libre; todos los demás, especialmente los mayores y los niños, deben limitar el esfuerzo prolongado al aire libre.
201 - 300	Muy dañina a la Salud	160-250	355-424	La gente con enfermedades respiratorias tal como asma, debe evitar todo el esfuerzo al aire libre; especialmente los mayores y los niños, deben limitar el esfuerzo prolongado al aire libre.
301 - 500	Peligroso	251-500	424-604	Todos deben evitar el esfuerzo al aire libre; gente con la enfermedad respiratoria tal como asma, debe permanecer dentro

Imagen 49. Leyenda para el análisis de material particulado y calidad del aire [49].

Tabla 10 Interpretación del material particulado y calidad del aire [Autor].

Nodo	ICCA	PM 10 [µg/m3]	PM 2,5 [µg/m3]	Calidad del Aire
1 - Indoor	87.85	N/A	N/A	Moderada
2 - Outdoor	N/A	21	22	Moderada
3 - Outdoor	N/A	28	24	Moderada
4 - Outdoor	N/A	25	23	Moderada

4.1.3. ÍNDICE DE RADIACIÓN SOLAR UV

Para la interpretación de este parámetro ambiental se utilizará la Imagen 55

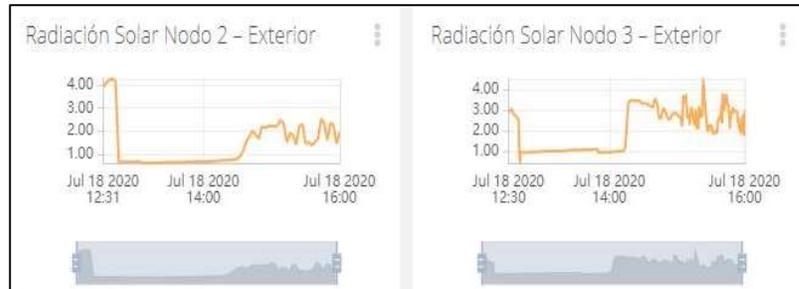


Imagen 50. Índice de radiación solar, nodo 2, 3 obtenidas de Ubidots [Autor]



Imagen 51. Índice de radiación solar, nodo 4 obtenidas de Ubidots [Autor]



Imagen 52. Leyenda de índice UV [50]

Tabla 11 Interpretación de índice UV [Autor].

Nodo	Índice UV	Radiación UV	Precauciones
2 - Outdoor	4.26	Moderada	Necesita protección, mantenerse bajo sombra. Use camisa, bloqueador solar, sombrero y gafas con protección UV-B y UV-A.
3 - Outdoor	4.54	Moderada	
4 - Outdoor	7.88	Alta	

4.1.4. RUIDO

Para la interpretación de este parámetro ambiental se utilizará la Imagen 58

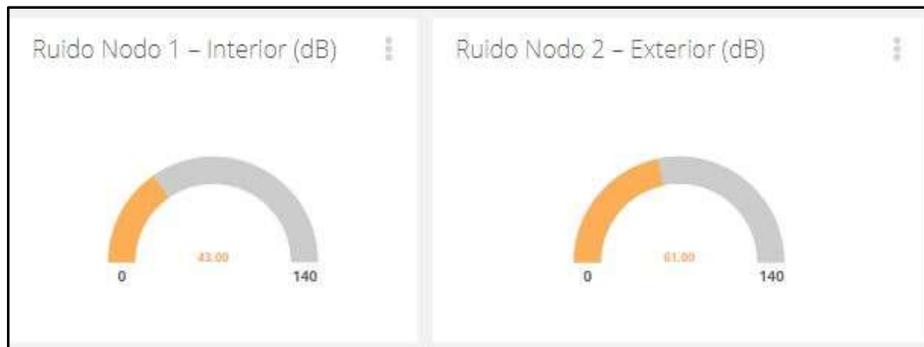


Imagen 53. Ruido, nodo 1 y 2 obtenidas de Ubidots [Autor]

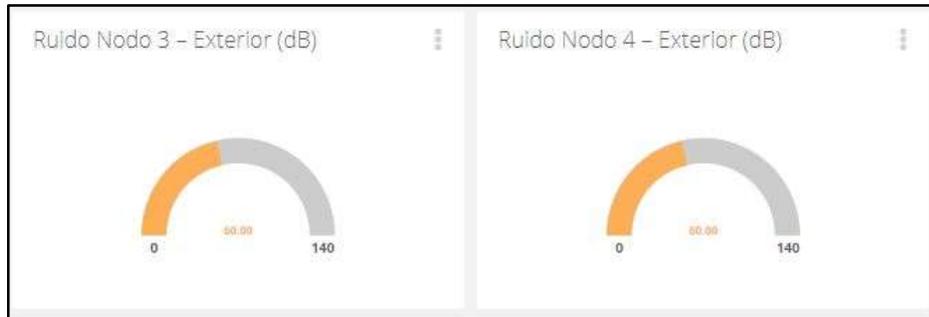


Imagen 54. Ruido, nodo 3 y 4 obtenidas de Ubidots [Autor]

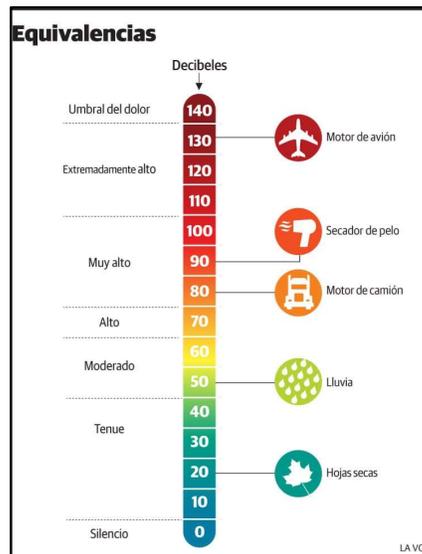


Imagen 55. Equivalencias de ruido (dB) [51].

Tabla 12 Interpretación del nivel de ruido [Autor].

Nodo	Ruido [dB]	Nivel de Ruido
1 - Indoor	43	Tenue
2 - Outdoor	61	Moderado
3 - Outdoor	60	Moderado
4 - Outdoor	60	Moderado

4.2. ANÁLISIS DE DATOS DEL MATERIAL PARTICULADO (PM 2.5 Y PM 10)

La recopilación de datos por el sensor Plantower modelo PMS3003 fue realizado mediante un muestreo con intervalos de 5 minutos durante 24 horas. El nodo se encontró operativo por el periodo del 28 de marzo al 26 de abril de 2020 [30 días] obteniendo información de manera continua.

Es de indicar, para el análisis de los datos que se obtuvieron referente a los niveles de material particulado tomaremos en cuenta siete días [23 al 29 de marzo] debido a que en ese periodo no se reflejaron precipitaciones que afecten a nuestros resultados.

Las mediciones se realizaron en tres puntos, el más alejado se encuentra a una distancia aproximada de 600 metros desde el nodo Gateway en línea recta. Todos los puntos de medición están ubicados en un entorno similar referente al grado de urbanización. A continuación, se presenta una imagen satelital de las ubicaciones.



Imagen 56. Ubicación satelital de nodos de medición en urbanización privada [52]

Los nodos se encuentran ubicados en el Oeste de la urbe y tienen como principales características físicas: la cercanía a las avenidas con mayor flujo vehicular, a un cuerpo de agua y a varias industrias que se dedican a la explotación mineral.

Para obtener una representación de datos comparables, se eligió una hora referencial comprendida entre las 11:00 AM a 13:00 PM. A la cual se calculó el promedio y se realizó una comparación entre el promedio por el periodo de 24hs de medición.

La justificación a la elección horaria se encuentra en uno de los factores más predominantes en la toma de lecturas, la radiación solar, la cual es mayor en las horas circundantes al mediodía. Además, es un fenómeno incidente de la estabilidad

atmosférica, al ser elevada la cantidad de radiación se propicia la inestabilidad en la atmósfera, lo que favorece al transporte de contaminantes.

4.2.1. ANÁLISIS COMPARATIVO DE MEDICIONES

Se presenta la comparación del promedio de las mediciones realizadas dentro de las 24 horas y las captadas durante las horas de referencia.

El muestreo se realizó en un lapso de tiempo en el cual no se presentaron perturbaciones fuera de los niveles máximos de lectura especificados de acuerdo a los parámetros nacionales otorgados por el Ministerio del Ambiente que indican valores máximos permitidos para PM 2,5 de 50 $\mu\text{g}/\text{m}^3$ y para PM 10 de 100 $\mu\text{g}/\text{m}^3$ con referencia a la concentración de particulado, concluyendo así que todas las mediciones muestran un nivel apropiado de contaminación ambiental apto para el ser humano y acorde con la densidad de transporte de contaminantes en perímetros urbanos.

El análisis de datos en promedio de lectura del sensor arroja valores de PM 2,5 y PM 10 correspondientes al intervalo de los siete días de extracción de datos seleccionados.

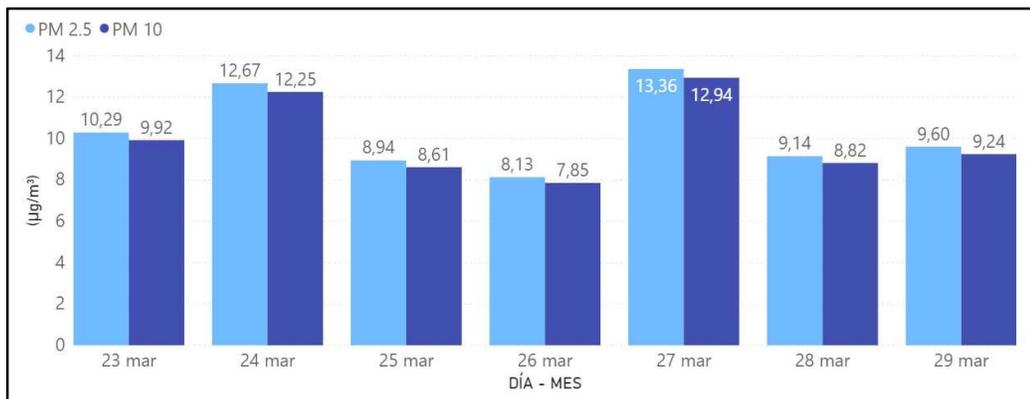


Imagen 57. Promedio semanal de particulado PM 2.5 y PM 10 [Autor].

Se realiza la comparación de datos obteniendo el promedio de la referencia horaria y el promedio diario de la semana de muestreo de los dos materiales particulados.

Tabla 13 Resumen de mediciones realizadas [Autor].

Días	Promedio - PM 2.5		Promedio - PM 10	
	Referencia horaria	24 horas	Referencia horaria	24 horas
23/03	10,0	10,30	9,68	9,90
24/03	11,72	12,70	11,24	12,30
25/03	5,68	8,90	5,60	8,60
26/03	4,72	8,10	4,40	7,90
27/03	12,32	13,40	11,64	12,90
28/03	8,88	9,10	8,56	8,80
29/03	10,16	9,60	9,84	9,20
Referencia Límite MAE	-	50,0	-	100,0

Cabe mencionar, que la recopilación de datos se realizó en condiciones extraordinarias por el semáforo rojo implementado en la ciudad de Guayaquil como medida de la emergencia sanitaria y por ende se observó tendencia de bajas mediciones.

4.2.2. ALCANCE - ANÁLISIS DE MATERIAL PARTICULADO

Con fecha 20 de mayo del 2020 la ciudad de Guayaquil pasó a semáforo amarillo, por tal motivo el 11 de julio del 2020 durante las 09:00 a 16:00 horas, se procedió a capturar datos a fin de determinar los niveles de material particulado con una mayor afluencia de transeúntes y tráfico vehicular en comparación al análisis realizado durante el semáforo rojo.

A efectos de estudio, se instaló un nodo sensor outdoor cerca de la Av. José Rodríguez Bonín, la cual se seleccionó debido a su alto nivel de afluencia y a la cercanía del domicilio de uno de los autores del presente proyecto.



Imagen 58. Imagen satelital de la localización del nodo sensor outdoor [52]



Imagen 59. Instalación de nodo sensor outdoor [Autor]

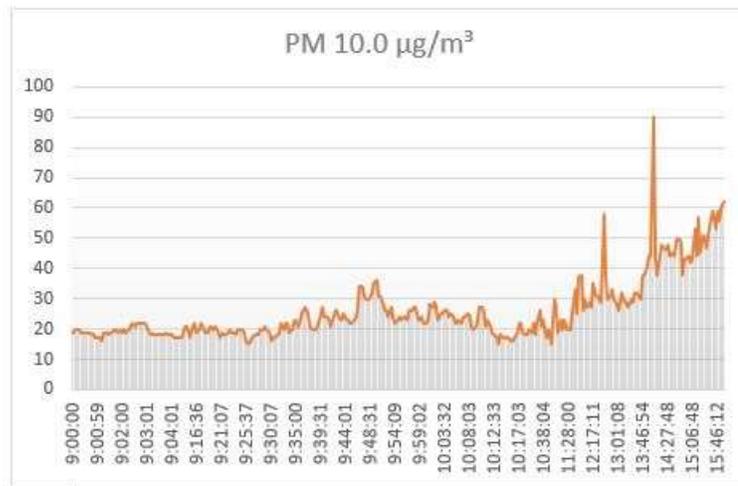


Imagen 60. Registro de datos PM 10 $\mu\text{g}/\text{m}^3$ [Autor]

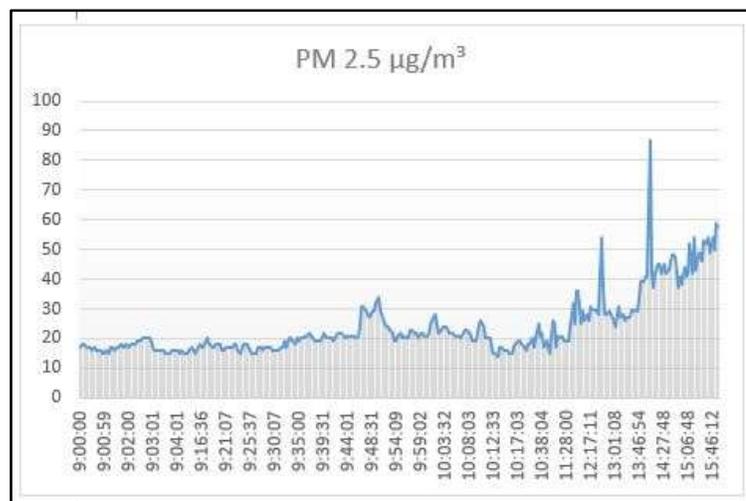


Imagen 61. Registro de datos PM 2.5 $\mu\text{g}/\text{m}^3$ [Autor]

Por el periodo seleccionado se determinó un promedio de 23.87 $\mu\text{g}/\text{m}^3$ por PM 2.5 y 25.99 $\mu\text{g}/\text{m}^3$ por PM 10, teniendo un pico máximo a las 14:04:10 de 87 $\mu\text{g}/\text{m}^3$ de PM 2.5 y 90 $\mu\text{g}/\text{m}^3$ de PM 10. Se concluye que la tendencia de concentración de partículas está dentro del rango moderado; sin embargo, a partir de las 14:00 horas se observó un incremento en los niveles de concentración de partículas que causarían daños a la salud por exposiciones prolongadas de tiempo.

4.3. COMPARATIVA DE DATOS OBTENIDOS EN LOS NODOS SENSORES VS ESTACIÓN METEOROLÓGICA

Para llevar a cabo esta comparación se realizó un seguimiento diario a las variables de los nodos sensores y a la página oficial de la Agencia espacial civil ecuatoriana (<http://exa.ec/>) por el periodo comprendido entre el 28 de marzo y 26 de abril de 2020, en el cual se obtuvieron los valores máximos por cada día de las variables temperatura y humedad.



Imagen 62. Temperatura y humedad reportado por EXA [53]

A fin de visualizar la comparación se elaboraron los siguientes gráficos:

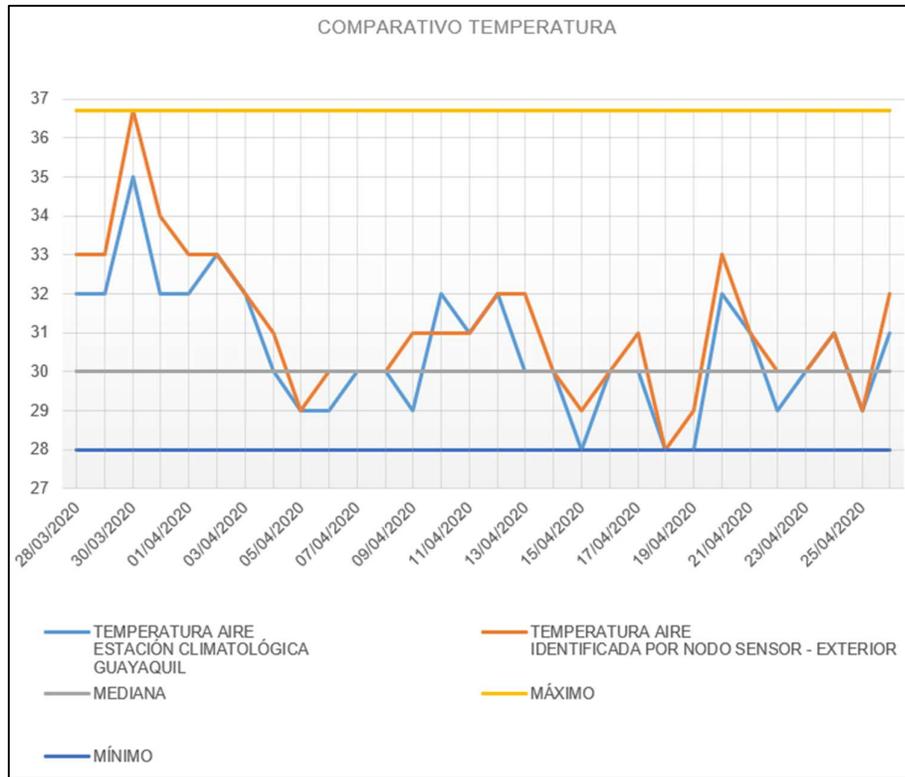


Imagen 63. Temperatura nodo sensor vs estación climatológica Guayaquil [Autor].

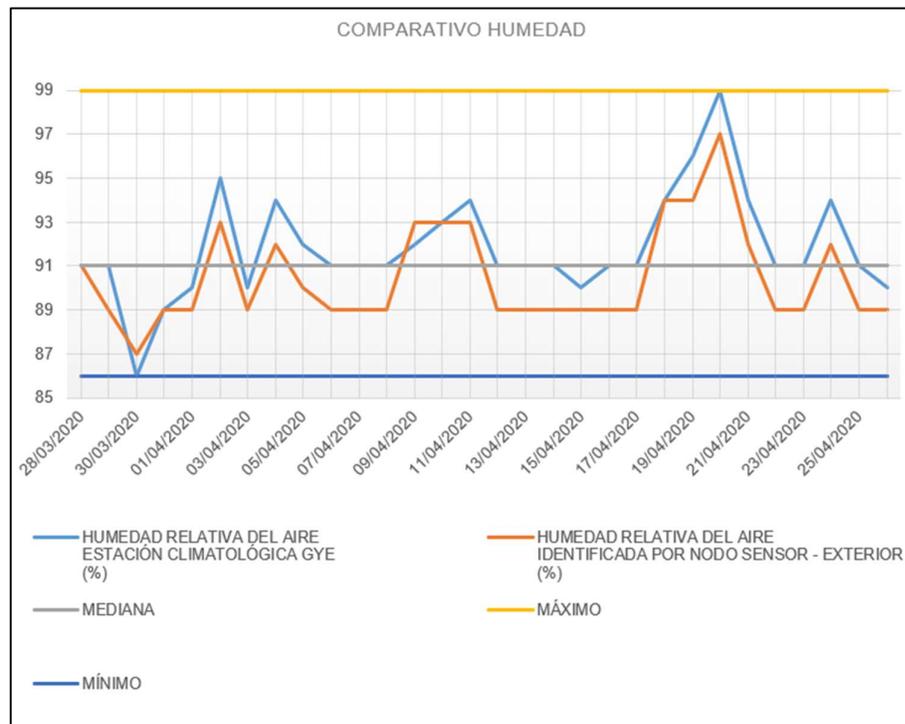


Imagen 64. Humedad nodo sensor vs estación climatológica Guayaquil [Autor].

Las mediciones efectuadas por los nodos de temperatura y humedad durante el periodo de tiempo analizado tienen un comportamiento y tendencia similar, lo cual se puede corroborar en las siguientes tablas:

Tabla 14 Resumen medición variable Temperatura [Autor].

FECHA HORA (GMT-5)	TEMPERATURA AIRE ESTACIÓN CLIMATOLÓGICA GUAYAQUIL	TEMPERATURA AIRE IDENTIFICADA POR NODO SENSOR - EXTERIOR	ERROR ABSOLUTO	ERROR RELATIVO
28/03/2020	32	33	1	3.1%
29/03/2020	32	33	1	3.1%
30/03/2020	35	36.7	1.7	4.9%
31/03/2020	32	34	2	6.3%
01/04/2020	32	33	1	3.1%
02/04/2020	33	33	0	0.0%
03/04/2020	32	32	0	0.0%
04/04/2020	30	31	1	3.3%
05/04/2020	29	29	0	0.0%
06/04/2020	29	30	1	3.4%
07/04/2020	30	30	0	0.0%
08/04/2020	30	30	0	0.0%
09/04/2020	29	31	2	6.9%
10/04/2020	32	31	-1	-3.1%
11/04/2020	31	31	0	0.0%
12/04/2020	32	32	0	0.0%
13/04/2020	30	32	2	6.7%
14/04/2020	30	30	0	0.0%
15/04/2020	28	29	1	3.6%
16/04/2020	30	30	0	0.0%
17/04/2020	30	31	1	3.3%
18/04/2020	28	28	0	0.0%
19/04/2020	28	29	1	3.6%
20/04/2020	32	33	1	3.1%
21/04/2020	31	31	0	0.0%
22/04/2020	29	30	1	3.4%
23/04/2020	30	30	0	0.0%
24/04/2020	31	31	0	0.0%
25/04/2020	29	29	0	0.0%
26/04/2020	31	32	1	3.2%
DESVIACIÓN ESTÁNDAR	1.63	1.83		2.4%
MEDIANA	30	31		1.6%
MÁXIMO	35	37		
MÍNIMO	28	28		

Tabla 15 Resumen medición variable Humedad [Autor].

FECHA HORA (GMT-5)	HUMEDAD RELATIVA DEL AIRE ESTACIÓN CLIMATOLÓGICA GYE (%)	HUMEDAD RELATIVA DEL AIRE IDENTIFICADA POR NODO SENSOR - EXTERIOR (%)	ERROR ABSOLUTO	ERROR RELATIVO
28/03/2020	91	91	0.0	0.0%
29/03/2020	91	89	2.0	2.2%
30/03/2020	86	87	1.0	1.2%
31/03/2020	89	89	0.0	0.0%
01/04/2020	90	89	1.0	1.1%
02/04/2020	95	93	2.0	2.1%
03/04/2020	90	89	1.0	1.1%
04/04/2020	94	92	2.0	2.1%
05/04/2020	92	90	2.0	2.2%
06/04/2020	91	89	2.0	2.2%
07/04/2020	91	89	2.0	2.2%
08/04/2020	91	89	2.0	2.2%
09/04/2020	92	93	1.0	1.1%
10/04/2020	93	93	0.0	0.0%
11/04/2020	94	93	1.0	1.1%
12/04/2020	91	89	2.0	2.2%
13/04/2020	91	89	2.0	2.2%
14/04/2020	91	89	2.0	2.2%
15/04/2020	90	89	1.0	1.1%
16/04/2020	91	89	2.0	2.2%
17/04/2020	91	89	2.0	2.2%
18/04/2020	94	94	0.0	0.0%
19/04/2020	96	94	2.0	2.1%
20/04/2020	99	97	2.0	2.0%
21/04/2020	94	92	2.0	2.1%
22/04/2020	91	89	2.0	2.2%
23/04/2020	91	89	2.0	2.2%
24/04/2020	94	92	2.0	2.1%
25/04/2020	91	89	2.0	2.2%
26/04/2020	90	89	1.0	1.1%
DESVIACIÓN ESTÁNDAR	2.41	2.27		0.8%
MEDIANA	91	89		2.1%
MÁXIMO	99	97		
MÍNIMO	86	87		

Tomando en cuenta los estimadores por ambos nodos, podemos determinar que los valores no difieren en gran medida entre ambas muestras; además, el error relativo resultante es de 1.6% y 2.1% por los nodos temperatura y humedad, respectivamente. Por lo que podemos concluir que los valores registrados por los nodos sensores son precisos y confiables dado que el nivel de error es bajo.

CONCLUSIONES

Se diseñó y desarrolló una red de sensores IoT utilizando tecnología LoRaWAN para el monitoreo de parámetros ambientales tanto en interiores y exteriores y por medio de una interfaz gráfica, la cual se puede acceder desde cualquier dispositivo electrónico con acceso a internet para visualizar los datos obtenidos por los sensores.

La implementación de este trabajo de titulación es de utilidad en el ámbito de investigación ambiental puesto que a través de las mediciones de los nodos sensores se va generar un histórico que sirva como antecedente para futuros estudios, tomando en cuenta que la base de datos será accesible de manera gratuita.

En la ciudad de Guayaquil no se identificó entidades públicas que ofrezcan información histórica de variables ambientales disponible para la ciudadanía; no obstante, mediante la escasa información publicada, de humedad y temperatura, se realizó la comparación contra los registros adquiridos por los nodos sensores.

Tras el análisis comparativo de las variables humedad y temperatura en la Tabla 14 y 15, se concluyó que los registros son confiables puesto que, tienen un margen de error bajo del 2.1% y 1.6%, respectivamente.

Posterior al análisis de las partículas PM 2.5 y PM 10 realizadas durante la emergencia sanitaria en la transición del semáforo rojo a amarillo establecido por el Comité de Operaciones de Emergencia (COE), se identificó un notable incremento en los niveles de concentración de material particulado, por lo que podemos corroborar que el aumento de afluencia de transeúntes y tráfico vehicular incide negativamente en la calidad del aire.

RECOMENDACIONES

- Con el fin de tener una óptima comunicación de largo alcance entre la el nodo Gateway y los dispositivos finales, se recomienda utilizar antenas que trabajen en la misma frecuencia y con una ganancia no menos de 8dBi.
- Tener en cuenta que el nodo Gateway debe recibir una buena conexión a internet para una eficiente enrutamiento de paquetes.
- Para una mayor eficiencia en la batería de los nodos sensores, se aconseja programar el arduino con un tiempo de envío de paquetes por cada 5 minutos.
- Debido a que se necesita una carga inicial antes del uso de los nodos sensores, se recomienda cargar las mismas utilizando el puerto usb durante 1 hora.
- Para validar el correcto envío de paquetes hacia el Gateway, se debe usar el monitor serie de la aplicación Arduino.

BIBLIOGRAFÍA.

- [1] Universidad de Barcelona, «OBS Business School,» [En línea]. Available: <https://www.obs-edu.com/int/noticias/estudio-obs/en-2020-mas-de-30-mil-millones-de-dispositivos-estaran-conectados-internet>. [Último acceso: 14 05 2019].
- [2] «Tecnología de comunicación para IoT,» [En línea]. Available: <https://www.efor.es/sites/default/files/tecnologias-de-comunicacion-para-iot.pdf>. [Último acceso: 12 05 2019].
- [3] B. G. González Romero, «Monitoreo ambiental con redes de sensores inalámbricos basados en microcontroladores de 32 bits y módulos RF, aplicado a un parqueadero cerrado,» de *Aplicaciones de la red de sensores inalámbricos*, Guayaquil, UCSG, 2019, p. 13.
- [4] Organización Panamericana de la Salud, «Contaminación del Aire Ambiental,» Pan American Health Organization, [En línea]. Available: https://www.paho.org/hq/index.php?option=com_content&view=article&id=12918:ambien-air-pollution&Itemid=72243&lang=es. [Último acceso: 20 01 2019].
- [5] «Tendencias científicas,» 8 05 2018. [En línea]. Available: https://www.tendencias21.net/La-OMS-alerta-de-la-contaminacion-ambiental_a44531.html. [Último acceso: 2 12 2018].
- [6] Ministerio del Ambiente, «PLAN NACIONAL DE LA CALIDAD DEL AIRE,» de *DIAGNÓSTICO SOBRE LA CALIDAD DE AIRE EN ECUADOR*, QUITO, Ministerio del Ambiente, 2010, pp. 1- 4.
- [7] El Universo, «Ruido, un mal latente en comercio y barrios de Guayaquil,» p. 25, 19 05 2018.
- [8] Ministerio de Ambiente, «Ecuador le dice ¡NO AL RUIDO!,» [En línea]. Available: <http://www.ambiente.gob.ec/hoy-ecuador-le-dice-no-al-ruido/>. [Último acceso: 10 02 2019].

- [9] C. Katherine y S. Daniela, «Análisis de rendimiento de la tecnología lorawan aplicado al desarrollo de un sistema de monitoreo de calidad de aire en la unidad educativa santo tomás apóstol Riobamba,» de *Red de monitoreo de la calidad del aire*, Riobamba, ESPOCH, 2019, p. 34.
- [10] Universidad Miguel Hernandes de Elche, «Prevención de riesgos laborales,» 2009. [En línea]. Available: <http://prevencion.umh.es/files/2011/10/humedad-y-contaminacion-espacios-interiores.pdf>. [Último acceso: 12 03 2019].
- [11] L. Iacono, P. Godoy, O. Marianetti, C. G. Garino y C. Párraga, «Estudio de la Integración entre WSN y redes TCP/IP,» *DIALNET*, nº 10, pp. 57-68, 2012.
- [12] F. S. Diana Archila, «Estado del arte de las redes de sensores inalámbricos,» *Revista digital TIA*, vol. 2, nº 1, p. 7, 2013.
- [13] Universidad de Valencia, «Redes de comunicaciones: Topologías y enlaces,» Universidad de Valencia, 2018.
- [14] Aprendiendo Arduino, «Topología Estrella,» [En línea]. Available: https://i0.wp.com/www.digi.com/resources/documentation/Digidocs/90001456-13/resources/images/rf_kits/more_excerpts_38.png. [Último acceso: 15 05 2019].
- [15] Ingeniería Investigación y Tecnología, «Sistema sensor para el monitoreo ambiental basado en redes,» *Redalyc.org*, vol. XVII, nº 2, pp. 211-222, 2015.
- [16] Adafruit, «Sensor VEML6075,» [En línea]. Available: https://cdn-learn.adafruit.com/assets/assets/000/062/575/medium800/adafruit_products_3964_iso_ORIG_2018_09.jpg?1537990292. [Último acceso: 20 05 2019].
- [17] L. Llamas, «BME280,» 15 04 2020. [En línea]. Available: <https://www.luisllamas.es/sensor-ambiental-arduino-bme280/>. [Último acceso: 09 05 2020].

- [18] Adafruit, «Sensor BME280,» [En línea]. Available: <https://www.adafruit.com/product/2652>. [Último acceso: 5 05 2019].
- [19] Plantower, «Sensor PMS3003,» [En línea]. Available: <http://www.plantower.com/en/content/?107.html>. [Último acceso: 15 02 2019].
- [20] M. Camara, «Sensores medioambientales para IoT,» 4 02 2019. [En línea]. Available: <https://www.diarioelectronicohoy.com/sensores-medioambientales-para-iot/>. [Último acceso: 30 11 05].
- [21] Adafruit, «Sensor BM680,» [En línea]. Available: <https://www.adafruit.com/product/3660>. [Último acceso: 12 05 2019].
- [22] Sparkfun , «Sparkfun sound detector,» [En línea]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/5/0/5/12642-01.jpg. [Último acceso: 15 12 2019].
- [23] N. Naik, «LPWAN Technologies for IoT systems: choice between ultra narrow band and spread spectrum,» de *2018 IEEE International Systems Engineering Symposium*, Roma, Italia, 2018.
- [24] L. Krupka, L. Vojtech y M. Neruda, «The issue of LPWAN technology coexistence in IoT,» de *2016 17th International Conference on Mechatronics - Mechatronika (ME)*, Prague, 2016.
- [25] A. Augustin, J. Yi, T. Clausen y W. M. Townsley, «Un estudio de LoRa: redes de largo alcance y baja potencia para Internet de las cosas,» *MDPI*, p. 18, 2016.
- [26] I. Ordóñez Monfort, «Estudio de la arquitectura y el nivel de desarrollo de la red LoRaWAN y de los dispositivos LoRa,» Universitat Oberta de Catalunya, España, 2017.
- [27] I. Ordoñez, «<http://openaccess.uoc.edu/webapps/o2/handle/10609/64365>,» de *Universitat Oberta de Catalunya*, Catalunya, 2017.

- [28] E. J. C. Peñalver, «Análisis y diseño de una red de sensores en un parque natural,» de *Elementos que conforman una red LoRaWAN*, España, Universidad Oberta de Catalunya, 2017, p. 15.
- [29] W. Q. Calcina, «Optimización de la transmisión de paquetes LPWAN/LoraWan,» UNIVERSIDAD PERUANA UNION, Juliaca, 2019.
- [30] A. Casanova, «Digimodes,» 2019. [En línea]. Available: <https://digimodes.wordpress.com/2017/02/18/seguridad-en-redes-lorawan-parte-i/>. [Último acceso: 2 Enero 2019].
- [31] RAK WIRELESS, «LoRa Gateway Rak 831,» [En línea]. Available: <https://doc.rakwireless.com/datasheet/rakproducts/rak831-lorawan-gateway-datasheet>. [Último acceso: 15 01 2020].
- [32] Rack Wireless, «RAK 831 WisNode Lora Module,» [En línea]. Available: https://cdn.shopify.com/s/files/1/0177/8784/6756/products/rak831-1_757x.progressive.png?v=1577338259. [Último acceso: 26 04 2019].
- [33] J. Kim y J. Song, «A Secure Device-to-Device Link Establishment Scheme for LoRaWAN,» *IEEE*, vol. 18, nº 5, pp. 2153 - 2160, 2018.
- [34] C. Luis, «Desarrollo de una plataforma abierta para el almacenamiento y gestión de información de nodos móviles para internet de las cosas de la célula inteligente de la Univesidad Politécnica Salesiana,» Universidad Politecnica Salesiana, Guayaquil, 2016.
- [35] W. Peter, «A definition for Internet of Things,» de *Learning Internet of Things*, BIRMINGHAM , Packt Publishing, 2015, p. 2.
- [36] L. Mainetti, L. Patrono y A. Vilei, «Evolution of Wireless Sensor Networks towards the Internet of Things: a Survey,» de *International Conference on Software, Telecommunications and Computer Networks*, Split, Croacia, 2015.

- [37] M. Freire, «Dspace UDLA,» 2019. [En línea]. Available: <http://dspace.udla.edu.ec/bitstream/33000/10914/1/UDLA-EC-TIRT-2019-01.pdf>. [Último acceso: 16 Febrero 2019].
- [38] J. Lopez y X. Vilajosama, «Universidad Oberta de Catalunya,» 11 06 2017. [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/64286/3/agonzalezgarcia0TFM0617memoria.pdf>. [Último acceso: 15 01 2019].
- [39] L. D. González, Test y despliegue de tecnología de comunicaciones LoRa para aplicaciones de Internet of Things, Madrid: Univesidad Politecnica de Madrid, 2019.
- [40] «Logo The Things Network,» [En línea]. Available: <https://media.mehrnews.com/d/2016/04/30/3/2063559.jpg>. [Último acceso: 15 02 2019].
- [41] B. S. IoT, «IoT Platform,» [En línea]. Available: <https://about.sofia2.com/2018/02/20/que-es-la-iniciativa-the-things-network/>. [Último acceso: 15 06 2019].
- [42] Ubidots.com, [En línea]. Available: https://ubidots.com/_nuxt/img/logo_color.9c7e99e.png. [Último acceso: 24 07 2019].
- [43] Arduino, «¿Qué es arduino?,» [En línea]. Available: <https://www.arduino.cc/en/Guide/Introduction#>. [Último acceso: 14 05 2019].
- [44] Mouser Electronics, «Arduino MKR WAN 1300 (LoRa Connectivity),» [En línea]. Available: <https://www.mouser.ec/new/arduino/arduino-mkr-wan-1300/>. [Último acceso: 25 04 2019].
- [45] Aprendiendo Arduino, «Arduino MKR WAN 1300,» [En línea]. Available: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/500x375/f8876a31b63532bbba4e781c30024a0a/a/b/abx00017_iso.jpg. [Último acceso: 26 04 2019].

- [46] N. Krish, «Hackster.io,» [En línea]. Available: <https://www.hackster.io/naresh-krish/getting-started-with-the-rak831-lora-gateway-and-rpi3-e3351d#things>. [Último acceso: 30 10 2019].
- [47] The Things Network, «Registro de puerta de enlace,» [En línea]. Available: <https://www.thethingsnetwork.org/docs/gateways/registration.html>. [Último acceso: 15 12 2019].
- [48] Agencia Estatal de Meteorología, «MeteoGlosario Visual,» [En línea]. Available: https://meteoglosario.aemet.es/objetos/mediums/867_6613193.jpg. [Último acceso: 30 01 2020].
- [49] Ministerio de Medio Ambiente , «ÍNDICE CENTROAMERICANO DE LA CALIDAD DEL AIRE,» Gobierno de El Salvador, 03 03 2016. [En línea]. Available: https://www.marn.gob.sv/wp-content/uploads/Indice_de_calidad_del_aire_pdf_-_Foxit_Reader.png. [Último acceso: 30 01 2020].
- [50] Mapa de Radiación solar Online, «Radiación Solar,» [En línea]. Available: <http://radiacionsolar.onlinegratis.tv/images/indice-UV-salud.jpg>. [Último acceso: 30 01 2020].
- [51] La Voz , «Un proyecto para bajar los decibeles,» La Voz , 19 06 2018. [En línea]. Available: https://cdn.lavoz.com.ar/sites/default/files/file_attachments/nota_periodistica/Equivalencias.jpg. [Último acceso: 30 01 2020].
- [52] Google Maps, «Ubicación de nodos sensores,» [En línea]. Available: <https://www.google.com.ec/maps/place/Renacer/@-2.1910026,-79.9447233,17z/data=!3m1!4b1!4m5!3m4!1s0x902d718ac68812cd:0x8bfa59b9c083bdb!8m2!3d-2.1910026!4d-79.9425346?hl=es-419&authuser=0>. [Último acceso: 23 05 2020].

- [53] Agencia Espacial Civil Ecuatoriana, «ESTACIÓN CLIMATOLÓGICA GUAYAQUIL: EXA-ISS-1,» [En línea]. Available: <http://gye.exa.ec/>. [Último acceso: 28 03 2020].
- [54] Co-fundador ElectronicCats hardware libre en México, «Medium,» [En línea]. Available: <https://medium.com/beelan/haciendo-iot-con-lora-cap%C3%ADtulo-1-qu%C3%A9-es-lora-y-lorawan-8c08d44208e8>. [Último acceso: 3 01 2019].
- [55] Secretaría de estado de medio ambiente y recursos naturales, «Norma ambiental de calidad del aire,» [En línea]. Available: <http://extwprlegs1.fao.org/docs/pdf/dom60781.pdf>. [Último acceso: 6 05 2019].
- [56] Secretaría de Ambiente, «Norma ecuatoria de calidad del aire,» 2011. [En línea]. Available: http://www.quitoambiente.gob.ec/ambiente/images/Secretaria_Ambiente/red_monitoreo/informacion/norma_ecuato_calidad.pdf. [Último acceso: 21 04 2019].
- [57] Semack , «Logo LoRa,» [En línea]. Available: https://www.semak.com.ar/image/0/0_0-LORA_02%20Lora%20logo.png. [Último acceso: 21 03 2019].
- [58] Shenzhen Rakwireless Technology Co, «LoRA,» [En línea]. Available: https://downloads.rakwireless.com/en/LoRa/RAK811/Hardware_Specificat ion/RAK811_LoRa_Module_Datasheet_V1.4.pdf. [Último acceso: 12 05 2019].

ANEXOS

Anexo 1: Fotos de las pruebas realizadas



Imagen 65. Ubicacion de nodo 1- pruebas [Autor].



Imagen 66. Ubicacion de nodo 2- pruebas [Autor].



Imagen 67. Ubicacion de nodo 3- pruebas [Autor].



Imagen 68. Compilación de código en nodo exterior [Autor].

Anexo 2: Programación arduino – nodo 1

```
const float dBAnalogQuiet = 10;
const float dBAnalogMedium = 12;
const float dBAnalogLoud = 17;
int db;

#include <MKRWAN.h>
#include "arduino_secrets.h"
#include <Wire.h>
#include <SPI.h>
#include "Adafruit_BME680.h"

#define BME_SCK 13
#define BME_MISO 12
#define BME_MOSI 11
#define BME_CS 10
#define PIN_ANALOG_IN A0
#define SEALEVELPRESSURE_HPA (1013.25)

Adafruit_BME680 bme; // I2C
//Adafruit_BME680 bme(BME_CS); // hardware SPI
//Adafruit_BME680 bme(BME_CS, BME_MOSI, BME_MISO, BME_SCK);

float hum_weighting = 0.25; // so hum effect is 25% of the total air quality score
float gas_weighting = 0.75; // so gas effect is 75% of the total air quality score

float hum_score, gas_score;
float gas_reference = 250000;
float hum_reference = 40;
int getgasreference_count = 0;

// Select your region (AS923, AU915, EU868, KR920, IN865, US915,
US915_HYBRID)
_lora_band region = US915;

LoRaModem modem(Serial1);

uint32_t temp,alt,hum,pres,iaq;
byte payload[9];

void setup() {
  Serial1.begin(9600);
  Serial.begin(115200);
```

```

pinMode(0, INPUT);
while (!Serial);
if (!modem.begin(region)) {
  Serial.println("Failed to start module");
  while (1) {}
};
Serial.print("Your device EUI is: ");
Serial.println(modem.deviceEUI());

int connected = modem.joinOTAA(appEui, appKey);
if (!connected) {
  Serial.println("Something went wrong; are you indoor? Move near a window and
retry");
  while (1) {}
}
Serial.println("Successfully joined the network!");

Serial.println("Enabling ADR and setting low spreading factor");
modem.setADR(true);
modem.dataRate(5);
Serial.println(F("BME680 test"));

Wire.begin();
if (!bme.begin()) {
  Serial.println("Could not find a valid BME680 sensor, check wiring!");
  while (1);
} else Serial.println("Found a sensor");

// Set up oversampling and filter initialization
bme.setTemperatureOversampling(BME680_OS_2X);
bme.setHumidityOversampling(BME680_OS_2X);
bme.setPressureOversampling(BME680_OS_2X);
bme.setIIRFilterSize(BME680_FILTER_SIZE_3);
bme.setGasHeater(320, 150); // 320°C for 150 ms
// Now run the sensor for a burn-in period, then use combination of relative humidity
and gas resistance to estimate indoor air quality as a percentage.
  GetGasReference();

}

void bme680() {
  Serial.print("Temperature = ");
  temp=bme.readTemperature()*100;

```

```

Serial.print(bme.readTemperature());
Serial.println("°C");

Serial.print("  Pressure = ");

Serial.print((bme.readPressure() / 100.0F)*0.014503773773);
pres=((bme.readPressure() / 100.0F)*0.014503773773)*100;
Serial.println(" hPa");

Serial.print("  Humidity = ");
hum=bme.readHumidity()*100;
Serial.print(bme.readHumidity());
Serial.println("%");

Serial.print("    Gas = ");
Serial.print(bme.readGas());
Serial.println("R\n");

//Calculate humidity contribution to IAQ index
float current_humidity = bme.readHumidity();
if (current_humidity >= 38 && current_humidity <= 42)
  hum_score = 0.25*100; // Humidity +/-5% around optimum
else
{ //sub-optimal
  if (current_humidity < 38)
    hum_score = 0.25/hum_reference*current_humidity*100;
  else
  {
    hum_score = ((-0.25/(100-hum_reference)*current_humidity)+0.416666)*100;
  }
}

//Calculate gas contribution to IAQ index
float gas_lower_limit = 5000; // Bad air quality limit
float gas_upper_limit = 50000; // Good air quality limit
if (gas_reference > gas_upper_limit) gas_reference = gas_upper_limit;
if (gas_reference < gas_lower_limit) gas_reference = gas_lower_limit;
gas_score = (0.75/(gas_upper_limit-gas_lower_limit)*gas_reference -
(gas_lower_limit*(0.75/(gas_upper_limit-gas_lower_limit))))*100;

//Combine results for the final IAQ index value (0-100% where 100% is good quality
air)
float air_quality_score = hum_score + gas_score;

```

```

Serial.println("Air Quality = "+String(air_quality_score,1)+"% derived from 25% of
Humidity reading and 75% of Gas reading - 100% is good quality air");
Serial.println("Humidity element was : "+String(hum_score/100)+" of 0.25");
Serial.println("  Gas element was : "+String(gas_score/100)+" of 0.75");
if (bme.readGas() < 120000) Serial.println("***** Poor air quality *****");
Serial.println();
if ((getgasreference_count++)%10==0) GetGasReference();
Serial.println(CalculateIAQ(air_quality_score));
Serial.println("-----");
delay(180000);
}

```

```

void GetGasReference(){
  // Now run the sensor for a burn-in period, then use combination of relative humidity
and gas resistance to estimate indoor air quality as a percentage.
  Serial.println("Getting a new gas reference value");
  int readings = 10;
  for (int i = 1; i <= readings; i++){ // read gas for 10 x 0.150mS = 1.5secs
    gas_reference += bme.readGas();
  }
  gas_reference = gas_reference / readings;
}

```

```

String CalculateIAQ(float score){
  String IAQ_text = "Air quality is ";
  score = (100-score)*5;
  iaq=score*100;
  if (score >= 301) IAQ_text += "Hazardous";
  else if (score >= 201 && score <= 300 ) IAQ_text += "Very Unhealthy";
  else if (score >= 176 && score <= 200 ) IAQ_text += "Unhealthy";
  else if (score >= 151 && score <= 175 ) IAQ_text += "Unhealthy for Sensitive
Groups";
  else if (score >= 51 && score <= 150 ) IAQ_text += "Moderate";
  else if (score >= 00 && score <= 50 ) IAQ_text += "Good";
  return IAQ_text;
}

```

```

void decibels(){

  int value;
  float decibelsValueQuiet = 30;
  float decibelsValueMedium = 40;
  float decibelsValueLoud = 60;

```

```

value = analogRead(PIN_ANALOG_IN);

if (value < 10)
{
  decibelsValueQuiet += 30 * log10(value/dBAnalogQuiet);
  db= (int) decibelsValueQuiet;
  Serial.print("decibelios:" +String (decibelsValueQuiet));

}
else if ((value > 10) && ( value <= 30 )
{
  decibelsValueMedium += 10 * log10(value/dBAnalogMedium);
  db= (int) decibelsValueMedium;
  Serial.print("decibelios:" +String (decibelsValueMedium));
}
else if(value > 22)
{
  decibelsValueLoud += log10(value/dBAnalogLoud);
  db = (int) decibelsValueLoud;
  Serial.print("decibelios:" +String (decibelsValueLoud));
}

delay(18000);
}

void loop() {

  modem.beginPacket();
  payload[0]=highByte(temp);
  payload[1]=lowByte(temp);
  payload[2]=highByte(hum);
  payload[3]=lowByte(hum);
  payload[4]=highByte(iaq);
  payload[5]=lowByte(iaq);
  payload[6]=highByte(pres);
  payload[7]=lowByte(pres);
  payload[8]=db;
  modem.write(payload,sizeof(payload));
  //modem.print(pres);
  //modem.print(alt);
  //modem.print(hum);
  //modem.print(radi);
  //modem.print(pmcf10);
  //modem.print(pmcf25);

```

```

//modem.print(pmcfl00);
Serial.println();

decibels();
bme680();

int err = modem.endPacket(false);

if (err > 0) {
  Serial.println("Big success!");
} else {
  Serial.println("Error");
}

delay(18000);
}

```

Anexo 3: Programación arduino – nodo 2

```

const float dBAnalogQuiet = 10;
const float dBAnalogMedium = 12;
const float dBAnalogLoud = 17;
int db;

#include <MKRWAN.h>
#include "arduino_secrets.h"
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include "Adafruit_VEML6075.h"
Adafruit_VEML6075 uv = Adafruit_VEML6075();

// Select your region (AS923, AU915, EU868, KR920, IN865, US915,
US915_HYBRID)
_lora_band region = US915;

LoRaModem modem(Serial1);
#define SEALEVELPRESSURE_HPA (1013.25)
#define PIN_ANALOG_IN A0
Adafruit_BME280 bme;
uint32_t temp,alt,hum,pres,radi;
byte payload[11];

```

```

long pmcf10=0;
long pmcf25=0;
long pmcf100=0;
long pmat10=0;
long pmat25=0;
long pmat100=0;

char buf[50];

void setup() {
  Serial1.begin(9600);
  Serial.begin(115200);
  pinMode(A0, INPUT);
  while (!Serial);
  if (!modem.begin(region)) {
    Serial.println("Failed to start module");
    while (1) {}
  };
  Serial.print("Your device EUI is: ");
  Serial.println(modem.deviceEUI());

  int connected = modem.joinOTAA(appEui, appKey);
  if (!connected) {
    Serial.println("Something went wrong; are you indoor? Move near a window and
retry");
    while (1) {}
  }
  Serial.println("Successfully joined the network!");

  Serial.println("Enabling ADR and setting low spreading factor");
  modem.setADR(true);
  modem.dataRate(5);

  Serial.println(F("BME280 test"));

  unsigned status;

  // default settings
  // (you can also pass in a Wire library object like &Wire2)
  status = bme.begin();
  if (!status) {

```

```

    Serial.println("Could not find a valid BME280 sensor, check wiring, address,
sensor ID!");
    Serial.print("SensorID was: 0x"); Serial.println(bme.sensorID(),16);
    Serial.print("    ID of 0xFF probably means a bad address, a BMP 180 or BMP
085\n");
    Serial.print(" ID of 0x56-0x58 represents a BMP 280,\n");
    Serial.print("    ID of 0x60 represents a BME 280.\n");
    Serial.print("    ID of 0x61 represents a BME 680.\n");
    while (1);
}

Serial.println("-- Default Test --");

Serial.println("VEML6075 Simple Test");

if (! uv.begin()) {
    Serial.println("Failed to communicate with VEML6075 sensor, check wiring?");
    while (1) { delay(100); }
}
Serial.println("Found VEML6075 sensor");

}

void pm30003() {
// put your main code here, to run repeatedly:
int count = 0;
unsigned char c;
unsigned char high;

while (Serial1.available()) {
    c = Serial1.read();
    if((count==0 && c!=0x42) || (count==1 && c!=0x4d)){
        Serial.println("check failed");
        break;
    }
    if(count > 15){
        //Serial.println("complete");
        break;
    }
    else if(count == 4 || count == 6 || count == 8 || count == 10 || count == 12 || count ==
14) {
        high = c;
    }
    else if(count == 5){

```

```

    pmcf10 = 256*high + c;
    Serial.print("CF=1, PM1.0=");
    Serial.print(pmcf10);
    Serial.println(" ug/m3");
}
else if(count == 7){
    pmcf25 = 256*high + c;
    Serial.print("CF=1, PM2.5=");
    Serial.print(pmcf25);
    Serial.print(" ug/m3");
}
else if(count == 9){
    pmcf100 = 256*high + c;
    // Serial.print("CF=1, PM10=");
    // Serial.print(pmcf100);
    // Serial.println(" ug/m3");
}
else if(count == 11){
    pmat10 = 256*high + c;
    // Serial.print("atmosphere, PM1.0=");
//Serial.print(pmat10);
    // Serial.println(" ug/m3");
}
else if(count == 13){
    pmat25 = 256*high + c;
    // Serial.print("atmosphere, PM2.5=");
    // Serial.print(pmat25);
    // Serial.println(" ug/m3");
}
else if(count == 15){
    pmat100 = 256*high + c;
    // Serial.print("atmosphere, PM10=");
    // Serial.print(pmat100);
    // Serial.println(" ug/m3");
}
    count++;
}
while(Serial1.available()) Serial1.read();
Serial.println();
}

void sensor_veml_6075(){
    Serial.print("UV Index reading: "); Serial.println(uv.readUVI());
    radi=(uv.readUVI()*100);
}

```

```

//Serial.println();
}

void bme280() {
  Serial.print("Temperature = ");
  Serial.print(bme.readTemperature());
  Serial.println(" *C");

  Serial.print("Pressure = ");

  Serial.print((bme.readPressure() / 100.0F)*0.014503773773);
  Serial.println(" psi");

  //Serial.print("Approx. Altitude = ");
  // Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));
  //Serial.println(" m");

  Serial.print("Humidity = ");
  Serial.print(bme.readHumidity());
  Serial.print(" %");

  Serial.println();
  temp=bme.readTemperature()*100;
  pres=((bme.readPressure() / 100.0F)*0.014503773773)*100;
  alt=(bme.readAltitude(SEALEVELPRESSURE_HPA))*100;
  hum=(bme.readHumidity()*100);
}

void decibels(){

  int value;
  float decibelsValueQuiet = 30;
  float decibelsValueMedium = 40;
  float decibelsValueLoud = 60;

  value = analogRead(PIN_ANALOG_IN);

  if (value < 10)
  {
    decibelsValueQuiet += 30 * log10(value/dBAnalogQuiet);
    db= (int) decibelsValueQuiet;
    Serial.print("decibelios:" +String (decibelsValueQuiet));
  }
}

```

```

    }
    else if ((value > 10) && ( value <= 30 )
    {
        decibelsValueMedium += 10 * log10(value/dBAnalogMedium);
        db= (int) decibelsValueMedium;
        Serial.print("decibelios:" +String (decibelsValueMedium));
    }
    else if(value > 22)
    {
        decibelsValueLoud += log10(value/dBAnalogLoud);
        db = (int) decibelsValueLoud;
        Serial.print("decibelios:" +String (decibelsValueLoud));
    }

    delay(15000);
}

void loop() {
    modem.beginPacket();
    payload[0]=highByte(temp);
    payload[1]=lowByte(temp);
    payload[2]=highByte(hum);
    payload[3]=lowByte(hum);
    payload[4]=highByte(radi);
    payload[5]=lowByte(radi);
    payload[6]=highByte(pres);
    payload[7]=lowByte(pres);
    payload[8]=pmcf25;
    payload[9]=pmcf100;
    payload[10]=db;
    modem.write(payload,sizeof(payload));
    //modem.print(pres);
    //modem.print(alt);
    //modem.print(hum);
    //modem.print(radi);
    //modem.print(pmcf10);
    //modem.print(pmcf25);
    //modem.print(pmcf100);
    Serial.println();
    bme280();
    sensor_veml_6075();
    pm30003();
    decibels();
}

```

```

int err = modem.endPacket(false);

if (err > 0) {
  Serial.println("Big success!");
} else {
  Serial.println("Error");
}
delay(150000);
}

```

Anexo 4: Programación arduino – nodo 3

```

const float dBAnalogQuiet = 10;
const float dBAnalogMedium = 12;
const float dBAnalogLoud = 17;
int db;

```

```

#include <MKRWAN.h>
#include "arduino_secrets.h"
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include "Adafruit_VEML6075.h"
Adafruit_VEML6075 uv = Adafruit_VEML6075();

```

```

// Select your region (AS923, AU915, EU868, KR920, IN865, US915,
US915_HYBRID)
_lora_band region = US915;

```

```

LoRaModem modem(Serial1);
#define SEALEVELPRESSURE_HPA (1013.25)
#define PIN_ANALOG_IN A0
uint32_t temp,alt,hum,pres,radi;
byte payload[11];

```

```

long pmcf10=0;
long pmcf25=0;
Adafruit_BME280 bme;
long pmcf100=0;
long pmat10=0;
long pmat25=0;

```

```

long pmat100=0;

char buf[50];

void setup() {
  Serial1.begin(9600);
  Serial.begin(115200);
  pinMode(A0, INPUT);
  while (!Serial);
  if (!modem.begin(region)) {
    Serial.println("Failed to start module");
    while (1) {}
  };
  Serial.print("Your device EUI is: ");
  Serial.println(modem.deviceEUI());

  int connected = modem.joinOTAA(appEui, appKey);
  if (!connected) {
    Serial.println("Something went wrong; are you indoor? Move near a window and
retry");
    while (1) {}
  }
  Serial.println("Successfully joined the network!");

  Serial.println("Enabling ADR and setting low spreading factor");
  modem.setADR(true);
  modem.dataRate(5);

  Serial.println(F("BME280 test"));

  unsigned status;

  // default settings
  // (you can also pass in a Wire library object like &Wire2)
  status = bme.begin();
  if (!status) {
    Serial.println("Could not find a valid BME280 sensor, check wiring, address,
sensor ID!");
    Serial.print("SensorID was: 0x"); Serial.println(bme.sensorID(),16);
    Serial.print("    ID of 0xFF probably means a bad address, a BMP 180 or BMP
085\n");
    Serial.print("    ID of 0x56-0x58 represents a BMP 280,\n");
    Serial.print("    ID of 0x60 represents a BME 280.\n");
  }
}

```

```

        Serial.print("    ID of 0x61 represents a BME 680.\n");
        while (1);
    }

    Serial.println("-- Default Test --");

Serial.println("VEML6075 Simple Test");

if (! uv.begin()) {
    Serial.println("Failed to communicate with VEML6075 sensor, check wiring?");
    while (1) { delay(4000); }
}
Serial.println("Found VEML6075 sensor");

}

void pm30003() {
    // put your main code here, to run repeatedly:
    int count = 0;
    unsigned char c;
    unsigned char high;

    while (Serial1.available()) {
        c = Serial1.read();
        if((count==0 && c!=0x42) || (count==1 && c!=0x4d)){
            Serial.println("check failed");
            break;
        }
        if(count > 15){
            //Serial.println("complete");
            break;
        }
        else if(count == 4 || count == 6 || count == 8 || count == 10 || count == 12 || count ==
14) {
            high = c;
        }
        else if(count == 5){
            pmcf10 = 256*high + c;
            Serial.print("CF=1, PM1.0=");
            Serial.print(pmcf10);
            Serial.println(" ug/m3");
        }
        else if(count == 7){

```

```

    pmcf25 = 256*high + c;
    Serial.print("CF=1, PM2.5=");
    Serial.print(pmcf25);
    Serial.print(" ug/m3");
}
else if(count == 9){
    pmcf100 = 256*high + c;
    // Serial.print("CF=1, PM10=");
    // Serial.print(pmcf100);
    // Serial.println(" ug/m3");
}
else if(count == 11){
    pmat10 = 256*high + c;
    // Serial.print("atmosphere, PM1.0=");
//Serial.print(pmat10);
    // Serial.println(" ug/m3");
}
else if(count == 13){
    pmat25 = 256*high + c;
    // Serial.print("atmosphere, PM2.5=");
    // Serial.print(pmat25);
    // Serial.println(" ug/m3");
}
else if(count == 15){
    pmat100 = 256*high + c;
    // Serial.print("atmosphere, PM10=");
    // Serial.print(pmat100);
    // Serial.println(" ug/m3");
}
    count++;
}
while(Serial1.available()) Serial1.read();
Serial.println();
}

```

```

void sensor_veml_6075(){
    Serial.print("UV Index reading: "); Serial.println(uv.readUVI());
    radi=(uv.readUVI()*100;
    //Serial.println();
}

```

```

void bme280() {

```

```

Serial.print("Temperature = ");
Serial.print(bme.readTemperature());
Serial.println(" *C");

Serial.print("Pressure = ");

Serial.print((bme.readPressure() / 100.0F)*0.014503773773);
Serial.println(" psi");

//Serial.print("Approx. Altitude = ");
// Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));
//Serial.println(" m");

Serial.print("Humidity = ");
Serial.print(bme.readHumidity());
Serial.print(" %");

Serial.println();
temp=bme.readTemperature()*100;
pres=((bme.readPressure() / 100.0F)*0.014503773773)*100;
alt=(bme.readAltitude(SEALEVELPRESSURE_HPA))*100;
hum=(bme.readHumidity()*100);
}

void decibels(){

int value;
float decibelsValueQuiet = 30;
float decibelsValueMedium = 40;
float decibelsValueLoud = 60;

value = analogRead(PIN_ANALOG_IN);

if (value < 10)
{
decibelsValueQuiet += 30 * log10(value/dBAnalogQuiet);
db= (int) decibelsValueQuiet;
Serial.print("decibelios:" +String (decibelsValueQuiet));

}
else if ((value > 10) && ( value <= 30) )
{
decibelsValueMedium += 10 * log10(value/dBAnalogMedium);
}
}

```

```

    db= (int) decibelValueMedium;
    Serial.print("decibelios:" +String (decibelValueMedium));
}
else if(value > 22)
{
    decibelValueLoud += log10(value/dBAnalogLoud);
    db = (int) decibelValueLoud;
    Serial.print("decibelios:" +String (decibelValueLoud));
}

delay(10000);
}

```

```

void loop() {
    modem.beginPacket();
    payload[0]=highByte(temp);
    payload[1]=lowByte(temp);
    payload[2]=highByte(hum);
    payload[3]=lowByte(hum);
    payload[4]=highByte(radi);
    payload[5]=lowByte(radi);
    payload[6]=highByte(pres);
    payload[7]=lowByte(pres);
    payload[8]=pmcf25;
    payload[9]=pmcf100;
    payload[10]=db;
    modem.write(payload,sizeof(payload));
    //modem.print(pres);
    //modem.print(alt);
    //modem.print(hum);
    //modem.print(radi);
    //modem.print(pmcf10);
    //modem.print(pmcf25);
    //modem.print(pmcf100);
    Serial.println();
    bme280();
    sensor_veml_6075();
    pm30003();
    decibels();

    int err = modem.endPacket(false);

    if (err > 0) {
        Serial.println("Big success!");
    }
}

```

```
    } else {  
      Serial.println("Error");  
    }  
    delay(18000);  
  }  
}
```

Anexo 4: Programación arduino – nodo 4

```
const float dBAnalogQuiet = 10;  
  
const float dBAnalogMedium = 12;  
  
const float dBAnalogLoud = 17;  
  
int db;
```

```
#include <MKRWAN.h>
```

```
#include "arduino_secrets.h"
```

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

```
#include <SPI.h>
```

```
#include <Adafruit_Sensor.h>
```

```
#include <Adafruit_BME280.h>
```

```
#include "Adafruit_VEML6075.h"
```

```
Adafruit_VEML6075 uv = Adafruit_VEML6075();
```

```
// Select your region (AS923, AU915, EU868, KR920, IN865, US915,  
US915_HYBRID)
```

```
_lora_band region = US915;
```

```
LoRaModem modem(Serial1);
```

```
#define SEALEVELPRESSURE_HPA (1013.25)
```

```

#define PIN_ANALOG_IN A0

Adafruit_BME280 bme;

uint32_t temp,alt,hum,pres,radi;

byte payload[11];

long pmcf10=0;

long pmcf25=0;

long pmcf100=0;

long pmat10=0;

long pmat25=0;

long pmat100=0;

char buf[50];

void setup() {

  Serial1.begin(9600);

  Serial.begin(115200);

  pinMode(A0, INPUT);

  while (!Serial);

  if (!modem.begin(region)) {

    Serial.println("Failed to start module");

    while (1) {}

  };

  Serial.print("Your device EUI is: ");

  Serial.println(modem.deviceEUI());

```

```

int connected = modem.joinOTAA(appEui, appKey);

if (!connected) {

    Serial.println("Something went wrong; are you indoor? Move near a window and
retry");

    while (1) {}

}

Serial.println("Successfully joined the network!");

Serial.println("Enabling ADR and setting low spreading factor");

modem.setADR(true);

modem.dataRate(5);

Serial.println(F("BME280 test"));

unsigned status;

// default settings

// (you can also pass in a Wire library object like &Wire2)

status = bme.begin();

if (!status) {

    Serial.println("Could not find a valid BME280 sensor, check wiring, address,
sensor ID!");

    Serial.print("SensorID was: 0x"); Serial.println(bme.sensorID(),16);

    Serial.print("    ID of 0xFF probably means a bad address, a BMP 180 or BMP
085\n");
}

```

```

    Serial.print(" ID of 0x56-0x58 represents a BMP 280,\n");
    Serial.print(" ID of 0x60 represents a BME 280.\n");
    Serial.print(" ID of 0x61 represents a BME 680.\n");
    while (1);
}

Serial.println("-- Default Test --");

Serial.println("VEML6075 Simple Test");
if (! uv.begin()) {
    Serial.println("Failed to communicate with VEML6075 sensor, check wiring?");
    while (1) { delay(100); }
}
Serial.println("Found VEML6075 sensor");
}

void pm30003() {
    // put your main code here, to run repeatedly:

    int count = 0;
    unsigned char c;
    unsigned char high;

    while (Serial1.available()) {
        c = Serial1.read();
        if((count==0 && c!=0x42) || (count==1 && c!=0x4d)){
            Serial.println("check failed");
            break;

```

```

}

if(count > 15){
    //Serial.println("complete");

    break;
}

else if(count == 4 || count == 6 || count == 8 || count == 10 || count == 12 || count ==
14) {

    high = c;

}

else if(count == 5){

    pmcf10 = 256*high + c;

    Serial.print("CF=1, PM1.0=");

    Serial.print(pmcf10);

    Serial.println(" ug/m3");

}

else if(count == 7){

    pmcf25 = 256*high + c;

    Serial.print("CF=1, PM2.5=");

    Serial.print(pmcf25);

    Serial.print(" ug/m3");

}

else if(count == 9){

    pmcf100 = 256*high + c;

    // Serial.print("CF=1, PM10=");

    // Serial.print(pmcf100);

```

```

    // Serial.println(" ug/m3");
}
else if(count == 11){
    pmat10 = 256*high + c;
    // Serial.print("atmosphere, PM1.0=");
//Serial.print(pmat10);
    // Serial.println(" ug/m3");
}
else if(count == 13){
    pmat25 = 256*high + c;
    // Serial.print("atmosphere, PM2.5=");
    // Serial.print(pmat25);
    // Serial.println(" ug/m3");
}
else if(count == 15){
    pmat100 = 256*high + c;
    // Serial.print("atmosphere, PM10=");
    // Serial.print(pmat100);
    // Serial.println(" ug/m3");
}
count++;
}
while(Serial1.available()) Serial1.read();
Serial.println();

```

```

}

void sensor_veml_6075(){

  Serial.print("UV Index reading: "); Serial.println(uv.readUVI());

  radi=(uv.readUVI()*100;

  //Serial.println();

}

void bme280() {

  Serial.print("Temperature = ");

  Serial.print(bme.readTemperature());

  Serial.println(" *C");

  Serial.print("Pressure = ");

  Serial.print((bme.readPressure() / 100.0F)*0.014503773773);

  Serial.println(" psi");

  //Serial.print("Approx. Altitude = ");

  // Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));

  //Serial.println(" m");

  Serial.print("Humidity = ");

  Serial.print(bme.readHumidity());

  Serial.print(" %");

```

```

Serial.println();

temp=bme.readTemperature()*100;

pres=((bme.readPressure() / 100.0F)*0.014503773773)*100;

alt=(bme.readAltitude(SEALEVELPRESSURE_HPA))*100;

hum=(bme.readHumidity()*100);

}

void decibels(){

    int value;

    float decibelsValueQuiet = 30;

    float decibelsValueMedium = 40;

    float decibelsValueLoud = 60;

    value = analogRead(PIN_ANALOG_IN);

    if (value < 10)

    {

        decibelsValueQuiet += 30 * log10(value/dBAnalogQuiet);

        db= (int) decibelsValueQuiet;

        Serial.print("decibelios:" +String (decibelsValueQuiet));

    }

    else if ((value > 10) && ( value <= 30 )

    {

        decibelsValueMedium += 10 * log10(value/dBAnalogMedium);

        db= (int) decibelsValueMedium;

        Serial.print("decibelios:" +String (decibelsValueMedium));

    }

}

```

```

    }
    else if(value > 22)
    {
        decibelsValueLoud += log10(value/dBAnalogLoud);
        db = (int) decibelsValueLoud;
        Serial.print("decibelios:" +String (decibelsValueLoud));
    }

    delay(4500);
}

void loop() {
    modem.beginPacket();
    payload[0]=highByte(temp);
    payload[1]=lowByte(temp);
    payload[2]=highByte(hum);
    payload[3]=lowByte(hum);
    payload[4]=highByte(radi);
    payload[5]=lowByte(radi);
    payload[6]=highByte(pres);
    payload[7]=lowByte(pres);
    payload[8]=pmcf25;
    payload[9]=pmcf100;
    payload[10]=db;
    modem.write(payload,sizeof(payload));
}

```

```
//modem.print(pres);

//modem.print(alt);

//modem.print(hum);

//modem.print(radi);

//modem.print(pmcfl10);

//modem.print(pmcfl25);

//modem.print(pmcfl100);

Serial.println();

bme280();

sensor_veml_6075();

pm30003();

decibels();

int err = modem.endPacket(false);

if (err > 0) {

    Serial.println("Big success!");

} else {

    Serial.println("Error");

}

delay(18000);

}
```