



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos

Naturales no Renovables

Carrera de Telecomunicaciones

**Diseño de una Red de Sensores Inalámbricos (WSN) de Bajo
Consumo Energético para el Monitoreo Estructural de Puentes**

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

Autor:

Jhanes Alexander Pineda Chamba

Director:

Ing. Renato Benjamín Torres Carrión Mg. Sc.

Loja – Ecuador

2024

Certificación

Loja, 16 de Agosto de 2023

Ing. Renato Benjamín Torres Carrión. Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Integración curricular denominado: **Diseño de una red de sensores inalámbricos (WSN) de bajo consumo energético para el monitoreo estructural de puentes**, previo a la obtención del título de **Ingeniero en Telecomunicaciones**, de la autoría del estudiante **Jhanes Alexander Pineda Chamba**, con **cédula de identidad Nro. 1900885888**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.



Ing. Renato Benjamín Torres Carrión. Mg. Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Yo, **Jhanes Alexander Pineda Chamba**, declaro ser autor del presente Trabajo de Integración Curricular y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular, en el Repositorio Digital Institucional – Biblioteca Virtual.



Firma:

Cédula de identidad: 1900885888

Fecha: 15 de marzo del 2024

Correo electrónico: jhanes.pineda@unl.edu.ec

Teléfono: +593 99 432 8416

Carta de autorización por parte del autor, para consulta, reproducción parcial o total, y/o publicación electrónica de texto completo del Trabajo de Titulación.

Yo, **Jhanes Alexander Pineda Chamba**, declaro ser autor del Trabajo de Integración Curricular denominado: **Diseño de una red de sensores inalámbricos (wsn) de bajo consumo energético para el monitoreo estructural de puentes**, como requisito para optar por el título de **Ingeniero en telecomunicaciones**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional:

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

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

Para constancia de esta autorización, en la ciudad de Loja, a los quince días del mes de marzo de dos mil veinticuatro.



Firma:

Autor: Jhanes Alexander Pineda Chamba

Cédula: 1900885888

Dirección: Cumbaratza-Zamora-Ecuador

Correo electrónico: jhanes.pineda@unl.edu.ec

Teléfono: +593 99 432 8416

DATOS COMPLEMENTARIOS:

Director de trabajo de integración curricular: Ing. Renato Benjamín Torres Carrión. Mg. Sc.

Dedicatoria

Quiero dedicar este trabajo a Dios, quien ha sido mi guía y fortaleza en cada paso de mi vida. Su divina presencia ha iluminado mi camino y me ha dado la sabiduría y perseverancia necesarias para ser un ejemplo a seguir.

A mis padres, Alexandra Chamba y Miguel Pineda, les agradezco profundamente por ser mi apoyo incondicional y por haberme inculcado el valor del esfuerzo y la dedicación. Su amor y sacrificio han sido fundamentales para mi formación académica y personal.

A mis hermanos, les agradezco por ser mis compañeros de vida y por brindarme su apoyo en cada etapa de mi desarrollo. En especial a mi sobrina, le dedico este trabajo con la esperanza de ser un ejemplo a seguir en su vida.

A mis familiares, abuelos, tíos y primos les agradezco por su cariño, comprensión y apoyo constante. Su amor y aliento han sido un motor en mi camino de vida.

A todos mis amigos, les agradezco por estar siempre presentes, por sus palabras de aliento y por compartir conmigo momentos inolvidables. Su amistad ha sido un bálsamo en los momentos difíciles y una fuente de alegría en los momentos felices.

Finalmente, quiero dedicar este trabajo a la planta docente, quienes con su sabiduría y vocación han guiado mi aprendizaje a lo largo de mi formación académica, dejando una huella imborrable en mi camino hacia el crecimiento intelectual y profesional.

Jhanes Alexander Pineda Chamba

Agradecimiento

En primer lugar, mi más profundo agradecimiento a Dios, quien ha sido mi guía y fortaleza en cada paso de este recorrido. Su presencia en mi vida ha sido una fuente inagotable de esperanza e inspiración.

A mis queridos padres, les debo un reconocimiento especial por su amor incondicional, apoyo inquebrantable y sacrificios para que pudiera alcanzar mis metas académicas. Su constante aliento y ejemplo han sido mi mayor motivación.

A toda mi familia, les agradezco por su cariño y respaldo durante todo este proceso. Su ánimo y palabras de aliento han sido un estímulo inigualable.

Asimismo, quiero expresar mi profundo agradecimiento a mi director de integración curricular, Ing. Renato Benjamín Torres Carrión, cuya dedicación y sabiduría fueron fundamentales para el éxito de este trabajo. Su guía experta y sus valiosas sugerencias contribuyeron significativamente a dar forma a esta investigación.

También, quiero expresar mi agradecimiento a mi carrera de Telecomunicaciones y a la Universidad Nacional de Loja, por brindarme la oportunidad de adquirir conocimiento, crecer como persona y desarrollarme académicamente. Los profesores y el personal de la universidad han sido un pilar fundamental en mi formación.

Jhanes Alexander Pineda Chamba

Tabla de contenido

Portada.....	i
Certificación	ii
Autoría	iii
Carta de autorización.	iv
Dedicatoria.....	v
Agradecimiento	vi
Tabla de contenido.....	vii
Índice de tablas.....	x
Índice de figuras.....	xii
Índice de Anexos	xv
1. Título.....	1
2. Resumen.....	2
Abstract.....	3
3. Introducción	4
4. Marco teórico	6
4.1 Estado del arte.....	6
4.2 Parámetros de mediciones de puentes.....	8
4.2.1 Niveles de vibración	9
4.2.2 Velocidad del viento	10
4.2.3 Niveles de temperatura y humedad.....	11
4.3 Red de sensores inalámbricos	12
4.3.1 Topologías.....	12
4.3.2 Entidades WSN.....	13
4.3.3 Modelo de referencia de IoT.....	15
4.4 Red de área amplia de baja potencia LoRaWAN	16
4.4.1 Tecnología LoRaWAN	19
4.4.2 LoRa Alliance	20
4.4.3 Arquitectura LoRaWAN.....	20
4.4.4 Clases de dispositivos LoRaWAN.....	22
4.4.5 Activación de dispositivos	23
4.4.6 Modulación LoRa	24
4.4.7 Tasa de transferencia de datos	26
4.4.8 Seguridad de LoRaWAN	27
4.4.9 Tipos de mensajes	28

4.4.10 Mensajes de datos	28
4.4.11 Planes de frecuencia LoRaWAN	29
4.5 Interfaz de programación de aplicaciones.....	30
4.5.1 SOAP	30
4.5.2 XML-RPC.....	30
4.5.3 JSON-RPC.....	31
4.5.4 REST.....	31
4.5.5 WebSocket.....	31
5. Metodología	34
5.1 Diseño del sistema WSN	35
5.1.1 Análisis del Escenario.....	35
5.1.2 Simulación de enlaces.....	36
5.2 Arquitectura del sistema de monitoreo	38
5.2.1 Nodo final	39
5.2.2 Gateway	40
5.2.3 Sensor de vibración.....	41
5.2.4 Anemómetro	42
5.2.5 Termómetro.....	43
5.3 Simulación de la WSN en Omnet++.....	44
5.4 Pruebas de funcionamiento de sensores.....	46
5.4.1 Sensor de temperatura y humedad DHT22	46
5.4.2 Acelerómetro ADXL345	48
5.4.3 Sensor de vibración SW-420	50
5.4.4 Sensor de viento.....	51
5.5 Materiales.....	53
5.5.1 Gateway UG63.....	53
5.5.2 Sensor de vibración SW-420	55
5.5.3 Modulo sensor DHT22	57
5.5.4 Acelerómetro ADXL345	58
5.5.5 Sensor JL -FS2.....	61
5.5.6 Arduino Uno R3.....	62
5.5.7 Módulo LoRa CubeCell HTCC-AB01	64
6. Resultados.....	66
6.1 Registro de dispositivos LoRa	66
6.1.1 Registro del Gateway TTN	66
6.1.2 Registro del dispositivo TTN.....	67
6.2 Decodificador payload TTN	68
6.3 Integración TTN con Ubidots	69
6.3.1 Dashboard Nodo 1	72
6.3.2 Dashboard Nodo 2	73
6.4 Diseño de los circuitos impresos.....	73
6.5 Resultados de las pruebas de funcionamiento del sistema WSN.....	75
6.6 Análisis de resultados del sistema WSN.....	81

6.6.1	Detecciones de vibraciones.....	81
6.6.2	Temperatura y humedad	83
6.6.3	Velocidad del viento	87
6.6.4	Acelerómetro.....	88
6.7	Resultados de simulación en Omnet++	90
7.	Discusión	94
8.	Conclusiones	96
9.	Recomendaciones	98
10.	Bibliografía	100
11.	Anexos	105

Índice de tablas:

Tabla 1. Arquitecturas entre OSI, WLAN y WSN	14
Tabla 2. Comparativa de Tecnologías LPWAN	18
Tabla 3. Dispositivos LoRaWAN	23
Tabla 4. Activación de Dispositivos LoRaWAN.....	24
Tabla 5. Velocidades de datos de LoRaWAN	26
Tabla 6. Key in LoRaWAN version 1.0.x.....	27
Tabla 7. Key in LoRaWAN version 1.1.x.....	28
Tabla 8. Tipos de mensajes MAC.....	28
Tabla 9. MAC payload.....	29
Tabla 10. The frame header (FHDR).....	29
Tabla 11. Plan de frecuencias US902-928	30
Tabla 12. Comparativa de APIs para IoT	32
Tabla 13. Nodos del sistema WSN	40
Tabla 14. Características de un acelerómetro	41
Tabla 15. Características de un anemómetro	42
Tabla 16. Características de un termómetro.....	43
Tabla 17. Parámetros generales de la simulación en Omnet++	45
Tabla 18. Código fuente DTH22.....	47
Tabla 19. Código fuente ADXL345.....	49
Tabla 20. Código fuente SW-420	50
Tabla 21. Código fuente del anemómetro.....	52
Tabla 22. Especificaciones Técnicas del UG63.....	54
Tabla 23. Especificaciones técnicas del SW-420.....	57
Tabla 24. Especificaciones Técnicas del sensor DHT22	58
Tabla 25. Descripción de las funciones de los pines ADXL345	60
Tabla 26. Especificaciones técnicas del ADXL345.....	60
Tabla 27. Especificaciones Técnicas JL-FS2.....	61
Tabla 28. Especificaciones técnicas del Arduino Uno.....	63
Tabla 29. Especificaciones técnicas CubeCell HTCC-AB01	65
Tabla 30. Resultados de las detecciones de vibraciones del nodo 1	81
Tabla 31. Resultados de las detecciones de vibraciones del nodo 2	82
Tabla 32. Resultados de temperatura y humead del nodo 1.....	83

Tabla 33. Resultados de temperatura y humeado del nodo 2.....	85
Tabla 34. Resultados de la velocidad del viento.....	87
Tabla 35. Resultados del acelerómetro	88
Tabla 36. Total de energía consumida por nodo 1 y nodo 2.....	90

Índice de figuras:

Figura 1. Nivel de vibración en función del tiempo y fuerzas g.....	9
Figura 2. Velocidad del viento en metros por segundo	10
Figura 3. Medición de temperatura ambiente en puentes	11
Figura 4. Topologías comunes de redes de sensores inalámbricos.....	13
Figura 5. Componentes básicos de un nodo sensor	14
Figura 6. Modelo de referencia de IoT	15
Figura 7. Velocidad de datos requerida frente a la capacidad de alcance de las tecnologías de radiocomunicación.....	16
Figura 8. Comparativas LPWAN en términos de IoT	17
Figura 9. Estructura de protocolos LoRaWAN	19
Figura 10. Estructura de una red LoRaWAN: Sensor y Gateway	20
Figura 11. Arquitectura de red LoRaWAN	21
Figura 12. Diagrama de tiempo de clase de los dispositivos LoRaWAN.....	22
Figura 13. Señal de banda estrecha con una señal de espectro ensanchada	25
Figura 14. Seguridad de LoRaWAN.....	27
Figura 15. Mensaje de datos	29
Figura 16. Puente sobre el Río Bombuscaro.....	35
Figura 17. Simulación de enlaces del Gateway y los nodos	36
Figura 18. Perfil de elevación entre el Gateway y Nodo 1	37
Figura 19. Perfil de elevación entre el Gateway y Nodo 2	37
Figura 20. Enlace de comunicación del sistema de monitoreo.....	37
Figura 21. Diagrama de cobertura del Gateway	38
Figura 22. Arquitectura del Sistema de monitoreo Estructural de Puentes	39
Figura 23. Nodo Sensor	40
Figura 24. Escenario de simulación en Omnet++.....	45
Figura 25. Diagrama de prueba sensor DHT22	46
Figura 26. Prueba de Campo del DTH22	47
Figura 27. Diagrama de prueba ADXL345	48
Figura 28. Prueba de campo del ADXL345	49
Figura 29. Diagrama de prueba SW-420	50
Figura 30. Prueba de campo del SW-240	51
Figura 31. Diagrama de prueba del anemómetro.....	51

Figura 32. Datos obtenidos del anemómetro	52
Figura 33. Pruebas de Campo del Anemómetro	53
Figura 34. Milesight UG63	54
Figura 35. Configuración de pines SW-420	56
Figura 36. Diagrama del SW-420	56
Figura 37. Módulo DHT22	57
Figura 38. Diagrama Sistemático del DHT22	58
Figura 39. Configuración de pines del ADXL345.....	59
Figura 40. Diagrama de bloques funcional del ADXL345	59
Figura 41. Sensor JL-FS2	61
Figura 42. Pin mapping Arduino UNO.....	62
Figura 43. Chip ATmega328P	63
Figura 44. Pin Mapping Heltec's Cubecell HTCC-AB01	64
Figura 45. Formulario de registro de Gateway TTN	66
Figura 46. Formulario de registro de aplicación TTN.....	67
Figura 47. Registro del nodo TTN.....	68
Figura 48. Decodificador payload TTN.....	69
Figura 49. Integraciones de TTN	70
Figura 50. Integración con Ubidots	70
Figura 51. Creación de token Ubidots	71
Figura 52. Creación del Plugin	71
Figura 53. Decoder Settings.....	72
Figura 54. Dashboard del Nodo 1	72
Figura 55. Dashboard del Nodo 2.....	73
Figura 56. Diseño del circuito impreso Nodo 1	74
Figura 57. Placa PCB del nodo 1	74
Figura 58. Diseño del circuito impreso Nodo 2.....	75
Figura 59. Placa PCB del nodo 2.....	75
Figura 60. Nodos finales con sensores integrados en caja protectora	76
Figura 61. Despliegue estratégico de los nodos de finales	77
Figura 62. Visualización de datos del anemómetro (Km/h).....	78
Figura 63. Visualización de datos del acelerómetro (g)	78
Figura 64. Visualización de datos de humedad (%)	79

Figura 65.	Visualización de datos de temperatura (°C).....	79
Figura 66.	Visualización de datos de sensación térmica (°C)	80
Figura 67.	Visualización de datos de detecciones de vibraciones	80
Figura 68.	Historial de las detecciones de vibraciones del nodo 1	82
Figura 69.	Historial de las detecciones de vibraciones del nodo 2.....	83
Figura 70.	Historial de temperatura del nodo 1	84
Figura 71.	Historial de humedad del nodo 1.....	85
Figura 72.	Historial de temperatura del nodo 2	86
Figura 73.	Historial de humedad del nodo 2.....	86
Figura 74.	Historial de la sensación térmica.....	87
Figura 75.	Historial de la velocidad del viento.....	88
Figura 76.	Historial del acelerómetro eje x (g).....	89
Figura 77.	Historial del acelerómetro eje y (g).....	89
Figura 78.	Historial del acelerómetro eje z (g)	90
Figura 79.	Energía consumida nodo 2	91
Figura 80.	Energía consumida nodo 1	91
Figura 81.	Total de energía consumida del sistema WSN.....	92
Figura 82.	Resultados de las simulaciones en OMNET++.....	93

Índice de Anexos:

Anexo 1. Guía de instalación HELTEC serie CubeCell.....	105
Anexo 2. Código fuente LoRaNetworkTsd.ned	107
Anexo 3. Código fuente n2-gw1.ini	108
Anexo 4. Código fuente energyConsumptionParameters.xml	109
Anexo 5. Código fuente cloudDelays.xml.....	109
Anexo 6. Código de programación del Nodo 1	110
Anexo 7. Código de programación del esclavo	114
Anexo 8. Código de programación del nodo 2	116
Anexo 9. Código de programación del decodificador del Nodo 1	120
Anexo 10. Código de programación del decodificador del Nodo 2	121
Anexo 11. Datasheet SW-420.....	122
Anexo 12. Datasheet ADXL345.....	123
Anexo 13. Datasheet DHT22.....	124
Anexo 14. Datasheet Heltec's CubeCell	126
Anexo 15. Datasheet Gateway UG63.....	130
Anexo 16. Wind Sensor (Pulse -T ype) JL -FS2.	131
Anexo 17. Diseño final para la placa base de los nodos finales.	132
Anexo 18. Escalabilidad de la red WSN para el monitoreo estructural de puentes	137
Anexo 19. Certificado de traducción del Abstract.....	138

1. Título

Diseño de una red de sensores inalámbricos (WSN) de bajo consumo energético para el monitoreo estructural de puentes

2. Resumen

La presente tesis se centra en el diseño de una red de sensores inalámbricos basada en la tecnología LoRa para el monitoreo del estado estructural de puentes. La red de sensores fue diseñada para medir y monitorear variables físicas clave, que incluyen temperatura, humedad, velocidad del viento y vibraciones. Para lograr esto, se utilizó una variedad de sensores, incluyendo el DHT22 para la medición de temperatura y humedad ambiental, el ADXL345 para la detección de vibraciones, el SW-420 para la detección de movimientos sísmicos y un anemómetro para la medición de la velocidad del viento. Los datos recolectados por los sensores fueron transmitidos a través de la red LoRa y enviados a la plataforma The Things Network (TTN) para su procesamiento y almacenamiento en la nube. Posteriormente, se llevó a cabo la integración de estos datos con la plataforma Ubidots, que permitió su visualización y análisis en tiempo real. El propósito principal de esta tesis fue desarrollar una solución eficiente y confiable para el monitoreo continuo del estado estructural de puentes utilizando tecnología IoT y LoRa.

Palabras Clave: LoRa, Sensores, The things networks, Ubidots, IoT, Ciudades inteligentes.

Abstract

The present thesis focuses on the design of a wireless sensor network based on LoRa technology for structural conditions of bridges monitoring. The sensor network was designed to measure and monitor key physical variables that include temperature, humidity, wind speed and vibrations. To achieve this, a variety of sensors were used, including the DHT22 for temperature and ambient humidity measurement, the ADXL345 for vibration detection, the SW-420 for seismic motion detection, and an anemometer for wind speed measurement. The data collected by the sensors were transmitted through the LoRa network and sent to The Things Network (TTN) platform for processing and storage in the cloud. Subsequently, the integration of this data with the Ubidots platform was carried out, which allowed its visualization and analysis in real-time. The main purpose of this thesis was to develop an efficient and reliable solution for the continuous monitoring of the structural condition of bridges using IoT and LoRa technology.

***Keywords:** LoRa, Sensors, The things networks, Ubidots, IoT, smart cities.*

3. Introducción

La inclusión de las nuevas tecnologías de comunicación inalámbricas ha dado lugar a un nuevo paradigma conocido como Internet de las Cosas (IoT, del inglés *Internet Of Things*). Este enfoque tecnológico permite obtener información detallada de forma remota y eficiente, utilizando menos recursos y energía. Entre las tecnologías de comunicación inalámbricas existentes en el ámbito de IoT se dispone de SigFox¹, LoRaWAN², WiFi³, Bluetooth Low Energy (BLE)⁴ y ZigBee⁵. Recalcando que la tecnología LoRaWAN se destaca por su largo alcance de conectividad, bajo costo y menor consumo de energía.

Dentro de las aplicaciones de IoT, el monitoreo de variables físicas relevantes para la toma de decisiones es especialmente notable. Esto incluye la medición y control de la contaminación en las ciudades, los niveles de humedad en edificaciones o plantaciones agrícolas, o la temperatura de los árboles en un bosque para alertar incendios forestales, entre otros. Sin embargo, uno de los campos de mayor importancia es el monitoreo del estado de infraestructuras civiles, como edificaciones, carreteras y puentes, siendo este último fundamental para la conectividad y el desarrollo de una región o país.

Los ministerios e instituciones gubernamentales encargadas de las supervisiones de las vías del país, carecen de una tecnología de bajo coste que registren estadísticas del estado estructural de los puentes. Estudios demuestran que un total de 72 puentes analizados durante un periodo de 23 años (2000-2023), solamente en la segunda mitad del periodo (11 años), 54 puentes colapsaron, lo que equivale al 75% del total. En cuanto al número de puentes que fallaron, la provincia con mayor número de colapsos es Guayas, con 13 casos de fallas. Seguida por la provincia de Pichincha, con 10 colapsos y la provincia de Zamora Chinchipe con 8 casos (Marín & Maldonado, 2022). Por tanto, la detección temprana de daños estructurales puede reducir los costos de reemplazo y reparación de puentes, prolongando su vida útil y garantizando la seguridad del tráfico vial.

Además, en la actualidad el monitoreo visual es la técnica más comúnmente utilizada para controlar la integridad estructural de puentes. Sin embargo, esta técnica tradicional depende en gran medida de la experiencia de los inspectores y ha demostrado ser insuficiente

¹ SigFox. 2023. [Home - Sigfox 0G Technology](#)

² Lora Alliance. LoraWAN. 2023. [Homepage - LoRa Alliance® \(lora-alliance.org\)](#)

³ WiFi Alliance. WiFi. 2023. [Wi-Fi Alliance](#)

⁴ Bluetooth Technology. BLE. 2023. [Bluetooth® Technology.](#)

⁵ Connectivity Standards Alliance (CSA). Zigbee. 2023. [Zigbee | Complete IOT Solution - CSA-IOT](#)

para garantizar la seguridad del puente. En respuesta a esta limitación, se han desarrollado otras técnicas de inspección, como los sistemas de monitoreo que utilizan sensores con líneas de transmisión, como cables de cobre o fibra óptica, para la comunicación de datos. Estos sistemas ofrecen la ventaja de proporcionar información en tiempo real sobre el estado estructural del puente. Sin embargo, su implementación resulta costosa desde el punto de vista económico, ya que requiere la instalación de cables de datos a lo largo del puente.

Por ende, el presente proyecto de titulación expone una solución tecnológica mediante el uso de LoRaWAN para monitoreo del estado estructural de puentes. Este sistema se realiza a través del despliegue de una red de sensores inalámbricos, la cual se encarga de obtener y transmitir los datos a una interfaz de programación de aplicaciones donde son analizados para la toma de decisiones.

Este trabajo tiene como objetivo general: Diseñar Una Red De Sensores Inalámbricos (WSN) De Bajo Consumo Energético Para El Monitoreo Estructural De Puentes, de la misma forma sus objetivos específicos son:

- Realizar una revisión del estado del arte de publicaciones científicas relacionadas con el tema propuesto.
- Llevar a cabo un análisis del puente seleccionado que permita describir su estado estructural con el propósito de identificar los sitios estratégicos donde se instalarán los sensores y funcionalidades requeridas para el diseño de la red.
- Diseñar una red de sensores inalámbricos que permita la monitorización del puente en base al escenario analizado y los requerimientos identificados.
- Desarrollar un prototipo funcional del diseño de la red de sensores inalámbrica para el monitoreo del estado estructural del puente.

Este trabajo aborda varios temas de interés general y está estructurado de la siguiente forma: en la sección 4, se definen los trabajos relacionados con el tema planteado e investigaciones de temas que contribuyan con el desarrollo del tema. En la sección 5, se describen los materiales y métodos a utilizar en el proyecto. Luego, en la sección 6 se presentan los resultados obtenidos a partir de las pruebas de funcionamiento del sistema de monitoreo estructural de puentes. Seguidamente, en la sección 7 se describe el aporte que ha generado el proyecto. Finalmente, en la sección 8 se dan algunas conclusiones del proyecto realizado y futuras líneas de investigación.

4. Marco teórico

En esta sección se realiza una revisión comprensiva de la literatura a las investigaciones relacionadas con el tema planteado y conceptos esenciales para el desarrollo de la misma. El propósito de esta revisión es adquirir conocimiento sobre las técnicas de inspección utilizadas para evaluar el estado estructural de puentes, así como las mediciones de parámetros físicos relevantes. La información obtenida será utilizada para cumplir con los objetivos establecidos en el proyecto.

4.1 Estado del arte

Algunas investigaciones centran sus esfuerzos en realizar estudios en nuevas técnicas para inspeccionar constantemente infraestructuras viales como en el artículo realizado por Quezada et al (2017), en la que han diseñado un sistema de monitoreo para estructuras de puentes utilizando sensores inteligentes y comunicación inalámbrica GSM/GPRS. Se emplearon el módulo SIM900 GSM/GPRS junto con acelerómetros MMA736. Además, se visualizaron los datos recolectados en la plataforma API Ubidots. Sin embargo, GSM/GPRS es más adecuado para aplicaciones que requieren un mayor ancho de banda, velocidades de transmisión más altas y dependería de la infraestructura de un proveedor de servicio móvil.

En el trabajo de Mejía & Salazar (2019), se plantea el desarrollo de un prototipo para monitorear la salud estructural mediante WSN y comunicación inalámbrica IEEE 802.15.4 (Zigbee). El dispositivo se construyó utilizando un Arduino Uno, dispositivos XBee y sensores de vibración SW-420. La información recolectada se almacenó en una base de datos SQLite, con la que se procesa y visualiza a través del software Matlab. Sin embargo, se debió considerar la limitación del alcance de la tecnología inalámbrica, ya que Zigbee tiene un alcance corto de entre 10 a 100 metros, lo cual resulta desventajoso dado que los puentes suelen estar ubicados en áreas remotas.

En el estudio de Sinchi (2022), desarrollaron sistema de monitoreo de estado de salud de un puente utilizando la técnica PPP-GNSS. La técnica PPP-GNSS se trata de procesamiento de señales de navegación global por satélite (GNSS) para determinar con alta precisión la posición de un receptor GNSS. Debido a que PPP-GNSS es muy costoso, construyeron un prototipo basado en GSM GPRS para asemejar a la técnica PPP-GNSS. Para ello, utilizaron componente como GPS neo 6M para el posicionamiento, Arduino nano, el módulo SIM800L para la comunicación mediante GSM/GPRS y un acelerómetro MPU6050. La información recolecta es enviada por mensajes de texto (SMS) con dirección a Google Maps. Sin

embargo, este sistema no es factible debido a que se necesita una plataforma para la visualización de datos.

Otro artículo publicado por Aguilar et al (2016), realizaron una evaluación y análisis de un prototipo de sensor inalámbrico para infraestructuras críticas de puentes a través de una comunicación inalámbrica Zigbee. El prototipo consistió en un conjunto de sensores que recopilaban información sobre las vibraciones presentes en el puente. Además, utilizaron Radio Mobile para el diseño de la red de transporte, y NS2 para evaluar la calidad y rendimiento de la red de sensores inalámbricos.

De la misma manera, en el estudio realizado por Naranjo (2021) han desarrollado un sistema de monitoreo de salud estructural (SHM) utilizando redes inalámbricas de sensores (WSN) con tecnología Zigbee. El sistema integró sensores como el ADXL335 para medir vibraciones, el DHT22 para medir temperatura y humedad, y el módulo XBee S2C. A diferencia de trabajos previos, se incorporó una tarjeta SD para almacenar los datos recopilados. Sin embargo, esta característica introduce una posible desventaja, ya que la información almacenada en la tarjeta SD puede estar en riesgo de pérdida debido a posibles daños en el hardware del componente.

En el estudio realizado por Salgado & Carranza Geomayra (2016), desarrollaron un prototipo para un sistema de monitoreo de vibración e inclinación en puentes utilizando ZigBee. La red constaba de tres nodos: un coordinador y dos sensores. Estos sensores miden y transmiten vibración e inclinación al coordinador, que procesa la información en una interfaz de usuario de MatLab. Se emplearon componentes como placas Arduino MEGA 2560, radios XBee S2, acelerómetros ADXL335 y módulos sw-420 para la detección de vibración. Las pruebas se llevaron a cabo en puentes de hormigón en los cantones Chambo y Penipe de Chimborazo. Se evaluó el prototipo en momentos de alta y baja actividad vehicular, recolectando datos durante una hora en intervalos de diez minutos.

En contraste, en el estudio publicado por Zambrano & Zambrano (2020) utilizaron sensores de fibra óptica basados en la red de Bragg para el monitoreo en un puente. Los materiales usados en dicho estudio incluyeron un equipo de adquisición de datos de fibra óptica, multiplexor, sensor de desplazamiento, sensor de temperatura, inclinómetros, acelerómetros y extensómetros. Sin embargo, a pesar de ser una innovación tecnológica, tiene desventajas con respecto al costo de implementación de los materiales y la visualización de datos es complejo.

En el artículo publicado por Kurata et al. (2013), realizaron la implementación de un sistema de monitoreo estructural inalámbrico diseñado específicamente para infraestructuras civiles a gran escala. El sistema se basó en una red WSN y constó de dos capas principales. La capa inferior utilizó WSN de baja potencia para registrar la carga ambiental y la respuesta estructural correspondiente. Mientras tanto, la capa superior habilitada para Internet proporcionó un marco de ciberinfraestructura para la gestión y consulta de datos. El sistema fue instalado en el puente Nuevo Carquinez, donde se desplegaron 28 nodos que recopilaban datos de 81 canales. Este enfoque permitió un monitoreo detallado y en tiempo real de la estructura del puente, facilitando la toma de decisiones informadas en cuanto a mantenimiento y seguridad.

En cambio, en el artículo desarrollado por Al-Radaideh et al. (2015) propusieron un sistema WSN de monitoreo estructural para puentes de carreteras. Este sistema utilizó una unidad de adquisición de datos inalámbrica, redes móviles públicas y un sistema de evaluación de datos para recopilar y procesar información sobre el estado del puente. Además, se utiliza un mapa interactivo basado en Google para mostrar la ubicación y el estado de cada puente. Este enfoque proporciona un monitoreo efectivo y en tiempo real de la salud estructural de los puentes de carreteras, permitiendo una gestión y mantenimiento más eficientes.

Finalmente, el proyecto desarrollado por Quinde (2016) ha implementado un sistema de monitoreo estructural utilizando el estándar IEEE 802.15.4 (Zigbee) donde los nodos fueron constituidos con el módulo Xbee pro S2B para el establecimiento de las comunicaciones. Su sistema integra sensores de humedad y temperatura y acelerómetro. Sin embargo, existen otras variables que no fueron consideradas como la velocidad del viento que es parte esencial para el correcto monitoreo del estado de salud estructural de puentes. De la misma forma, la tecnología empleada como lo es Zigbee resulta muy corta debido a su pequeño alcance por lo que es conveniente integrarse con tecnologías de largo alcance como LoRa.

4.2 Parámetros de mediciones de puentes

En esta sección, se llevó a cabo una indagación detallada acerca de las variables físicas necesarias para el monitoreo exhaustivo del estado estructural de puentes. Estas variables son fundamentales para evaluar la integridad y el rendimiento de las estructuras,

proporcionando información crítica para el mantenimiento, la seguridad y la prolongación de la vida útil de los puentes.

4.2.1 Niveles de vibración

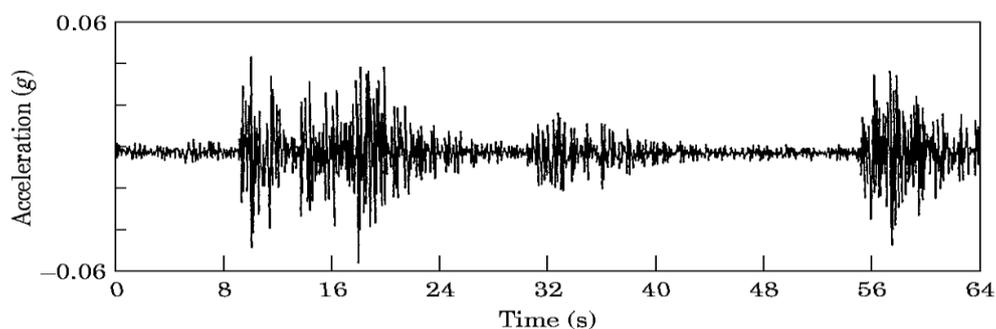
El nivel de vibración es el movimiento oscilatorio de un cuerpo provocado por la oscilación de las partículas que lo forman, respecto a una posición de equilibrio. La vibración es la respuesta repetitiva, periódica u oscilatoria de un sistema mecánico. Los movimientos repetitivos que son libres de aberraciones y regulares y ocurren a frecuencias relativamente bajas se denominan comúnmente "oscilaciones", mientras que cualquier movimiento repetitivo, incluso a altas frecuencias, con amplitudes bajas y que tiene un comportamiento irregular y aleatorio cae en la clase general de vibración (Clarence W. de Silva, n.d.).

Las mediciones de vibración deben ser hechas con la finalidad de producir los datos necesarios, para realizar significativas conclusiones de un sistema bajo prueba. Un cuerpo vibra cuando experimenta cambios alternativos, de modo que sus puntos oscilan sincrónicamente en torno a sus posiciones de equilibrio, sin que el campo cambie de lugar. En otras palabras, es un intercambio de energía cinética en cuerpos con rigidez y masa finitas, el cual surge de una entrada de energía dependiente del tiempo (Daniel & Pernia-Márquez, 2004).

En la **Figura 1**, se muestra un ejemplo de historia temporal típica medida en un puente durante las pruebas de vibración ambiental con acelerómetros piezoeléctricos con una gama de frecuencias especificada de 1 a 2.000 Hz y una gama de amplitudes de 24 g.

Figura 1.

Nivel de vibración en función del tiempo y fuerzas g



Nota. La figura muestra mediciones de vibraciones en función del tiempo y fuerzas g. Fuente: (C. R Farrar & G. H James, 1997).

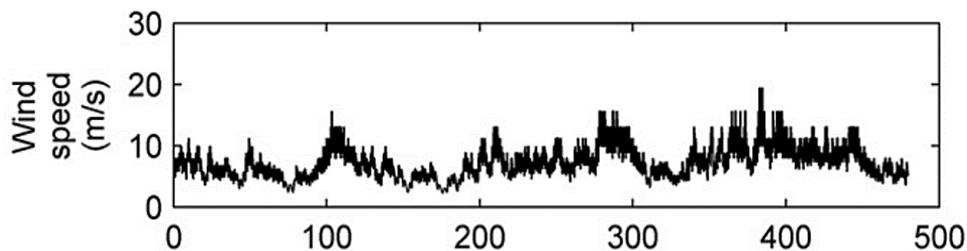
4.2.2 Velocidad del viento

La velocidad del viento es una variable física que mide horizontalmente el desplazamiento del aire en un punto determinado. Su medición es importante en el análisis climático, meteorológico y sobre todo como medida derivada para el procesamiento de datos ambientales. El instrumento con el que se mide la velocidad del viento es un anemómetro y actualmente el método de medición más utilizado implica el uso de sistemas mecánicos que se ven afectados con el tiempo por el desgaste y rozamiento, entre otros (Ospina Rojas, 2021).

El monitoreo del viento es clave durante todas las etapas del ciclo de vida de un puente, desde el diseño y la construcción hasta la operación y el mantenimiento. Los puentes suelen estar expuestos a fuertes corrientes de viento que los golpean perpendicularmente, y su gran tamaño resulta en cargas mecánicas muy altas inducidas por el viento. Incluso si un puente está diseñado adecuadamente de acuerdo con las condiciones del viento, el viento también puede representar un riesgo para los vehículos (Windcrane, 2019).

Figura 2.

Velocidad del viento en metros por segundo



Nota. Niveles de velocidad del viento en metros por segundo. Fuente: (Kurata et al., 2013).

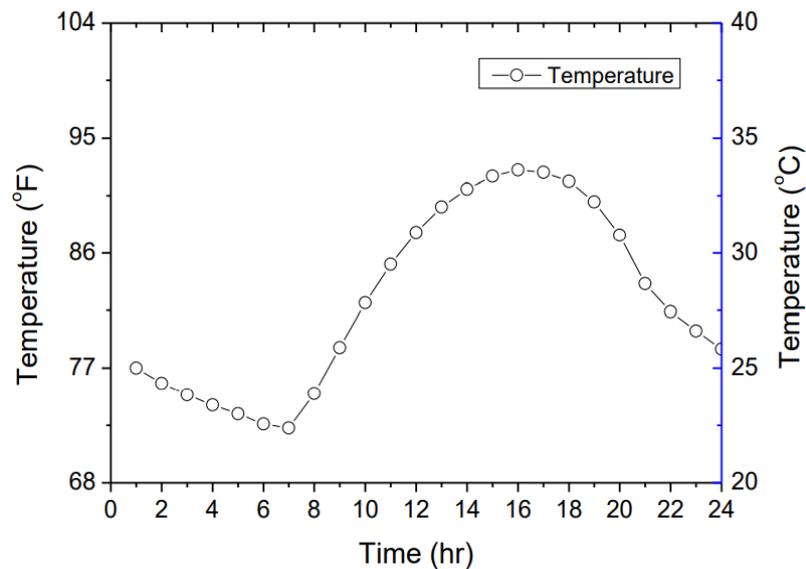
Adicionalmente, la influencia del viento en la resistencia y fiabilidad de las estructuras flexibles es muy grande. Las siguientes reacciones Aero elásticas pueden aparecer como resultado del efecto del flujo de aire sobre las estructuras del puente: fenómeno de aleteo, galope, excitación de vórtice, amortiguamiento y resonancia paramétrica. Estos fenómenos son causados por los efectos dinámicos del viento. Cabe señalar que cada una de las reacciones estructurales enumeradas representa un gran peligro para las personas (Poddaeva et al., 2019).

4.2.3 Niveles de temperatura y humedad

Las variaciones de temperatura podrían tener una lenta pero un efecto muy significativo en la respuesta a largo plazo de los puentes, y debe tomarse en consideración al abordar el diseño y monitoreo de puentes. Los estudios muestran que la carga térmica de la variación de temperatura diurna y estacional influye en su respuesta y durabilidad a largo plazo (Borah et al., 2021).

Figura 3.

Medición de temperatura ambiente en puentes



Nota. La gráfica muestra la medición de temperatura ambiente en puentes dados en grados Fahrenheit. Fuente: (He et al., n.d.).

Así mismo, las variaciones espaciales y temporales de temperatura en las estructuras de los puentes debido a los cambios en el clima circundante causan la expansión del puente que, si se restringe, puede inducir tensiones secundarias en las estructuras. Los efectos de la temperatura siempre se han considerado uno de los problemas más críticos en el rendimiento de los puentes tradicionales (He et al., 2021).

Por otra parte, la presencia de humedad puede provocar corrosión en elementos metálicos, lo que afectaría la resistencia y la durabilidad del puente. Por ende, el seguimiento de los niveles de humedad en el ambiente circundante y dentro de la estructura ayuda a identificar posibles problemas de corrosión en elementos metálicos.

Según (Cristian Camilo Vargas Duran, 2020), al verificar el fenómeno climatológico, se consideran los cambios exógenos y endógenos que experimenta un cuerpo, manifestándose en niveles mínimos y máximos de temperatura durante el día. Estas modificaciones son

resultado de la interacción con otros fenómenos climáticos, como la velocidad del viento y la humedad, y son las responsables del comportamiento térmico de una estructura, pudiendo generar procesos de dilatación y contracción.

4.3 Red de sensores inalámbricos

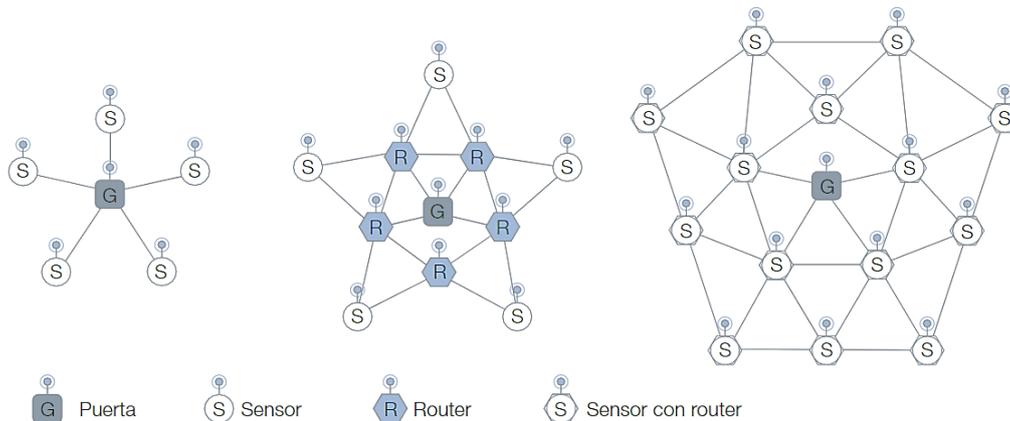
En cuanto a las redes de sensores inalámbricos (WSN, del inglés *Wireless Sensor Networks*) son una tecnología emergente con una amplia gama de aplicaciones potenciales, como los sistemas de monitorización de pacientes, la detección de terremotos, la monitorización del entorno y las aplicaciones militares. Las redes de sensores distribuyen espacialmente sensores autónomos para supervisar las condiciones físicas y ambientales en diferentes lugares, como la temperatura, la presión, el sonido del movimiento, la vibración, etc (Sharma et al., 2013).

Las redes WSN están formadas por sensores conectados a nodos que se comunican de forma inalámbrica a través de una pasarela común con usuarios finales remotos mediante aplicaciones, páginas web o el control directo de los dispositivos. A los nodos puede conectarse una amplia gama de sensores ambientales. Cada nodo suele estar formado por un microcontrolador, una electrónica de apoyo y un módulo de comunicación por radio (Ekanayake & Hedley, 2018).

4.3.1 Topologías

Las dos topologías principales de las WSN son la topología en estrella y la topología en malla. En la **Figura 4**, se ilustra las topologías en estrella y en malla. En la topología en estrella, un conjunto de nodos comúnmente utilizado, tiene dos tipos de nodos, un nodo coordinador y muchos nodos finales. Las topologías de malla son mucho más flexibles en lo que respecta a la comunicación, ya que los nodos suelen ser capaces de enrutar los datos entre ellos hasta llegar a un destino designado, y este destino suele ser un coordinador o pasarela que se encarga de registrar o reenviar los datos a una red externa (Ekanayake & Hedley, 2018).

Figura 4.
Topologías comunes de redes de sensores inalámbricos



Nota. Las topologías WSN integran la puerta de enlace, sensores y router. Fuente: (Aakvaag & Frey, 2010).

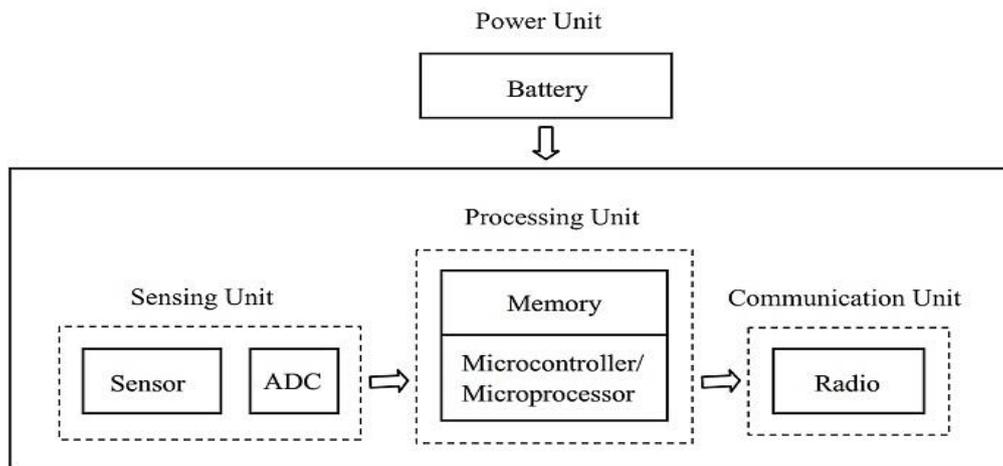
4.3.2 Entidades WSN

Una red de sensores WSN tradicional se puede dividir en dos entidades esenciales. Estas entidades se mencionan a continuación: Nodo sensor y la arquitectura de red en general.

El nodo sensor es encargado de obtener datos del mundo físico, el análisis de redes, la correlación de datos y la fusión de datos de otros sensores con sus propios datos. Un nodo sensor en un sistema WSN se compone de cuatro elementos básicos representados en la **Figura 5**, las cuales se describen a continuación:

1. **Fuente de alimentación (*Power Unit*):** encargado de alimentar energéticamente al nodo sensor.
2. **Sensor (*Sensing Unit*):** recopila los datos analógicos del mundo físico y un ADC convierte estos datos en datos digitales.
3. **Unidad de procesamiento (*Processing Unit*):** suele ser un microprocesador o un microcontrolador, realiza un procesamiento y manipulación inteligente de datos.
4. **Sistema de comunicación (*Communication Unit*):** consiste en un sistema de radio para la transmisión y recepción de datos.

Figura 5.
Componentes básicos de un nodo sensor



Nota. Un nodo sensor en una red WSN contiene cuatro componentes esenciales. Fuente: (Electronics Hub, 2019).

Por otra parte, la arquitectura más común en un sistema WSN sigue el modelo de interconexión de sistemas abiertos (OSI, del inglés *Open Systems Interconnection*). Por lo general, en una red de sensores inalámbricos se incluyen cuatro capas como se muestra en la **Figura 6**. En la siguiente **Tabla 1**, se destacan las diferencias de las redes WSN con respecto a la red WLAN basados en el modelo OSI.

Tabla 1.
Arquitecturas entre OSI, WLAN y WSN

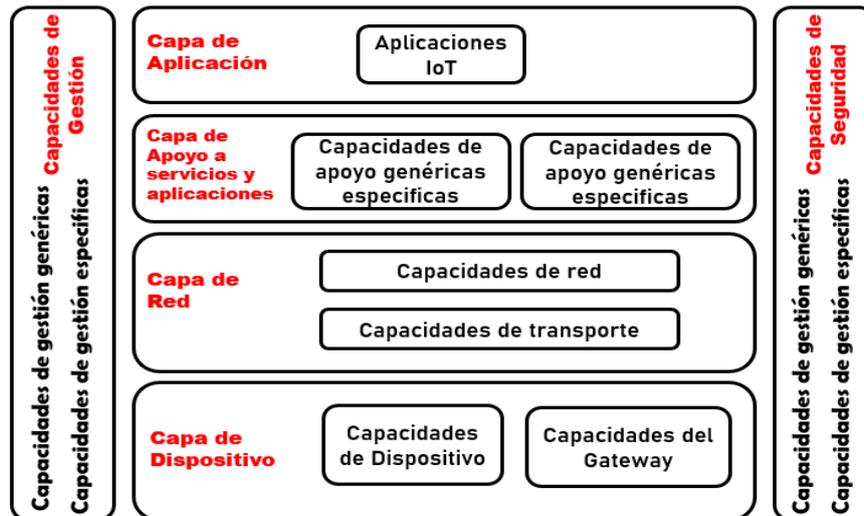
WSN	WLAN	OSI Model
WSN Application	Application programs	Application layer
WSN Middleware	Middleware	Presentation layer
	Socket API	Session layer
WSN Transport protocols	TCP/UDP	Transport layer
WSN routing protocols	IP	Network layer
Error control	WLAN Adapter & device driver	Data link layer
WSN MAC protocols	WLAN MAC protocols	
Transceiver	Transceiver	Physical layer

Nota. La tabla destaca las principales diferencias de arquitecturas entre OSI, WLAN y WSN. Fuente: (Alkhatib, 2011).

4.3.3 Modelo de referencia de IoT

De acuerdo a la Unión Internacional de Telecomunicaciones (ITU, 2022), define que la arquitectura de IoT generalmente está compuesta de cuatro capas: capa de aplicación, capa de apoyo a servicios y aplicaciones, capa de red y capa de dispositivo; y dos módulos: gestión y seguridad, como se ilustra en la **Figura 6**.

Figura 6.
Modelo de referencia de IoT



Nota. Capas de IoT según UIT. Fuente: (UIT-T, 2012).

- **Capa de aplicación:** Básicamente contiene las aplicaciones IoT y pone en uso la gran cantidad de información creada a partir del IoT.
- **Capa de soporte de servicios y aplicaciones:** comúnmente definida como la computación en la nube (*Cloud Computing*), consiste en los dos siguientes grupos de capacidades:
 - **Capacidades de soporte genéricas:** se trata de las capacidades que se puede utilizarlas diferentes aplicaciones IoT, como procesamiento o almacenamiento de datos.
 - **Capacidades de soporte específicas:** Capacidades para satisfacer las necesidades específicas de diversas aplicaciones. De hecho, pueden constar de diferentes grupos de funciones precisas que brindan diferentes funciones para admitir diferentes aplicaciones de IoT.
- **Capa de red:** es la infraestructura de red, consiste en dos tipos de capacidades:

- **Capacidades de red:** brinda control de la conectividad en red, como funciones de control de acceso y de recursos de transporte, gestión de la movilidad y autenticación, autorización y contabilidad (AAA).
- **Capacidades de transporte:** encargada de la conectividad para el transporte de información, datos específicos de servicios, aplicaciones IoT, transporte de información de control y gestión relacionada con IoT.
- **Capa de dispositivos:** conecta todo con el Internet y la infraestructura para el IoT. Se puede efectuar una clasificación lógica de las capacidades de la capa de dispositivo, en dos tipos, capacidades de dispositivo y Capacidades del Gateway.

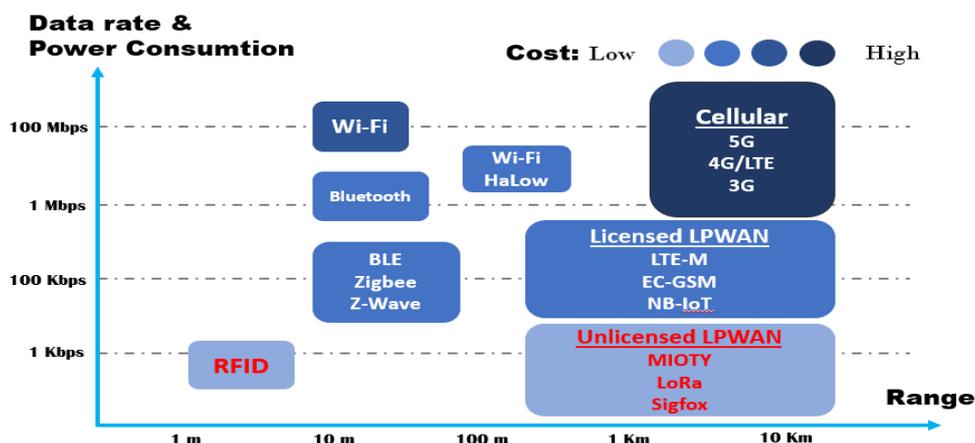
4.4 Red de área amplia de baja potencia LoRaWAN

Las redes de área amplia de baja potencia (LPWAN, del inglés *Low Power Wide Area Networks*) son tecnologías inalámbricas con características como grandes áreas de cobertura, poco ancho de banda, posiblemente de paquetes y datos de la capa de aplicación, y funcionamiento con baterías de larga duración. De la misma forma, las redes WSN se utilizan para conectar dispositivos de bajo consumo como sensores y controladores en IoT y las comunicaciones máquina a máquina (M2M, del inglés *machine to machine*). Las comunicaciones IoT y M2M requieren una red de sensores inalámbricos con ciertos requisitos, como largo alcance, duración de la batería y un coste muy bajo (Farrell, 2018; Naik, 2018).

En la **Figura 7**, observamos una comparativa de la velocidad de datos y el rango de cobertura de las tecnologías de comunicación inalámbrica. LPWAN se encuentra en la categoría de largo alcance con bajas tasas de transmisión siendo ideal para sistemas IoT.

Figura 7.

Velocidad de datos requerida frente a la capacidad de alcance de las tecnologías de radiocomunicación

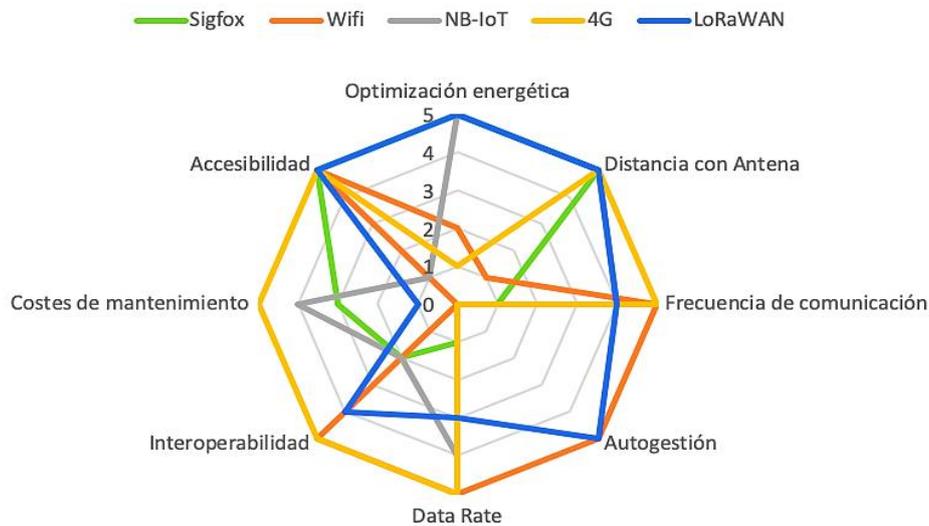


Nota. Adaptado de (Adamu et al., 2021)

En la **Figura 8**, se presenta una comparativa de las principales tecnologías LPWAN que se utilizan en la actualidad. Se pueden observar diferentes aspectos a considerar en estas tecnologías IoT, como la eficiencia energética, el alcance de cobertura, la frecuencia de operación, entre otros. Si analizamos estos parámetros, podemos concluir que LoRaWAN es especialmente adecuado para aplicaciones que requieren una gran cobertura, ya que puede alcanzar distancias de hasta 5 km en entornos urbanos y 20 km en entornos rurales. Además, ofrece la ventaja de ser una solución rentable, con una larga duración de la batería y un tiempo de implementación reducido.

Figura 8.

Comparativas LPWAN en términos de IoT



Nota. La gráfica se basa en la optimización de energía, alcance de cobertura, frecuencia de operación, etc. Fuente: (Logitek & Fernando Campos, 2020).

Adicionalmente, en la **Tabla 2** se comparan las principales características técnicas de las tecnologías LPWAN. Al seleccionar una tecnología adecuada para aplicaciones IoT, es necesario considerar varios factores, como la cobertura, la duración del dispositivo, la escalabilidad, la seguridad, el tiempo de implementación y el costo. Basándonos en el análisis proporcionado, se puede concluir que LoRa ofrece características de seguridad como cifrado de extremo a extremo, autenticación mutua y confidencialidad de los datos. LoRa proporciona una conectividad de largo alcance con un bajo consumo energético, lo que permite una vida útil de la batería de hasta 10 años. Además, opera en bandas de frecuencia ISM (del inglés *Industrial, Scientific and Medical*). La razón por la cual se elige esta tecnología LPWAN se debe a sus ventajas en cuanto a costos, duración de la batería e implementación eficiente.

Tabla 2.
Comparativa de Tecnologías LPWAN

Property or Technology	NB-IoT	SigFox	Telensa	Ingenu RPMA	LoRa
Founder	3 GPP	SigFox	Telensa	Ingenu	Semtech
Standard	3 GPP Release 13 and 14	SigFox and ETSI LTN	Telensa and ETSI LTN	RPMA and IEEE 802.15.4 K	LoRa Alliance
Frequency band	Licensed 7–900 MHz	Sub-GHz ISM EU: 868 MHz US: 902 MHz	Sub-GHz ISM EU: 868 MHz US: 915 MHz AS: 430 MHz	2.4 GHz ISM	Sub-GHz ISM EU: 868 or 433 MHz US: 915 MHz AS: 430 MHz
Bandwidth	180 kHz	100 or 600 Hz DL: 1.5 kHz	100 kHz	1 MHz	125,250,500 kHz
Modulation	DL: QPSK UL: $\pi/4$ -QPSK, $\pi/2$ -BPSK, QPSK	DL: QPSK UL: $\pi/4$ -QPSK, $\pi/2$ -BPSK, QPSK	UNB 2-FSK	UL: RPMADSSS DL: CDMA	CSS, FSK
MAC	TDMA-based MAC	ALOHA MAC	n/a	TDMA-based MAC	LoRaWAN/ALOHA based
Protocol ownership	Standard	Proprietary	Standard	Proprietary	Partially proprietary
Data rate	UL: 64 kbps DL: 25 kbps	UL: 100 or 600 bps DL: 600 bps	UL: 62.5 bps DL: 500 bps	UL: 624 kbps DL: 156 kbps	LoRa: 0.3–37.5 kbps, FSK: 50 kbps
Range	Urban: 1.5 km Rural: 20–40 km	Urban: 3–10 km Rural: 30–50 km	Urban: 3 km Rural: 16 km (NLOS)	Urban: 15 km Rural: 48 km	Urban: 5 km Rural: 45 km
Link budget (dB)	189	EU: 162 US: 146	EU: 161 US: 149	EU: 168 US: 180	EU: 151 US: 171
Max. Payload size (bytes)	13	UL: 12 DL: 8	65k	64	250
Tx power (dBm)	35	24	14	21	21
Security	L2 security	No or encryption at higher level	Yes	AES 256 b	AES 128 b
Interference immunity	Low	Very high	Very high	Low	Very low
Adaptive data rate	No	No	No	Yes	Yes
Localization	No	Yes (RSSI)	No	No	Yes (TDOA)
Topology	Star	Star	Star or tree	Star or tree	Star of stars
Energy consumption	Low	Very low	Low	High	Very low
Link symmetry	No	No	No	No	Yes
Allow private networks	No	No	No	Yes	Yes
Over the air updates	No	No	Yes	Yes	Yes
Error Correction	CRC	UL: CRC-16 DL: CRC-8	Yes	CRC	FEC and CRC-8/16
Channel or orthogonal signal	12 carrier	360	multiple	40	EU: 10 US: UL 64+8 DL: 8 + SF
Nodes per gateway	52,000	>1,000,000	5000	500,000	>1,000,000
Cost	Moderate	Moderate	High	Low	Very low

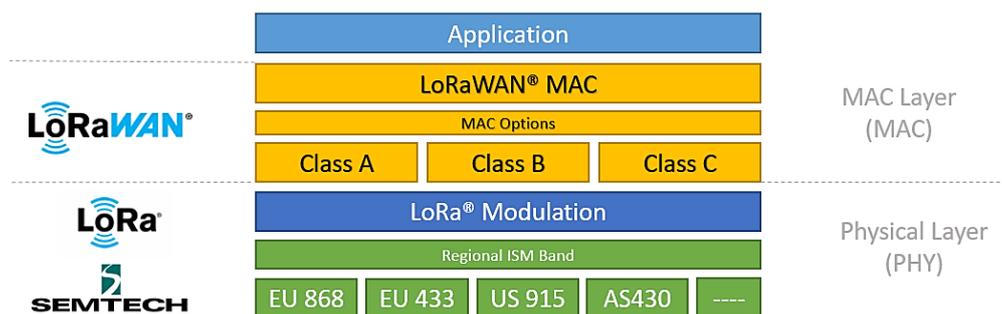
Nota. Obtenido de: Elaborado por el autor.

4.4.1 Tecnología LoRaWAN

LoRa es una tecnología emergente en el mercado actual, que opera en una banda sin licencia por debajo de 1 GHz para el funcionamiento de enlaces de comunicación de largo alcance. LoRa es un esquema de modulación de espectro ensanchado que es un derivado de la modulación de espectro ensanchado (CSS, del inglés *Chirp Spread Spectrum*) y que intercambia la velocidad de datos por sensibilidad dentro de un ancho de banda de canal fijo (Sinha et al., 2017).

LoRaWAN es un protocolo MAC de código abierto desarrollado en torno a LoRa. El Gateway en LoRaWAN utilizan bandas sin licencia ISM y se dirigen a comunicaciones de baja potencia y largo alcance. Normalmente los nodos suelen transmitir unos pocos mensajes al día.

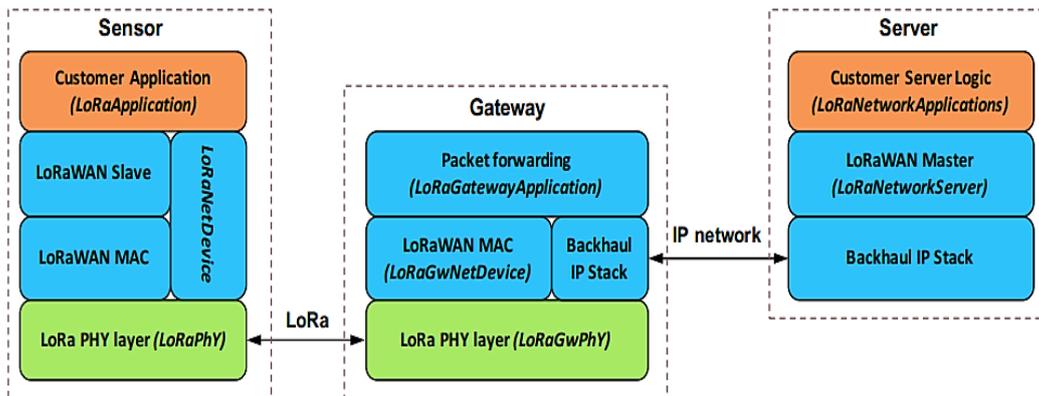
Figura 9.
Estructura de protocolos LoRaWAN



Nota. LoRaWAN utiliza las bandas ISM (EU868, EU433, US915, AS430). Fuente: (Semtech Corporation, 2020).

Es importante recalcar que LoRa pertenece a la capa física (PHY) como se ilustra en la **Figura 9**. Es decir, LoRa, es la modulación inalámbrica utilizada para crear el enlace de comunicación de largo alcance mientras que LoRaWAN es un protocolo de red abierta que ofrece servicios seguros de comunicación, movilidad y localización bidireccionales estandarizados y mantenidos por LoRa Alliance.

Figura 10.
Estructura de una red LoRaWAN: Sensor y Gateway



Nota. LoRa pertenece a la capa física (PHY). Fuente: (Reynders et al., 2018).

4.4.2 LoRa Alliance

La LoRa Alliance es una asociación sin fines de lucro y de carácter abierto que ha experimentado un rápido crecimiento desde su establecimiento en 2015. Se ha convertido en una de las alianzas más grandes en el campo de la tecnología. Sus miembros colaboran estrechamente y comparten experiencias con el objetivo de promover y respaldar el éxito del estándar LoRaWAN. Este estándar se ha posicionado como líder a nivel mundial en la conectividad segura de IoT mediante redes LPWAN.

La LoRa Alliance ofrece flexibilidad técnica para abordar una amplia variedad de aplicaciones de IoT, tanto estáticas como móviles. Además, cuenta con un programa de certificación que garantiza la interoperabilidad entre los dispositivos y soluciones que utilizan el estándar LoRaWAN. De esta manera, se promueve un ecosistema robusto y confiable para el despliegue de soluciones IoT basadas en esta tecnología (LoRa Alliance, 2022).

4.4.3 Arquitectura LoRaWAN

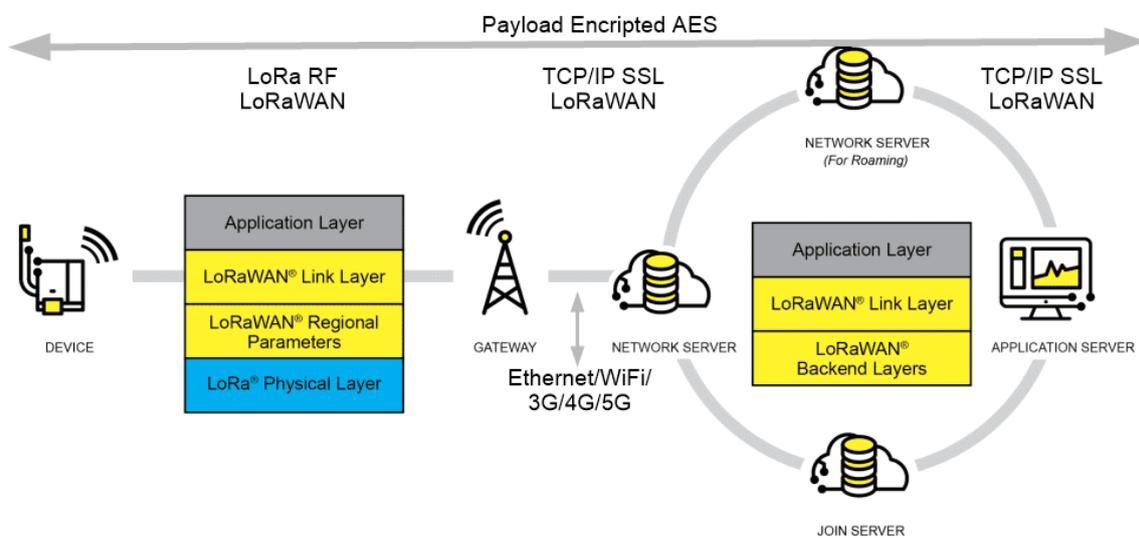
La arquitectura de LoRaWAN está compuesta por cuatro componentes fundamentales: los nodos, los Gateways, el servidor de red y el servidor de aplicación. Esta estructura se ilustra en la **Figura 11**.

En primer lugar, los dispositivos finales, también conocidos como nodos, establecen comunicación utilizando la técnica de modulación LoRa con los Gateways. A su vez, los Gateways actúan como puentes transparentes entre los dispositivos finales y el servidor de red.

En segundo lugar, el servidor de red, se comunica a través de IP con los Gateways para recopilar los datos enviados por los dispositivos finales. Posteriormente, estos datos son transferidos al servidor de aplicación, donde pueden ser visualizados y procesados.

Es importante destacar que, durante todo el proceso de comunicación, se implementa un alto nivel de seguridad mediante el estándar de cifrado avanzado (AES, del inglés *Advanced Encryption Standard*). Esto garantiza la protección de los datos transmitidos y recibidos en el sistema LoRaWAN.

Figura 11.
Arquitectura de red LoRaWAN



Nota. Arquitectura adaptada de LoRa Alliance. Fuente: (LoRa Alliance, 2021).

- **Dispositivos finales (End Nodes):** un dispositivo cliente LoRa, a veces llamado "mota" envían mensajes inalámbricos modulados LoRa al Gateway o reciben mensajes de forma inalámbrica desde el Gateway.
- **Gateway:** una radio en el lado de la infraestructura, a veces llamada "concentrador". Se comunica con los dispositivos finales y, vía IP, con un servidor de red.
- **Servidor de red (Network Server, NS):** El servidor de red termina la capa MAC de LoRaWAN para los dispositivos finales conectados a la red.
- **Servidor de unión (Join Server, JS):** es un servidor en el lado de Internet de un NS que procesa las solicitudes de unión de los dispositivos finales.
- **Servidor de aplicación (Application Server):** se refiere al código de la capa de aplicación tanto en el dispositivo final como en el que se ejecuta "detrás" de la NS.

4.4.4 Clases de dispositivos LoRaWAN

De acuerdo a LoRa Alliance (2022), LoRaWAN tiene tres clases diferentes de dispositivos finales destinadas a diferentes necesidades y características distintivas, estos son: clase A, clase B y clase C.

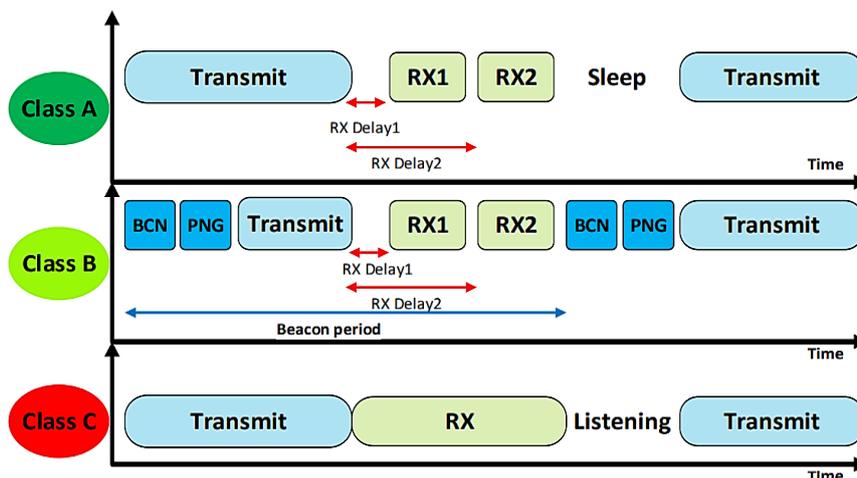
Clase A: Este es un tipo de protocolo ALOHA. La clase A siempre es iniciada por el dispositivo final y es totalmente asíncrona. Cada transmisión de enlace ascendente (UL) se puede enviar en cualquier momento y es seguida por dos ventanas cortas de enlace descendente (DL), dando la oportunidad de comunicación bidireccional o comandos de control de red si es necesario.

Clase B: la clase B se sincronizan con la red mediante balizas periódicas y abren 'ranuras de ping' de DL a horas programadas, proporcionando a la red la capacidad de enviar comunicaciones de DL con una latencia determinista y el consumo de energía adicional es lo suficientemente bajo en el dispositivo final.

Las balizas Gateway se envían periódicamente cada 128 segundos para garantizar la sincronización. La ventana de balizas utilizables ventana de baliza utilizable es un periodo de tiempo entre dos balizas. Se agrupa en 212 ranuras de ping de 30 ms, cada una de las cuales cuenta de 0 a 4095 (Almuhaya et al., 2022).

Figura 12.

Diagrama de tiempo de clase de los dispositivos LoRaWAN



Nota. Diagrama de tiempo de las clases C, B y C. Fuente: (Almuhaya et al., 2022).

Clase C: la clase C reduce aún más la latencia en el DL al mantener el receptor del dispositivo final constantemente en modo escucha todo el tiempo en que el dispositivo no está transmitiendo. El servidor de red puede iniciar una transmisión de DL en cualquier

momento suponiendo que el receptor del dispositivo final está abierto, por lo que no hay latencia.

Tabla 3.
Dispositivos LoRaWAN

Criteria	Class A	Class B	Class C
Access approach	ALOHA	Slotted ALOHA	ALOHA
Collision	High	Moderate	Moderate
Time of reception	2 s if ON	Relay on slot time	Always unless transmitting
DL latency	High	Moderate	Low
Real-time support	No	No	Yes
Most of time state	Sleeping	Beacon	Listening
Power consumption	Very low	Low	Moderate

Nota. La tabla compara las clases de dispositivos LoRaWAN. Fuente: (Almuhaya et al., 2022).

Por su parte, el método de acceso al medio ALOHA es un protocolo de acceso al medio donde las estaciones envían tramas sin verificar la disponibilidad del canal. Se espera una confirmación, y en caso de no recibirla, se realiza una retransmisión. Existen dos versiones: ALOHA puro (sin sincronización) y ALOHA ranurado (con intervalos de tiempo específicos).

ALOHA puro permite compartir el canal sin preocuparse por las colisiones, mientras que ALOHA ranurado reduce las colisiones al establecer momentos específicos para la transmisión. Ambas versiones utilizan confirmaciones para asegurar la recepción correcta de las tramas.

4.4.5 Activación de dispositivos

La especificación LoRaWAN permite dos tipos de activación: activación por aire (OTAA) y activación por personalización (ABP). La activación es crucial para establecer de manera segura los parámetros esenciales de aprovisionamiento, como identificadores, claves de cifrado y ubicaciones de servidores, tanto para el dispositivo final como para la red.

La **Tabla 4** muestra las diferentes características de cada uno de estos tipos de activación de dispositivos. De acuerdo a la especificación *LoRaWAN 1.0.3* emitido por (LoRa Alliance, 2018) menciona que ABP utiliza los siguientes parámetros:

- **End-device address (DevAddr):** dirección de dispositivo de 32 bits (no única).

- **Network session key (NwkSKey):** proporciona integridad y autenticidad. Además, se utiliza para la interacción entre el nodo y el servidor de red. Esta clave se implementa para validar la integridad de cada mensaje mediante su código de integridad del mensaje.
- **Application session key (AppSKey):** proporciona confidencialidad. Se utiliza para el cifrado y descifrado de la carga útil. La carga útil está totalmente cifrada.

Por otra parte, OTAA es la forma preferida y más segura de conectarse ya que los dispositivos finales realizan un procedimiento de unión (*Join*) con el servidor, durante el cual se asigna una dinámica y se negocian claves de seguridad con el dispositivo. OTAA utiliza los siguientes parámetros:

- **End-device identifier (DevEui):** es un ID de aplicación global en el espacio de direcciones IEEE EUI64 que identifica unívocamente a la entidad capaz de procesar la trama JoinReq.
- **Application identifier (AppEUI):** es un ID global de dispositivo final en el espacio de direcciones IEEE EUI64 que identifica de forma única el dispositivo final.
- **Application key (AppKey):** es una clave AES-128 específica del dispositivo final.

Tabla 4.
Activación de Dispositivos LoRaWAN

Over the Air Activation (OTAA)	Activation by Personalization (ABP)
<ul style="list-style-type: none"> • Los fabricantes de dispositivos generan de forma autónoma los parámetros esenciales de aprovisionamiento. • Las claves seguras (de sesión y derivadas) pueden renovarse periódicamente. • Los dispositivos pueden almacenar múltiples "identidades" para cambiar de forma dinámica y segura de red y operador durante su vida útil. • Existen opciones de seguridad de alto nivel y a prueba de manipulaciones. 	<ul style="list-style-type: none"> • Puesta en marcha asimétrica (menos segura) proceso. • Las identificaciones y las llaves se personalizan en la fabricación. • Los dispositivos se vuelven inmediatamente funcionales al encenderse; se salta el procedimiento de Join. • Los dispositivos están vinculados a una red/servicio específico.

Nota. Obtenido de: Elaborado por el autor.

4.4.6 Modulación LoRa

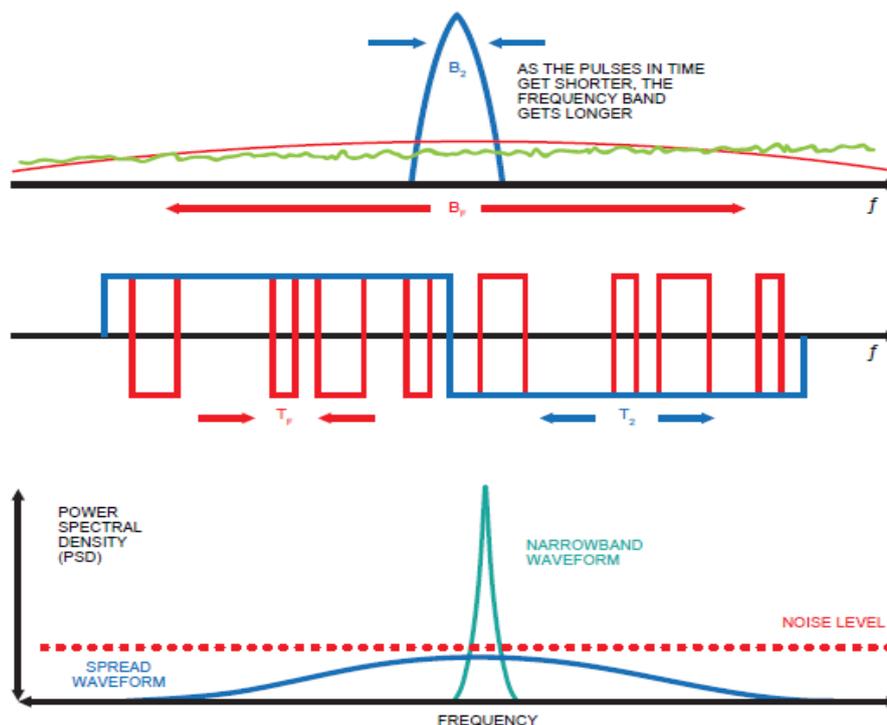
LoRa utiliza el *Chirp Spread Spectrum* (CSS) para permitir la comunicación de largo alcance. El CSS utiliza diferentes factores de dispersión (SF) para ajustar la velocidad de

datos y lograr un mayor alcance. En la especificación de LoRa, se pueden seleccionar SF7 a SF12 como opciones de factor de dispersión. El SF controla la tasa de *chirp*, lo que a su vez determina la velocidad de transmisión de datos. Los factores de dispersión más bajos implican chirridos más rápidos y, por lo tanto, una mayor tasa de transmisión de datos.

Adicionalmente, un sistema de espectro ensanchado es aquel en el que la señal transmitida se distribuye en una amplia banda de frecuencias, mucho más amplia, de hecho, que el ancho de banda mínimo requerido para transmitir la información que se envía (Harold E. Price, 1995).

En la **Figura 13**, se muestra una comparación de una señal de banda estrecha con una señal de espectro ensanchado de secuencia directa. La señal de banda estrecha se suprime al transmitir el espectro ensanchado.

Figura 13.
Señal de banda estrecha con una señal de espectro ensanchada



Nota. Comparación de una señal de banda estrecha con una señal de espectro ensanchado de secuencia directa. Fuente:(SEMTECH, 2022).

En la modulación LoRa, se logra una expansión del espectro mediante la generación de una señal *chirp* que varía de manera continua en frecuencia. Esta técnica presenta una ventaja significativa, ya que las compensaciones de tiempo y frecuencia entre el transmisor y

el receptor son equivalentes, lo que reduce considerablemente la complejidad del diseño del receptor. El ancho de banda de frecuencia de este chirrido es igual al ancho de banda espectral de la señal, lo que permite una mayor eficiencia en términos de ocupación del espectro (SEMTECH, 2015).

4.4.7 Tasa de transferencia de datos

La señal de datos deseados se fragmenta a una tasa de datos más alta y se modula en la señal de chirrido. Por ende, la modulación LoRa también incluye un esquema de corrección de errores variable mejorando la robustez de la señal transmitida a expensas de la redundancia. En la especificación “AN1200.22 LoRa™ Modulation Basics” de SEMTECH (2015) define la tasa de bits nominal de la señal de datos presentado en la **Ecuación 1**:

$$R_b = SF * ((4/(4 + CR)))/((2^{SF}/BW)) \quad \text{Ecuación 1.}$$

- **Spreading factor (SF):** define el número de bits usados para codificar un símbolo. A mayor SF, menor velocidad de transferencia tendremos, pero mayor inmunidad a interferencias. LoRa opera con valores de SF=7,8,9,10,11,12.
- **Coding rate (CR):** indica la forma de codificar para corrección de errores. Es decir, según la técnica especificada, añade símbolos de control para saber si los datos son correctos e incluso poder determinar los valores correctos. CR= code rate (1,...,4).
- **Bandwidth (BW, Hz):** indica el ancho de frecuencia que vamos a usar.

Tabla 5.
Velocidades de datos de LoRaWAN

Modulation	Spreading Spectrum	Bandwidth (kHz)	Air bit rate (kb/s)	Sensitivity (dBm)
LoRa	7	125	5.468	-124
LoRa	7	250	10.936	-122
LoRa	8	125	3.125	-126
LoRa	9	125	1.757	-129
LoRa	10	125	0.976	-132
LoRa	11	125	0.537	-134.5
LoRa	12	125	0.293	-137
FSK	-----	150	50	-122

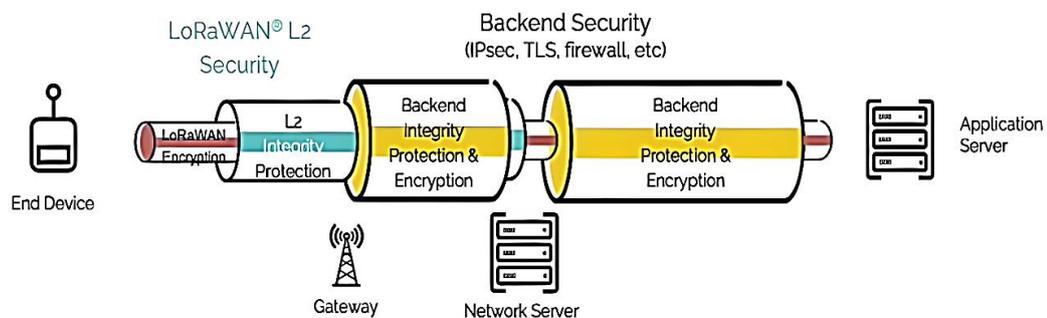
Nota. Velocidades de datos de LoRaWAN basado en los diferentes modulaciones y configuración para la banda de 868 MHz de la UE. Fuente: (Petäjajarvi et al., 2017).

4.4.8 Seguridad de LoRaWAN

Uno de los aspectos importantes en las tecnologías inalámbricas es la seguridad, LoRaWAN especifica un número de claves de seguridad. Dichas claves tienen una longitud de 128 bits. Para ello, utiliza el algoritmo AES de 128 bits para garantizar su seguridad de extremo a extremo.

Como se muestra en la **Figura 14**, LoRaWAN proporciona encriptación de datos para la transmisión de extremo a extremo desde los dispositivos finales al servidor de red mediante el uso de *Network Session Key* (NwkSKey) y desde los dispositivos finales a los servidores de aplicaciones mediante la utilización de *Application Session Key* (AppSKey).

Figura 14.
Seguridad de LoRaWAN



Nota. Arquitectura de seguridad según LoRa Alliance. Fuente: (LoRa Alliance, 2019).

LoRaWAN proporciona seguridad en dos capas diferentes. En primer lugar, en la capa de red la seguridad es proporcionada la clave de sesión de red (NwkSKey). En segundo lugar, la capa de aplicación cuya seguridad la proporciona la clave de sesión de la aplicación (AppSKey):

En las **Tabla 6** y

Tabla 7, se presentan claves en diferentes versiones de LoRaWAN. Por su parte, la

Tabla 7 representa la clave de sesión de red (NwkKey) donde se crean tres claves con diferentes propósitos.

Tabla 6.
Key in LoRaWAN version 1.0.x

Application Key (AppKey)	Application Session Key (AppSKey)
	Network Session Key (NwkSKey)

Tabla 7.*Key in LoRaWAN version 1.1.x*

Application Key (AppKey)	Application Session Key (AppSKey)
Network Key (NwkKey)	Forwarding Network Session Integrity Check (FNwkSintKey)
	Serving Network Session Integrity Key (SNwkKSintKey)
	Network Session Encryption Key (NwkSEncKey)

4.4.9 Tipos de mensajes

Los tipos de mensajes se utilizan para transportar comandos MAC y datos de aplicaciones. Por ende, LoRaWAN define varios tipos de mensajes MAC. A continuación, se presentan los tipos de mensajes MAC que se emplean en LoRaWAN. En la **Tabla 8**, se muestran los tipos de mensajes MAC que se pueden encontrar en versiones de LoRaWAN 1.0.x y 1.1.

Tabla 8.*Tipos de mensajes MAC*

LoRaWAN 1.0.x	LoRaWAN 1.1	Descripción
Join-request	Join-request	mensaje de vínculo ascendente, utilizado por el procedimiento de activación inalámbrica (OTAA)
Join-accept	Join-accept	mensaje de vínculo descendente, utilizado por OTAA
Unconfirmed Data Up	Unconfirmed Data Up	Un marco de datos de enlace ascendente, no se requiere confirmación
Unconfirmed Data Down	Unconfirmed Data Down	Un marco de datos de enlace descendente, no se requiere confirmación
Confirmed Data Up	Confirmed Data Up	Un marco de datos de enlace ascendente, se solicita confirmación
Confirmed Data Down	Confirmed Data Down	Un marco de datos de enlace descendente, se solicita confirmación
RFU	Rejoin-request	1.0.x - Reservado para uso futuro 1.1 - Solicitud de reincorporación de activación por aire (OTAA)
Proprietary	Proprietary	Se utiliza para implementar formatos de mensaje no estándar

Nota. Información obtenida de TTN. Fuente: (The things network, 2021).

4.4.10 Mensajes de datos

Existen 4 tipos de mensajes de datos utilizados tanto en LoRaWAN. Estos tipos de mensajes de datos se utilizan para transportar tanto los comandos MAC como los datos de aplicación, que pueden combinarse juntos en un solo mensaje. Los mensajes de datos pueden

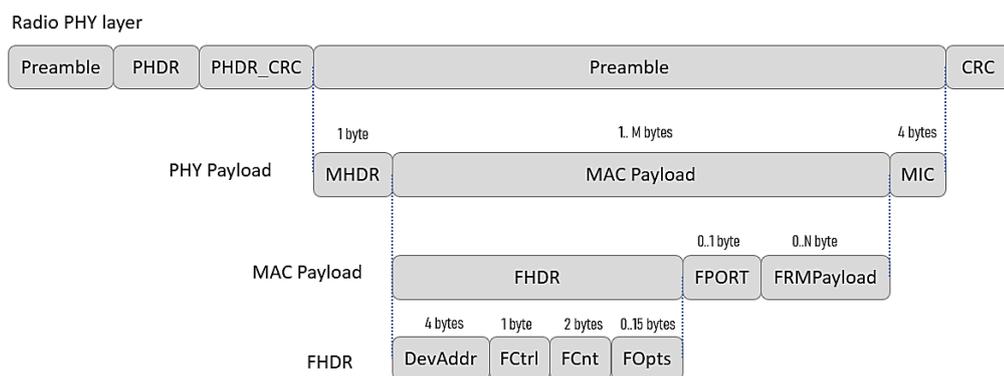
ser confirmados o no confirmados. Los mensajes de datos confirmados deben ser reconocidos por el receptor, mientras que los mensajes de datos no confirmados no necesitan ser reconocidos por el receptor (The things network, 2021).

Tabla 9.
MAC payload

7 a 22 bytes	0 a 1 byte	0 a N bytes
FHDR	FPort	FRMPayload

En la **Figura 15**, se muestra cómo se construye un mensaje de datos, la carga útil MAC (MAC payload) de los mensajes de datos consta de una cabecera de trama (FHDR) seguida de un campo de puerto opcional (FPort) y una carga útil de trama opcional (FRMPayload).

Figura 15.
Mensaje de datos



Nota. Construcción de mensajes de datos según TTN. Fuente: (The things network, 2021).

Por otra parte, la cabecera de la trama (FHDR) de la carga útil MAC consta de los campos presentados en la **Tabla 10**.

Tabla 10.
The frame header (FHDR)

4 bytes	1 byte	2 bytes	0 a N bytes
DevAddr	FCtrl	FCnt	FOpts

4.4.11 Planes de frecuencia LoRaWAN

La lista de planes de frecuencia de LoRaWAN se determina según las regulaciones de radiocomunicaciones y los planes de frecuencia específicos de cada región. Para las regiones

de EE. UU., Canadá y América del Sur, se utiliza el plan de frecuencias US902-928. En la **Tabla 11** se detalla el plan de frecuencias asignado para estas regiones.

Tabla 11.

Plan de frecuencias US902-928

	Uplink	SFBW	Downlink	SFBW
1	903.9	SF7BW125a SF10BW125	923.3	SF7BW500 a SF12BW500
2	904.1	SF7BW125 a SF10BW125	923.9	SF7BW500 a SF12BW500
3	904.3	SF7BW125 a SF10BW125	924.5	SF7BW500 a SF12BW500
4	904.5	SF7BW125 a SF10BW125	925.1	SF7BW500 a SF12BW500
5	904.7	SF7BW125 a SF10BW125	925.7	SF7BW500 a SF12BW500
6	904.9	SF7BW125 a SF10BW125	926.3	SF7BW500 a SF12BW500
7	905.1	SF7BW125 a SF10BW125	926.9	SF7BW500 a SF12BW500
8	905.3	SF7BW125 a SF10BW125	927.5	SF7BW500 a SF12BW500
9	904.6	SF8BW500	923.3	SF12BW500

4.5 Interfaz de programación de aplicaciones

Las interfaces de programación de aplicaciones (API, del inglés *Application programming interface*) son herramientas que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos. Por lo tanto, permiten conectar todo el sistema IoT, monitorear, almacenar, supervisar, procesar los datos desde la nube y ofrecer interfaces estándares al usuario.

Se han desarrollado diferentes tipos de protocolos API para proporcionar un conjunto de reglas definidas que especifican los tipos de datos y comandos aceptados. Estos protocolos permiten el intercambio de información estandarizada. Los tipos de protocolos API más usados se describen a continuación:

4.5.1 SOAP.

Simple Object Access Protocol (SOAP) es un protocolo API creado a partir de XML. Es decir, el cliente y el servidor intercambian mensajes mediante XM, esto permite a los usuarios enviar y recibir datos a través de SMTP y HTTP.

4.5.2 XML-RPC.

Extensible Markup Language Remote Procedure Call (XML-RPC) es un protocolo que se basa en un formato específico de XML para transferir datos. Permiten que el software que se ejecuta en sistemas operativos dispares, ejecutándose en diferentes entornos, realice llamadas a procedimientos a través de Internet. Es una llamada a procedimiento remoto utilizando HTTP como transporte y XML como codificación.

4.5.3 JSON-RPC.

JavaScript Object Notation Remote Procedure Call (JSON-RPC) es un protocolo similar a XML-RPC, pero este usa JSON en lugar del formato XML para transferir datos. Es decir, utiliza el protocolo HTTP para llamadas a procedimientos remotos y JSON para la representación de datos. Debido a que JSON-RPC se basa en HTTP, no requiere que ejecute un servidor web independiente. Solo se necesita tener al menos un navegador web con conexión a Internet.

4.5.4 REST.

Representational State Transfer (REST) es un conjunto de principios de arquitectura de API web. Su ventaja es que no tiene estado, esto significa que los servidores no guardan los datos del cliente entre las solicitudes. REST admite la comunicación bidireccional entre las aplicaciones cliente y el servidor.

4.5.5 WebSocket.

El protocolo Web Socket utiliza objetos JSON para transmitir datos, admitiendo una comunicación bidireccional entre las aplicaciones cliente y el servidor. El servidor puede enviar mensajes de devolución de llamadas a los clientes conectados, siendo más eficiente que el protocolo REST.

Por otra parte, en la **Tabla 12**, se presenta una recopilación de APIs orientados específicamente a IoT. Las variables a considerar para la elección de una API dependen del tipo de aplicación y dimensión del proyecto, para este caso de estudio se discurre que sea de código abierto, sin licencia, que provea seguridad avanzada y compatible con los protocolos de comunicación.

El servidor de red que se adapta al proyecto es The Things Network (TTN), se trata de una iniciativa basada en la comunidad para establecer una red global de IoT, su principal objetivo es tener redes LoRaWAN instaladas en todas las ciudades del mundo y donde

cualquiera pueda participar en proyectos IoT. Algunos ejemplos de proyectos pueden incluir: estaciones meteorológicas, monitoreo ambiental, seguimiento mediante GPS entre otros.

Tabla 12.
Comparativa de APIs para IoT

API	Open source	Free	scope	Device management	Integration	Security	Protocols for data collection
Actility Thing Park	No	No gateway and 5 devices	1 All	Yes (needs gateways)	REST and Real-time APIs	LoRaWAN	LoRaWAN, LTE-M, NB IoT
AirVantage	Yes	Free Trial	Cloud	Yes (needs gateways)	REST API	Unknown	MQTT, CoAP
Apache 2.0 Arm MBED device platform ARM Pelion IoT Platform	Yes	Yes	All	Yes (Pelion Device Management)	REST API, SDK	SSL/TLS	CoAP, LWM2M NB IoT, BLE, 6LoWPAN
Appcelerator (part of AMPLIFY/axway)	Yes	Yes	Cloud	No	REST API	Link Encryption (SSL,Ipsec, AES-256)	MQTT, HTTP
AWS IoT platform (Amazon Web Services)	No	No	all	Yes	REST API	Link Encryption (TLS), Authentication (SigV4, X.509)	MQTT, HTTP 1.1, WebSockets
Ayla Network	No	Yes	All	Yes (needs gateways)	REST API	TLS, AES-128	HTTP, ZigBee, Z-Wave, BLE
Carriots / Altair SmartWorks™	Yes	Up to 2 devices	All	yes	REST API, (XML, JSON)	(HTTPs) TCP, TSL	MQTT, HTTP
Cisco IoT Cloud Connect	No	No	All	Yes (needs gateways)	Unknown	Unknown	Unknown
Eclipse IoT Projects	Yes	Yes	All	Yes	HTTP JSON API	SSL/TLS	MQTT CoAP LWM2M
EcoStruxture	No	No	All	Yes (needs gateways)	Unknown	Unknown	Unknown
EveryWare IoT	Yes	No	All	Yes (needs gateways)	REST API	SSL	MQTT v 3.1
EVERYTHING - IoT Smart Products Platform	No	No	Cloud	No	REST API	Link Encryption (SSL)	MQTT, CoAP, WebSockets
Exosite (Murano)	No	No	All	Yes	REST API	Link Encryption (SSL)	CoAP, WebSocket
Google Cloud Platform (GCP)	No	Free Trial	Cloud	Yes (needs gateways)	REST API	JSON Web Tokens, RSA or Elliptic Curve algorithms, TLS 1.2, Cloud Identity and Access Management (IAM)	MQTT, HTTP
IBM IoT Foundation Device Cloud IBM Watson	No	Free trial	All	Yes	REST and Real-time APIs	Link Encryption (TLS), Authentication (IBM Cloud SSO), Identity management (LDAP)	MQTT, HTTPS
Idigi Device Cloud	Yes	Yes	All	Yes (needs	REST API	SSL	MQTT

						gateways)			
Kaa	Yes	Yes	All	Yes		REST API JSON	Encryption Authentication	MQTT CoAP	
Legato	Yes	Yes	All	No		REST API	Encryption	HTTP	
Lelylan	Yes	Yes	All	Yes		REST API	Link Encryption (SSL/TLS), Authentication (SIM based)	MQTT, WebSocket	
mangOH Yellow	Yes	Yes	All	Yes (needs gateways)		REST API	Unknown	MQTT, CoAP	
Microsoft Azure IoT Suite	No	Student	All	Yes		REST API	Link Encryption (SSL/TLS)	HTTP, AMQP, MQTT	
Mindsphere Siemens)	(by No)	Demo	All	Yes		REST API	Encryption	HTTP MQTT AMQP LoRaWAN Modbus CoAP XMPP 6LowPAN LWM2M	
Sensor Cloud	No	Yes	All	Yes (needs gateways)		REST API	TLS	HTTP	
Temboo	No	Yes	All	Yes		REST API	TLS	MQTT, CoAP	
The Things Network	Yes	Yes	All	Yes		API	TCI/IP SSL/AES LoRaWAN	MQTT LoRaWAN	
Thinger	Yes	Yes (2 devices)	All	Yes		REST API	SSL / TLS	Protoson	
Things Board	Yes	Yes (Community edition No Professional edition)	All	Yes (needs gateways)		REST API	SSL	MQTT, CoAP, HTTP, OPC-UA, Sigfox, Modbus, NB IoT, LoRaWAN	
ThingSpeak	Yes	Yes (1 device)	All	Yes		REST API	SSL	MQTT, CoAP, WebSockets	
ThingWorx	No	Free Trial (120 days)	All	Yes		REST API	Standards (ISO 27001), Identity management (LDAP) TLS/AES 128	MQTT, AMQP, CoAP, XMPP, DDS, WebSockets	
Ubidots	No	No	All	Yes		REST API	Unknown	HTTP, MQTT, TCP, UDT	

Nota. Obtenido de: Elaborado por el autor.

5. Metodología

Método documental. – fue usado con el objetivo de recopilar información útil para el desarrollo del diseño de la red de sensores. Así mismo, se investigaron trabajos relacionados en los que se esté desarrollando soluciones similares al presente diseño. Se buscaron métodos de programación de los elementos utilizados en sistemas WSN, capas de protocolos y las variables físicas que son necesarias para el monitoreo de puentes debidamente justificados.

Método de ensayo y error. -Empleado para el desarrollo del enlace de comunicación y su funcionamiento para transmisión los datos mediante la tecnología LoRa. Se realizaron pruebas de funcionamiento del todo el sistema WSN para evitar inconvenientes al momento de realizar las pruebas del sistema WSN en el escenario elegido para el monitoreo estructural.

Para la presente propuesta de este proyecto se utilizó una metodología de cuatro fases, mismas que permitieron desarrollar el prototipo de manera eficiente y organizada con el propósito de obtener los resultados deseados.

- **Fase de investigación y adquisición:** consistió en realizar una indagación general acerca de las redes LPWAN. Además, se realizó una investigación de los parámetros más importante en el monitoreo del estado de salud estructural de puentes. También, se adquirieron los módulos LoRa, así como los respectivos sensores.
- **Fase de diseño:** permitió diseñar el sistema de monitoreo para el estado de salud estructural de puentes. El proceso del diseño incluyó los sistemas de adquisición y acondicionamiento de las señales provenientes del acelerómetro, anemómetro, y termómetro.
- **Fase de construcción e implementación:** consistió en realizar la construcción del prototipo para el monitoreo del estado de salud estructural de puentes. El objetivo fue construir las placas PCB (Printed Circuit Board), construcción de la carcasa, soporte de los nodos y los códigos de programación para el sistema.
- **Fase de pruebas:** En la fase 4 se procedió a realizar las pruebas de campo que permitan comprobar el funcionamiento del sistema. En general, en la fase de pruebas se verificó el funcionamiento del prototipo y enviar los datos a la API para su procesamiento, análisis, almacenamiento y visualización de los datos obtenidos de los nodos.

5.1 Diseño del sistema WSN

En esta sección se analiza el escenario elegido para el sistema de monitoreo del estado de salud estructural de puentes, a través de simulaciones de los enlaces de comunicaciones. Además, se diseña la arquitectura del sistema de monitoreo.

5.1.1 Análisis del Escenario

El escenario donde se realizó las pruebas de funcionamiento del sistema de monitoreo estructural de puentes está ubicado en la provincia de Zamora Chinchipe, Cantón Zamora. Una de las principales razones por la que se eligieron estos escenarios es debido a que la provincia de Zamora Chinchipe se encuentran dos grandes empresas mineras a través de los proyectos “Cóndor Mirador y Fruta del Norte”, los dos megaproyectos con los que Ecuador inauguró la minería a gran escala.

El escenario para las pruebas de funcionamiento del sistema de monitoreo se ilustra en la **Figura 16**, este puente es de tipo vehicular y peatonal. Se encuentra ubicado en el cantón Zamora, está construido de hormigón con cables de acero colgantes, sus dimensiones son de 113 metros de largo y 10.7 metros de ancho. Este puente es uno de los más antiguos de la ciudad de Zamora y es usando mayormente para carga pesada y cruce de peatones. Su ubicación es $4^{\circ}04'13.5''S$ $78^{\circ}56'56.6''W$.

Figura 16.

Puente sobre el Río Bombuscaro



Nota. Obtenido de: Elaborado por el autor.

5.1.2 Simulación de enlaces

Para la simulación del enlace de comunicación entre el Gateway y el nodo transmisor se ha optado por el software Radio Mobile para obtener datos acerca de la línea de vista entre las entidades y sus respectivos parámetros de conexión entre otros.

Ubicación del Gateway:

Analizando el escenario, se ha hecho un recorrido por el entorno para ubicar la mejor posición del Gateway, por lo que se ha optado por ubicar en la siguiente coordenada: 4°04'16.0"S 78°57'04.4"W.

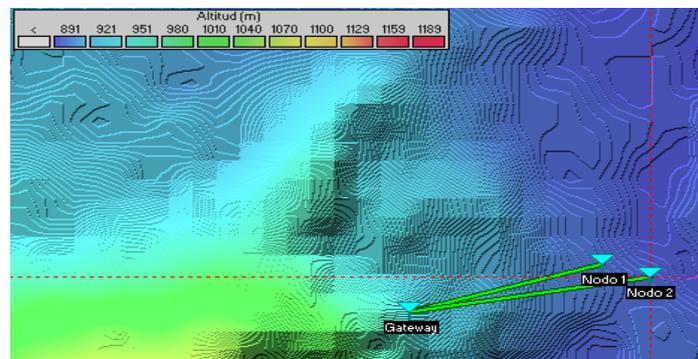
Perfil de elevación del Gateway y nodos:

En la **Figura 17**, se presenta la simulación de los enlaces de comunicaciones del Gateway y los nodos. Los nodos están a 5 metros de altura y se ubican en las siguientes coordenadas:

- **Nodo 1:** 4°04'13.4"S 78°56'56.6"W.
- **Nodo 2:** 4°04'14.1"S 78°56'54.6"W.

Figura 17.

Simulación de enlaces del Gateway y los nodos

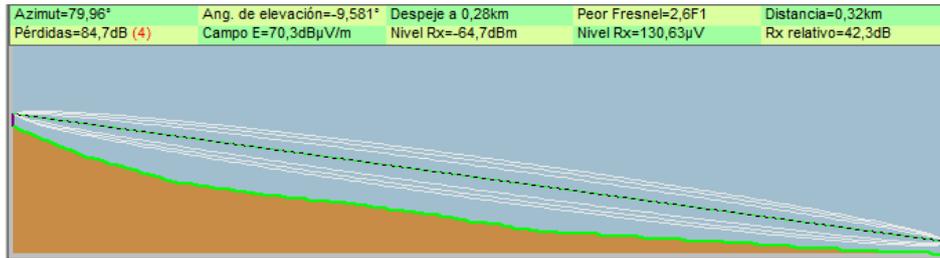


Nota. Obtenido de: Elaborado por el autor.

El perfil de elevación del enlace de comunicación entre el Gateway y el Nodo1 se ilustra en la **Figura 18**, donde se observa que su enlace tiene una línea de vista sin obstáculos en su comunicación.

Figura 18.

Perfil de elevación entre el Gateway y Nodo 1

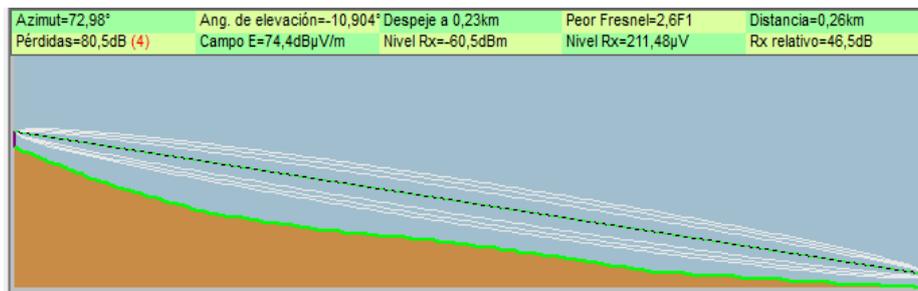


Nota. Obtenido de: Elaborado por el autor.

Adicionalmente, en la **Figura 19** se simula el enlace de comunicación entre el Gateway y el Nodo 2. Su comunicación tiene línea de vista lo que colabora que su conectividad no se vea afectada por los obstáculos.

Figura 19.

Perfil de elevación entre el Gateway y Nodo 2



Nota. Obtenido de: Elaborado por el autor.

Con el objetivo de colaborar para que los enlaces de comunicación no tengan obstáculos en el camino de propagación entre el Gateway y los nodos. En la **Figura 20**, se presenta una simulación de comunicación realizada en Google Earth, donde se visualiza que sus enlaces tienen línea de vista.

Figura 20.

Enlace de comunicación del sistema de monitoreo.



Nota. Obtenido de: Elaborado por el autor.

Concluyendo con la simulación de los enlaces de comunicación del sistema de monitoreo. En la **Figura 21**, se presenta el diagrama de cobertura del Gateway donde se visualiza que su radiación cubre todo el escenario a censar.

Figura 21.

Diagrama de cobertura del Gateway



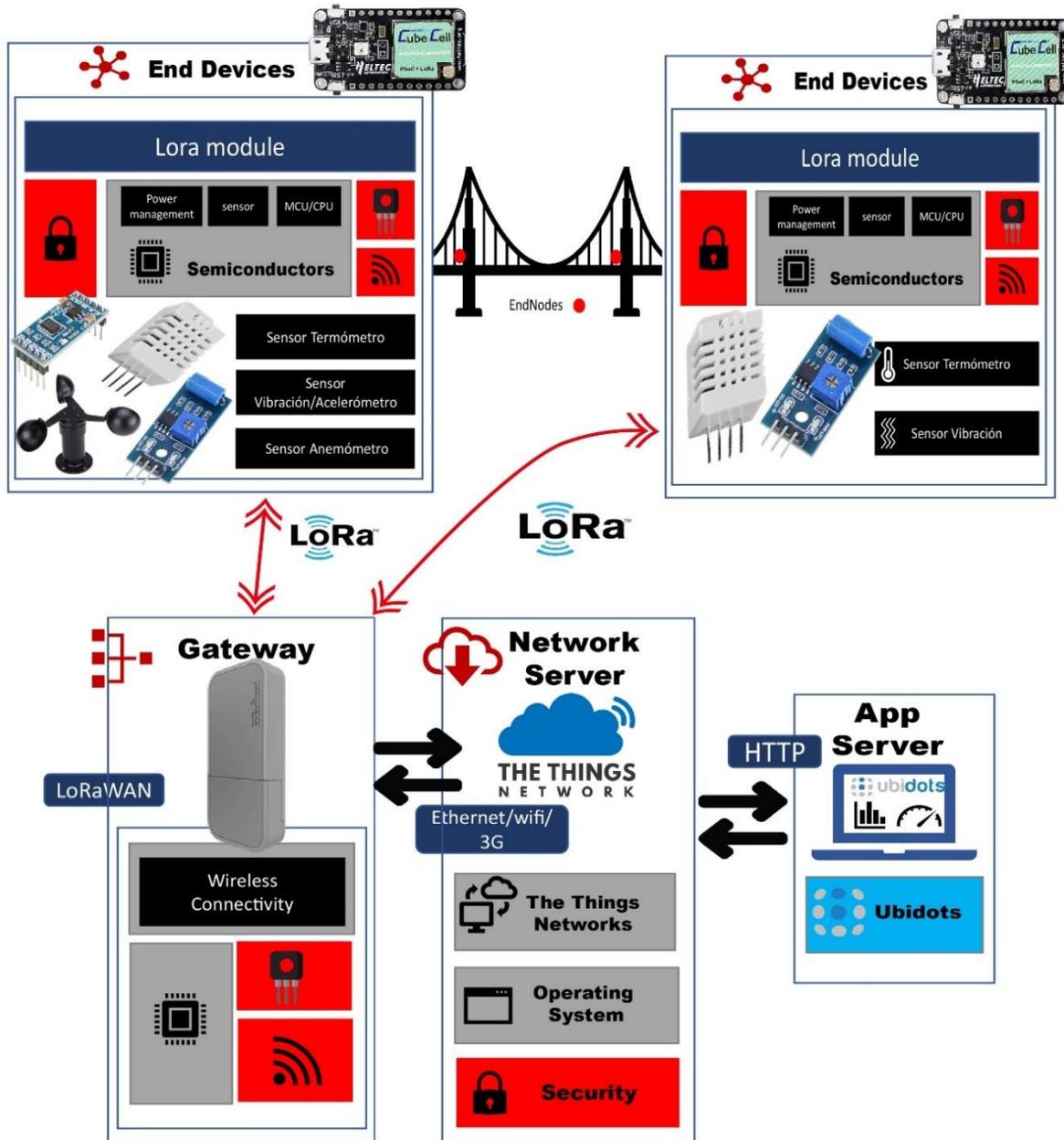
Nota. Obtenido de: Elaborado por el autor.

5.2 Arquitectura del sistema de monitoreo

La arquitectura del sistema de monitoreo del estado estructural de puentes se ilustra en la **Figura 22**. Su diagrama de red está compuesto en 3 bloques: Nodos finales, Gateway y server o red troncal. A continuación, se describe cada uno de ellos.

Los nodos incluyen los sensores de vibración, DHT22 y el anemómetro, los cuales se conectan al módulo de comunicación para enviar los datos a través de LoRaWAN. El Gateway, por su parte, está compuesto por una unidad receptora que se encarga de transportar los datos de los nodos hacia la red troncal o servidor de red.

Figura 22.
Arquitectura del Sistema de monitoreo Estructural de Puentes

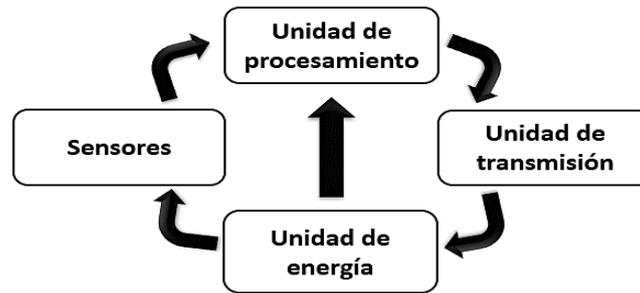


Nota. Obtenido de: Elaborado por el autor.

5.2.1 *Nodo final*

El nodo final está integrado en 4 componentes las cuales son: la unidad de procesamiento, unidad de transmisión, sistema de alimentación y los sensores. El diagrama de bloques del nodo se muestra en la **Figura 23**.

Figura 23.
Nodo Sensor



Nota. Obtenido de: Elaborado por el autor.

La unidad de energía proporciona la alimentación necesaria para el funcionamiento del nodo sensor, suministrando energía a los componentes y sensores. La unidad de procesamiento, generalmente un microcontrolador, se encarga del procesamiento de los datos recopilados por los sensores. Por otro lado, la unidad de comunicación, que consta de un módulo transmisor, tiene la tarea de enviar los datos obtenidos desde los sensores a través de LoRa. En la **Tabla 13**, se aprecia con mejor detalle los sensores que integran cada nodo.

Tabla 13.
Nodos del sistema WSN

Sensores	Nodos
DHT22	Nodo 1
SW-420	
ADXL345	
Anemómetro	Nodo 2
DHT22	
SW-420	

Nota. Obtenido de: Elaborado por el autor.

5.2.2 Gateway

El Gateway permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación. Su principal objetivo es traducir la información del protocolo utilizado en una red al protocolo usado en la red de destino (NAT, del inglés *Network Address Translation*). Esta capacidad de traducción de direcciones permite aplicar una técnica llamada enmascaramiento de IP. En términos de IoT, el Gateway conecta sensores, módulos de comunicación inalámbrica IoT y dispositivos inteligentes a la nube (servidores IoT).

5.2.3 Sensor de vibración

Un acelerómetro es un dispositivo que proporciona la capacidad de medir y analizar la aceleración lineal y angular, son utilizados en mediciones de aceleración gravitacional estática, lo que le permite determinar el ángulo de desviación del objeto medido de la vertical, así como en mediciones de aceleración dinámica debido a golpes, movimiento, impacto o vibración, es decir, vibraciones de baja amplitud y baja frecuencia, que alcanzan varias docenas de Hz.

Los acelerómetros es un tipo representativo de sensor utilizado en la medición de la vibración de miembros estructurales, mide la vibración de cables colgantes, pilones, armaduras de refuerzo en SHM (*Structural Health Monitoring*). Varían mucho en tipos dependiendo de dónde se mida, se someten a un examen en profundidad del rango de medición y la resolución para garantizar la aplicación del uso apropiado de la estructura de los acelerómetros. En la **Tabla 14**, se describen las características de un acelerómetro.

Tabla 14.
Características de un acelerómetro

Característica	Descripción
Medición de aceleración	Mide la aceleración en una o varias direcciones, en unidades como metros por segundo cuadrado (m/s^2) o gravedades (g).
Rango de medición	Intervalo de aceleraciones que el acelerómetro puede medir de manera precisa.
Sensibilidad	La mínima aceleración que el acelerómetro puede detectar y medir con precisión.
Resolución	La mínima aceleración que el acelerómetro puede detectar y medir con precisión.
Frecuencia de respuesta	Rango de frecuencias de vibración o movimiento que el acelerómetro puede detectar y medir.
Tipo de acelerómetro	Puede ser piezoeléctrico, capacitivo, MEMS (Microelectromechanical Systems), entre otros.
Número de ejes	Indica la cantidad de ejes en los que el acelerómetro mide la aceleración (1, 2 o 3 ejes).
Interfaz y salida de datos	Algunos acelerómetros tienen salidas analógicas (voltaje), mientras que otros son digitales (I2C, SPI).
Consumo de energía	La cantidad de energía que el acelerómetro requiere para funcionar.

Nota. Obtenido de: Elaborado por el autor.

5.2.4 Anemómetro

Un anemómetro es un instrumento meteorológico empleado para la medición de la velocidad del viento con el objetivo de estudiar el recurso, pronosticar el clima o hacer evaluaciones específicas. Los anemómetros miden las cargas de viento que actúan sobre las estructuras, de las cuales los sensores de paletas y los sensores ultrasónicos son los más comunes.

Según con (Madeleine Renom, 2011), es común expresar la velocidad del viento como un vector bidimensional que considera únicamente la dimensión horizontal, ya que la componente vertical cerca de la superficie es generalmente insignificante. Sin embargo, en algunos casos el componente vertical es importante y existen hoy en día anemómetros diseñados para medir las 3 componentes. El vector se puede escribir en sus componentes ortogonales: u , v y w .

Otra alternativa es escribir el vector con su intensidad y dirección, en el caso del vector horizontal, la dirección del viento es la dirección desde donde el viento viene medido en grados en sentido del reloj desde el Norte. En contraste, las unidades estándares para la medición de la velocidad del viento son m/s, Km/H y nudos (kt). En la **Tabla 15**, se narran las principales características de un anemómetro.

Tabla 15.
Características de un anemómetro

Característica	Descripción
Medición de velocidad	Mide la velocidad del viento en unidades como metros por segundo (m/s), kilómetros por hora (km/h) o nudos.
Medición de dirección	Algunos anemómetros también pueden indicar la dirección del viento, como norte, sur, este, oeste, etc.
Precisión	Capacidad del anemómetro para suministrar mediciones cercanas al valor real de la velocidad del viento.
Resolución	La menor diferencia de velocidad de viento que el anemómetro puede detectar y mostrar.
Tiempo de respuesta	Tiempo que tarda el anemómetro en mostrar una lectura estable después de cambios en la velocidad del viento.
Rango de velocidad	Intervalo de velocidades de viento en el que el anemómetro puede medir correctamente.
Tipo de anemómetro	Puede ser de diferentes tipos, como anemómetros de copa, de hélice, de hilo caliente, ultrasónicos, etc.
Material de construcción	Los anemómetros están contruidos con materiales

resistentes y duraderos para su uso en exteriores.

Nota. Obtenido de: Elaborado por el autor.

5.2.5 Termómetro

El termómetro se trata de un instrumento para medir la temperatura sea corporal o ambiental. Se estima que el inventor del termómetro fue Galileo Galilei, astrónomo y físico italiano, jefe de Matemática en la Universidad de Padua, considerado uno de los padres de la ciencia moderna.

Existen varios tipos de termómetros, por ejemplo, el tipo de termopar, el tipo de detector de temperatura de resistencia (RTD) y el tipo de resistencia térmicamente sensible (termistor) son las categorías de termómetros más utilizadas, que vienen en una amplia variedad de modelos y productos, dependiendo de lo que se quiera medir.

Dado que los requisitos para las especificaciones de los termómetros en SHM incluyen un rango de medición entre $-40\text{ }^{\circ}\text{C}$ y $60\text{ }^{\circ}\text{C}$ o superior y una resolución de $0,5\text{ }^{\circ}\text{C}$ o superior, se podría utilizar cualquier tipo que satisfaga dichas especificaciones. En la **Tabla 16**, se describen las características esenciales de un termómetro.

Tabla 16.
Características de un termómetro

Característica	Descripción
Sensor de temperatura	Elemento sensible a los cambios de temperatura que permite medir y detectar los cambios en el calor.
Escala de medición	Escala numérica que indica la temperatura en grados Celsius ($^{\circ}\text{C}$), Fahrenheit ($^{\circ}\text{F}$) o Kelvin (K).
Precisión	Capacidad del termómetro para proporcionar mediciones cercanas al valor real de la temperatura.
Resolución	La menor diferencia de temperatura que el termómetro puede detectar y mostrar.
Tiempo de respuesta	Tiempo que tarda el termómetro en mostrar una lectura estable después de exponerlo a una temperatura diferente.
Rango de temperatura	Intervalo de temperaturas en el que el termómetro puede funcionar correctamente.
Material de construcción	Material del que está hecho el termómetro, como vidrio, plástico o metal.
Tipo de termómetro	Diferentes tipos de termómetros, como de mercurio, digitales, infrarrojos, de resistencia, etc.

Nota. Obtenido de: Elaborado por el autor.

5.3 Simulación de la WSN en Omnet++

En la simulación de redes LoRaWAN, existen varios simuladores disponibles, cada uno con sus propias características técnicas. OMNeT++, ns-3 y LoRaSim son tres opciones populares que ofrecen diferentes enfoques y capacidades. OMNeT++ se destaca por su enfoque modular y orientado a componentes, ns-3 proporciona un conjunto completo de herramientas de simulación de redes y LoRaSim se centra específicamente en la simulación de redes LoRaWAN. A continuación, se presenta una breve comparación de estas herramientas para seleccionar la más adecuada para la investigación.

OMNeT++ se destaca por su enfoque modular y orientado a componentes, lo que facilita la reutilización de código y la creación de simulaciones personalizadas. Proporciona un marco de trabajo flexible y extensible para el desarrollo de simulaciones de redes.

En contraste, ns-3 es un simulador de redes de propósito general que incluye un módulo para simular redes LoRaWAN. Ofrece una amplia gama de modelos de comunicación y protocolos de red, lo que brinda una gran flexibilidad para diseñar y simular redes LoRaWAN completas.

Finalmente, LoRaSim es un software de simulación de redes LoRaWAN específico, desarrollado en MATLAB y Python. Permite simular el rendimiento de redes LoRaWAN en diferentes escenarios y ajustar parámetros como la distancia, la atenuación de la señal y la interferencia.

En resumen, OMNeT++ se destaca por su enfoque modular y orientado a componentes, lo cual permite una mayor flexibilidad y reutilización de código. Además, es compatible con el sistema operativo Windows 10, lo cual lo hace una opción adecuada para la investigación.

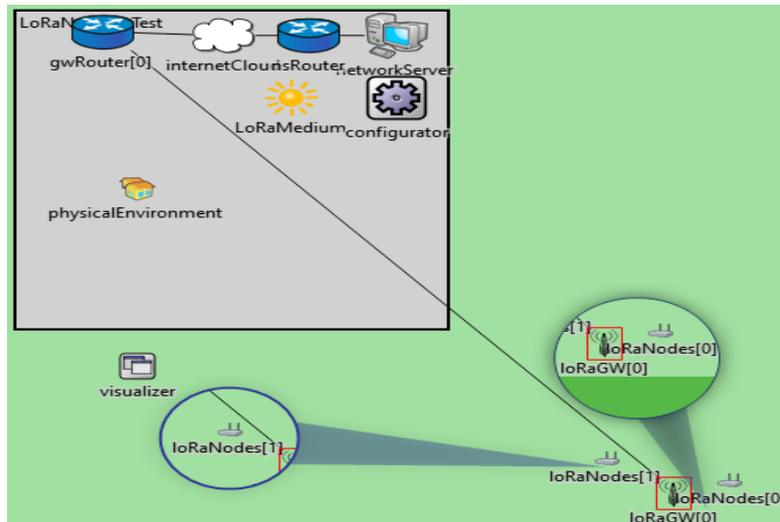
A continuación, se realiza una simulación en Omnet++ utilizando FLoRa (*Framework for LoRa*). FLoRa es un framework de software para llevar a cabo simulaciones end-to-end de redes de largo alcance (LoRa), permite la creación de redes LoRa con módulos para nodos LoRa, puertas de enlace y un servidor de red. Además, es posible obtener indicadores de rendimiento como la tasa de extracción de datos (DER), energía consumida por nodos, pérdida de paquetes, entre otras.

En la **Figura 24**, se muestra la simulación en omnet++ donde integra los submódulos LoRaNode (dos nodos), LoRaGW (un solo Gateway), visualizer, gwRouter, internetCloud,

nsRouter, LoRaMedium, Configuraton, physicalEnvironment y networkServer. Por su parte, en los parámetros importantes a tener en cuenta en la simulación es el ancho de banda, code rate y el factor de dispersiones en cada nodo.

Figura 24.

Escenario de simulación en Omnet++



Nota. Obtenido de: Elaborado por el autor.

En cuanto a los parámetros generales de la simulación se resumen en la **Tabla 17**, el ancho de banda en la que funcionará es de 125 kHz con un code rate de 4. La potencia de transmisión de cada nodo dependerá del módulo de comunicación a emplearse, en base a las especificaciones técnicas de la **Tabla 29** del CubeCell HTCC-AB01 trabajará con potencias de transmisión de $2\text{dBm} \pm 18\text{dBm}$. Para mayor información del código de programación de la simulación en Omnet++ ver **Anexo 2**, **Anexo 3**, **Anexo 4** y **Anexo 5**.

Tabla 17.

Parámetros generales de la simulación en Omnet++

Sección	Valor
Ancho de banda	125 Khz
Code rate	4
Número de nodos	2
Número de Gateway	1
Puerto local GW	2000
Puerto de destino GW	1000
Puerto local NetworkServer	2000
Puerto de destino NetworkServer	1000
Tiempo de ejecución de la simulación (en días)	1d
Resolución de simulación	-11

Número de repeticiones	2
Factor de dispersión (SF)	7,8,9,10,11,12
Potencia de transmisión (TP)	2dBm±18dBm

Nota. Obtenido de: Elaborado por el autor.

5.4 Pruebas de funcionamiento de sensores

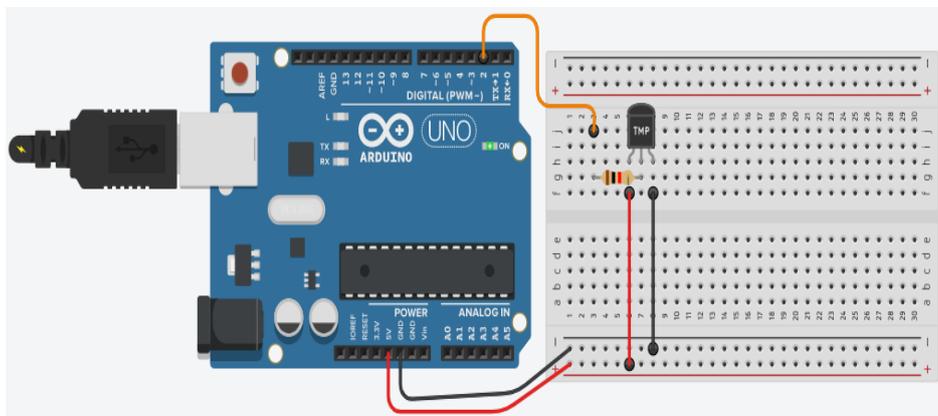
En esta sección se realizan las pruebas de funcionalidad de los componentes sensores con el objetivo de colaborar con su estado de funcionamiento. Se utiliza el software Tinkercad⁶ para realizar las simulaciones y verificación del código fuente de los sensores. Adicionalmente, con el software Arduino IDE⁷ se envía el código fuente al Arduino Uno para obtener los datos de cada sensor.

5.4.1 Sensor de temperatura y humedad DHT22

En la **Figura 25**, se presenta el diagrama de prueba de funcionamiento del sensor DHT22 realizado en Tinkercad. Para la simulación se ha requerido usar una resistencia de 1 K Ω , de la misma manera, el pin de datos del sensor está conectado al Arduino en el pin 2 digital.

Figura 25.

Diagrama de prueba sensor DHT22



Nota. Obtenido de: Elaborado por el autor.

En la **Tabla 18**, se muestra el código fuente del sensor DHT22, adicionalmente se muestran los datos obtenidos del mismo. Los resultados corroboran que el sensor DHT22 está funcionando con normalidad.

⁶ Autodesk. (2023). Tinkercad. [Tinkercad | Panel principal](#)

⁷ Arduino. (2023). Arduino IDE. [Software | Arduino](#)

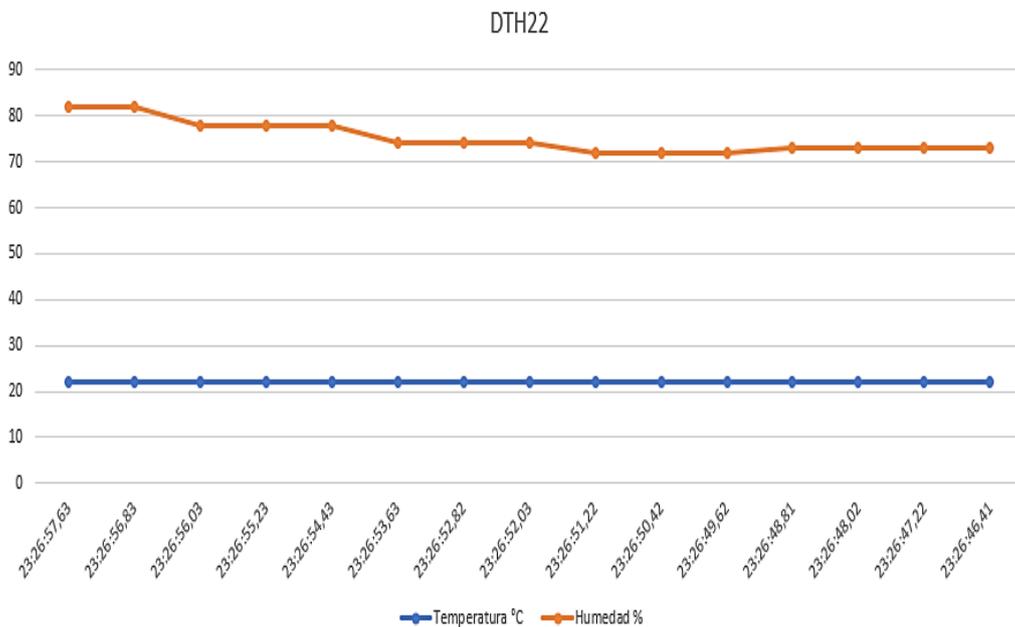
Tabla 18.
Código fuente DTH22

<pre>#include <DHT.h> #include <DHT_U.h> int temperatura; int humedad; DHT dht(2,DHT22); void setup() { Serial.begin(9600); dht.begin (); } void loop() { temperatura=dht.readTemperature(); humedad=dht.readHumidity(); Serial.print(" Temperatura: "); Serial.print(temperatura); Serial.print(" Humedad: "); Serial.println(humedad); }</pre>	<pre>Temperatura: 23 Humedad: 67 Temperatura: 23 Humedad: 69 Temperatura: 23 Humedad: 69 Temperatura: 23 Humedad: 69 Temperatura: 23 Humedad: 69 Temperatura: 23 Humedad: 69</pre>
--	--

Nota. Obtenido de: Elaborado por el autor.

En la **Figura 26**, se presentan los datos obtenidos de las pruebas de campo del sensor. Las gráficas representan dos variables, temperatura dados en grados Celsius y humedad dado en porcentaje.

Figura 26.
Prueba de Campo del DTH22



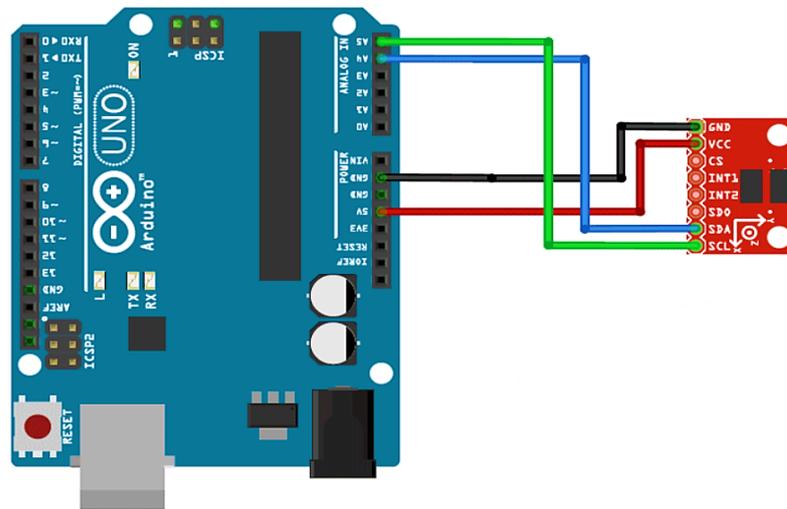
Nota. Obtenido de: Elaborado por el autor.

5.4.2 Acelerómetro ADXL345

En la **Figura 27**, se presenta el diagrama de prueba de funcionamiento del ADXL345. El pin SDA va conectado al pin A4, el pin SCL conectado al pin A5, GND Y VCC van conectados al Arduino Uno en los pines de alimentación.

Figura 27.

Diagrama de prueba ADXL345



Nota. Obtenido de: Elaborado por el autor.

La **Tabla 19**, se presenta el código fuente programado para el sensor ADXL345. Se han utilizado las librerías Wire.h, Adafruit_Sensor.h y Adafruit_ADXL345_U.h. Se ha incluido los datos obtenidos del mismo, donde se colabora que el estado de funcionamiento del sensor es normal por lo que es adecuado integrarlo en el diseño del sistema de monitoreo estructural de puentes.

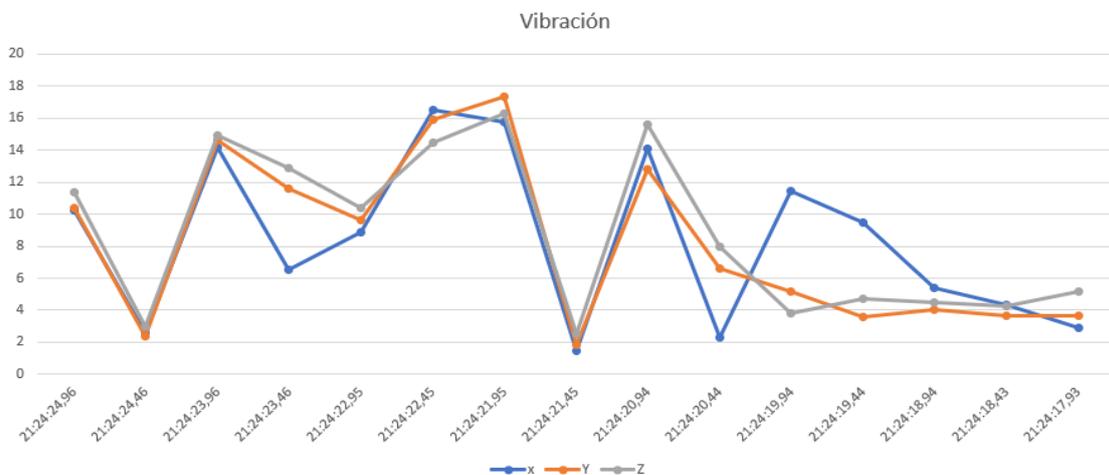
Tabla 19.
Código fuente ADXL345

<pre>#include <Wire.h> #include <Adafruit_Sensor.h> #include <Adafruit_ADXL345_U.h> Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(); void setup(void) { Serial.begin(9600); if(!accel.begin()){ Serial.println("No valid sensor found"); while(1);}} void loop(void) { sensors_event_t event; accel.getEvent(&event);Serial.print("X: "); Serial.print(event.acceleration.x); Serial.print(" ");Serial.print("Y: "); Serial.print(event.acceleration.y); Serial.print(" ");Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print(" "); Serial.println("m/s^2 ");delay(500);</pre>	<pre>X: 0.08 Y: 0.47 Z: 10.90 m/s^2 X: 0.12 Y: 0.47 Z: 10.94 m/s^2 X: 0.08 Y: 0.47 Z: 10.98 m/s^2 X: 0.12 Y: 0.47 Z: 10.94 m/s^2 X: 0.08 Y: 0.43 Z: 10.94 m/s^2 X: 0.12 Y: 0.43 Z: 11.02 m/s^2 X: 0.08 Y: 0.51 Z: 10.98 m/s^2 X: 0.08 Y: 0.47 Z: 10.94 m/s^2 X: 0.12 Y: 0.51 Z: 10.98 m/s^2 X: 0.12 Y: 0.51 Z: 10.94 m/s^2 X: 0.08 Y: 0.51 Z: 10.98 m/s^2 X: 0.12 Y: 0.47 Z: 10.94 m/s^2 X: 0.12 Y: 0.47 Z: 10.90 m/s^2 X: 0.12 Y: 0.39 Z: 10.94 m/s^2 X: -0.35 Y: 1.92 Z: 5.73 m/s^2 X: -1.29 Y: 7.45 Z: 20.04 m/s^2 X: 0.16 Y: 0.31 Z: 10.59 m/s^2 X: 0.82 Y: -1.92 Z: 11.57 m/s^2 X: 0.71 Y: -2.51 Z: 9.92 m/s^2 X: 0.16 Y: -5.22 Z: 11.65 m/s^2 X: 1.84 Y: 2.43 Z: 11.02 m/s^2 X: -0.20 Y: 4.24 Z: 9.45 m/s^2 X: 4.16 Y: -7.49 Z: 12.04 m/s^2 X: 1.77 Y: -2.24 Z: 11.65 m/s^2 X: 0.98 Y: 0.59 Z: 15.49 m/s^2 X: 0.08 Y: 0.47 Z: 10.94 m/s^2 X: 0.08 Y: 0.43 Z: 10.98 m/s^2</pre>
---	--

Nota. Obtenido de: Elaborado por el autor.

En la **Figura 28**, podemos observar las pruebas de campo del sensor ADXL34, la gráfica representa tres variables o ejes “X”, “Y” y “Z” en función de la fuerza gravitatoria y del tiempo. Es interesante observar que los tres ejes siguen gráficamente el mismo patrón con ligeras variaciones en los datos leídos por el sensor ADXL345.

Figura 28.
Prueba de campo del ADXL345

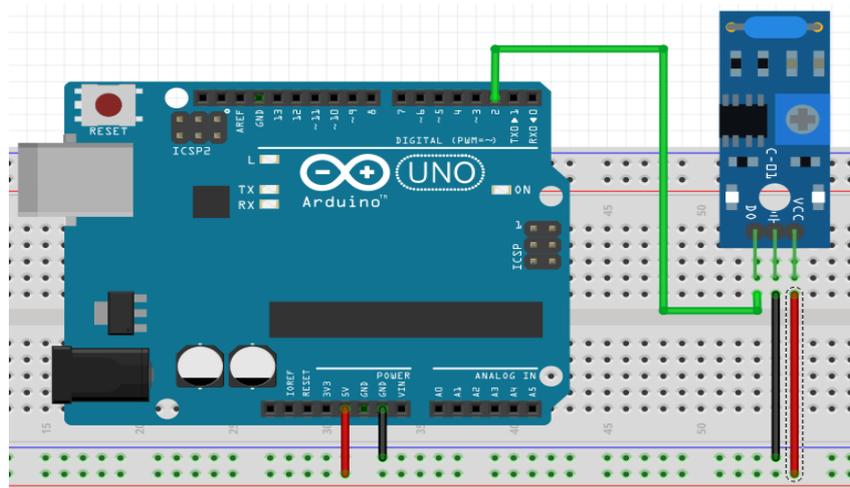


Nota. Obtenido de: Elaborado por el autor.

5.4.3 Sensor de vibración SW-420

En la **Figura 29**, se presenta el diagrama de prueba de funcionamiento del sensor SW-420. El pin de entrada digital va conectado al pin 2, GND Y VCC van conectados al Arduino Uno en los pines de alimentación correspondientes.

Figura 29.
Diagrama de prueba SW-420



Nota. Obtenido de: Elaborado por el autor.

La **Tabla 20**, se presenta el código fuente programado para el sensor SW-420. Una ventaja de utilizar este sensor es que no es necesario usar una librería debido a que el sensor brinda señales digitales (0,1). Además, se ha incluido los datos obtenidos del mismo, donde se colabora que el estado de funcionamiento del sensor es normal por lo que es adecuado integrarlo en el diseño del sistema de monitoreo estructural de puentes.

Tabla 20.
Código fuente SW-420

```
int vibration = 2;//GPIO1; // variable para conectar la salida del
sensor D0
int val; // almacenar la lectura (val es 1 cuando detecta vibración)
void setup () {Serial.begin(9600);
  pinMode (vibration, INPUT); // sensor como entrada (op)
} void loop () {
  val = digitalRead (vibration); // leer valor
  Serial.print(val);Serial.println();delay(500);}
```

Nota. Obtenido de: Elaborado por el autor.

En la **Figura 30**, podemos observar las pruebas de campo del sensor SW-420, la gráfica representa datos obtenidos del sensor con entradas de 1 o 0 en función del tiempo.

Dependiendo de la calibración o sensibilidad del sensor se detectarán vibraciones cuando se produzcan en la misma.

Figura 30.
Prueba de campo del SW-240

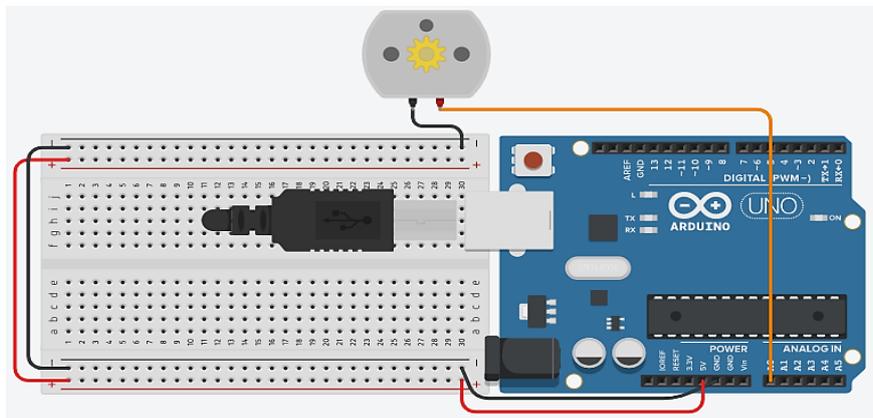


Nota. Obtenido de: Elaborado por el autor.

5.4.4 Sensor de viento

Internamente un anemómetro es integrado con un motor generador de energía, por cada revolución del motor genera un voltaje de salida. En base al voltaje de salida se puede calcular la velocidad del viento. La **Figura 31**, se presenta el diagrama de prueba del anemómetro, se utiliza la entrada analógica A0 para obtener el voltaje de salida.

Figura 31.
Diagrama de prueba del anemómetro



Nota. Obtenido de: Elaborado por el autor.

En la **Tabla 21**, se muestra el código de programación del anemómetro. Adicionalmente, utilizando el monitor serial se visualizan los datos obtenidos del componente.

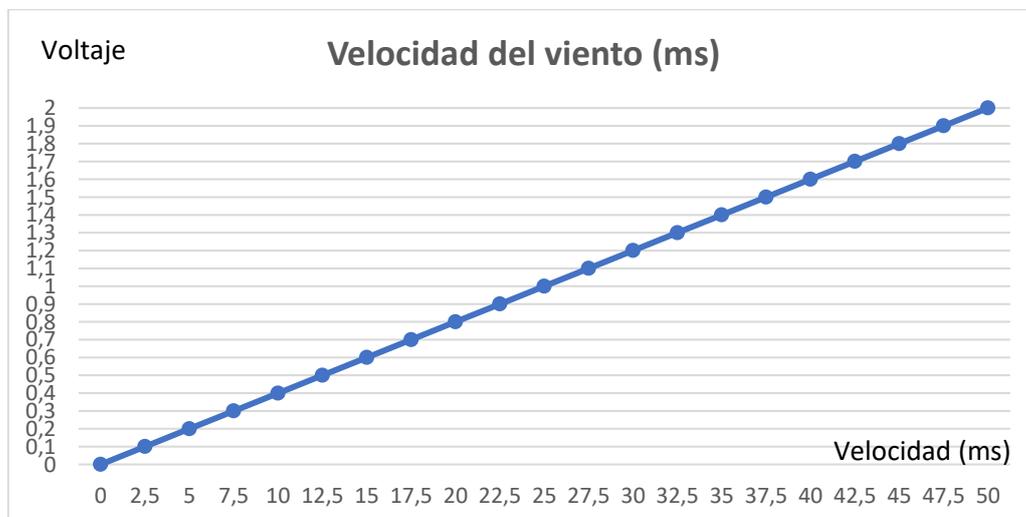
Tabla 21.
Código fuente del anemómetro

<pre>void setup() {Serial.begin(9600); } void loop() { float valorMedido = analogRead(A0); //convertimos la lectura analógica (que va de 0 a 1023) a voltaje (0- 5V): float voltaje = valorMedido* (5.0/1023.0); float velocidad= voltaje*25; float velocidad1=velocidad*(3600/1000); Serial.print("El valor medido en el pin A0 es= "); Serial.println(valorMedido); Serial.print(" que representa un valor de tension (V) de ");Serial.println(voltaje);Serial.print(" velocidad del viento (ms)=");Serial.println(velocidad); Serial.print(" velocidad del viento (km/h)= ");Serial.println(velocidad1); delay(2000);}</pre>	<pre>El valor medido en el pin A0 es= 16.00 que representa un valor de tension (V) de 0.08 velocidad del viento (ms)= 1.96 velocidad del viento (km/h)= 5.87 El valor medido en el pin A0 es= 11.00 que representa un valor de tension (V) de 0.05 velocidad del viento (ms)= 1.34 velocidad del viento (km/h)= 4.03 El valor medido en el pin A0 es= 7.00 que representa un valor de tension (V) de 0.03 velocidad del viento (ms)= 0.86 velocidad del viento (km/h)= 2.57 El valor medido en el pin A0 es= 5.00 que representa un valor de tension (V) de 0.02 velocidad del viento (ms)= 0.61 velocidad del viento (km/h)= 1.83</pre>
---	--

Nota. Obtenido de: Elaborado por el autor.

En la **Figura 32** es interesante observar que los datos obtenidos de un anemómetro generan una gráfica de forma de línea recta perpendicular. En el eje Y, representa la señal obtenida del anemómetro en Voltaje. El eje X representa la velocidad del viento en términos de metros sobre segundos (ms).

Figura 32.
Datos obtenidos del anemómetro



Nota. Obtenido de: Elaborado por el autor.

La **Figura 33**, representa las pruebas de campo del anemómetro en función de la velocidad del viento dados en Km/h sobre la señal de salida (voltaje) con picos máximos de 2 voltios.

Figura 33.
Pruebas de campo del anemómetro



Nota. Obtenido de: Elaborado por el autor.

5.5 Materiales

En este apartado se describen los materiales que se utilizaron en el tema planteado, se presentarán sus respectivas especificaciones técnicas y diagramas de bloques que conforman a los sensores usados para el desarrollo de la misma.

5.5.1 Gateway UG63

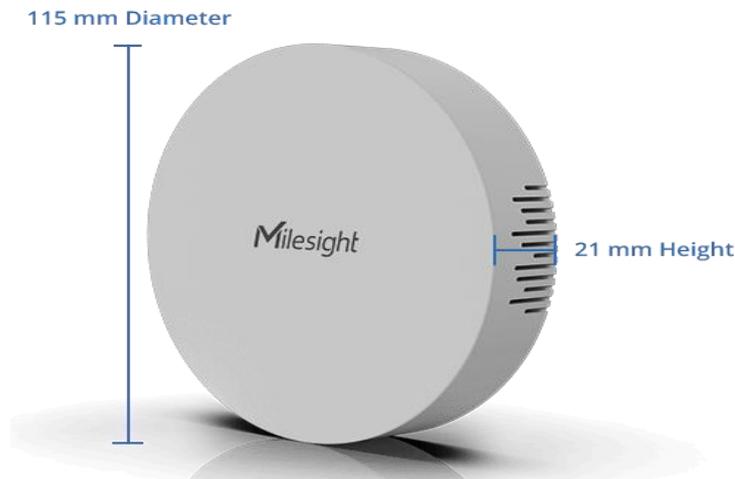
El Milesight UG63 es una pasarela LoRaWAN equipada con 8 canales y utiliza el chip SX1302 y una CPU de alto rendimiento. Esta combinación de hardware permite que el UG63 establezca una conexión de reenvío de paquetes entre más de 2000 nodos finales. Además de su capacidad de reenvío de paquetes, el UG63 también puede actuar como una pasarela de relleno ciego en entornos urbanizados, cubriendo distancias de hasta 2 km.

Una característica destacada del UG63 es que cuenta con un servidor de red integrado, lo que significa que puede gestionar los datos de unos pocos nodos finales de forma independiente. Esto hace que sea adecuado para pequeños despliegues LoRaWAN donde no se requiere una infraestructura de servidor de red adicional.

El UG63 es compatible con el último reenviador de paquetes de Semtech, lo que garantiza una comunicación eficiente y confiable en la red LoRaWAN. También es

compatible con múltiples plataformas de servidores de red, como TTN, ChirpStack, Senet y Orbiwise. En la **Figura 34**, se presentan las dimensiones del Gateway.

Figura 34.
Milesight UG63



Nota. La figura muestra las dimensiones físicas del UG63. Fuente: (Milesight, 2022)

En la **Tabla 22** se detallan las especificaciones técnicas del Gateway UG63, incluyendo parámetros técnicos como la frecuencia de operación, sensibilidad de recepción y la interfaz de comunicación, entre otros. También se proporciona información sobre el suministro de energía y el consumo de energía del Gateway.

Tabla 22.
Especificaciones técnicas del UG63

Hardware	CPU	528 MHz, ARM Cortex-A7
	System	Memory
	Flash	4 GB eMMC
LoRaWAN	Antenna	2 × Internal Antennas
	Channel	8 (Half/Full-duplex)
	Frequency Band	CN470/IN865/EU868/RU864/US915/AU915/KR920/AS923-1&2&3&4
	Sensitivity	-140dBm Sensitivity @292bps
	Output Power	27dBm Max
	Protocol	V1.0 Class A/Class B/Class C and V1.0.2 Class A/Class B/Class C
	LBT	Support
Ethernet Interface	Port	1 × RJ45 (PoE PD supported)
	Physical Layer	10/100 Base-T (IEEE 802.3)
	Data Rate	10/100 Mbps (Auto-Sensing)
	Interface	Auto MDI/MDIX

	Mode	Full or Half Duplex (Auto-Sensing)
Others	Reset Button	1 × RST
	Console Port	1 × Type-C
	LED Indicators	1 × SYS, 1 × LoRa
	Built-in	Watchdog, RTC, Timer
Software	Network Protocols	PPPoE, SNMP v1/v2c/v3, TCP, UDP, DHCP, DDNS, HTTP, HTTPS, DNS, ARP, SNTP, Telnet, SSH, MQTT, etc.
	VPN Tunnel	OpenVPN/IPsec/PPTP/L2TP/GRE/DMVPN
	Management	Web, CLI, DeviceHub, Milesight IoT Cloud, Yeastar Workplace Platform
Power Supply	Power Input	1. 1 × 802.3af standard PoE input 2. 5V by Type-C Port
Physical Characteristics	Ingress Protection	IP30
	Dimensions	Φ 115 x 21 mm (Φ 4.52 x 0.83 in)
	Color & Material	White, PC + ABS
	Installation	Desktop, Wall or Ceiling Mounting

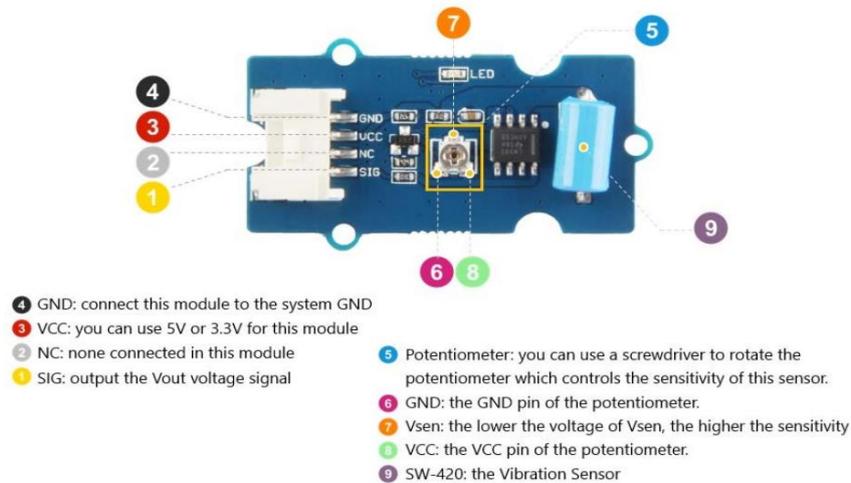
Nota. Datos obtenidos del datasheet. Fuente: (Milesight, 2022)

5.5.2 Sensor de vibración SW-420

El módulo de vibración utiliza el sensor de vibración SW-420 y el comparador LM393 para detectar señales de vibración que superen un umbral determinado. El umbral de vibración se puede ajustar mediante un potenciómetro integrado. Cuando no se detectan señales de vibración, el módulo activa un indicador LED que muestra un estado bajo o alto, dependiendo de la configuración correspondiente.

Este módulo se caracteriza por ofrecer una única señal de salida digital, la cual puede ser procesada por un microcontrolador. Este tipo de sensor se suele implementar en sistemas de seguridad, como alarmas antirrobo, vibraciones de un objeto físico, detección de terremotos y alarmas para automóviles, entre otros.

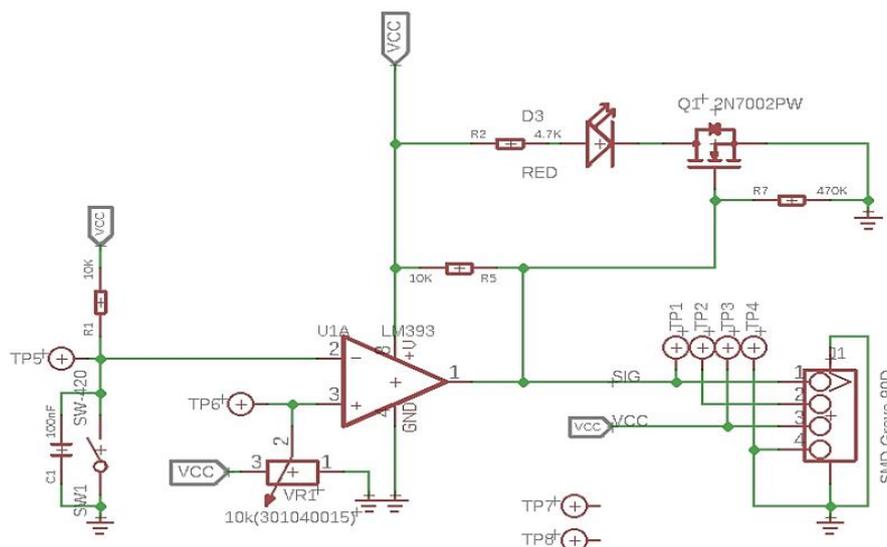
Figura 35.
Configuración de pines SW-420



Nota. La figura muestra los pines del sensor SW-420. Fuente: (Digikey, 2023)

En la **Figura 36**, se muestra el diagrama del SW-420. En primer lugar, el SW1 ubicado la esquina inferior izquierda es el módulo de vibración SW-420. Cuando el módulo está en un estado estable, el módulo se enciende. En segundo lugar, El pin 2 del U1A está conectado a GND a través del SW1. Finalmente, el VR1 es el potenciómetro, el Pin 2 del potenciómetro está conectado al Pin 3 de U1A. El U1A es un comparador, V+ se conecta al Pin 3, V- se conecta al Pin 2 y Vout se conecta al Pin 1.

Figura 36.
Diagrama del SW-420



Nota. La figura muestra el diagrama del SW-420. Fuente: (Digikey, 2023)

En la **Tabla 23**, se presentan las especificaciones técnicas del sensor SW-420. El sensor tiene un voltaje de operaciones en el rango de 3V a 5V, el consumo de corriente es de 15A e integra un indicar led cuando que se enciende cuando se detectan vibraciones.

Tabla 23.
Especificaciones técnicas del SW-420

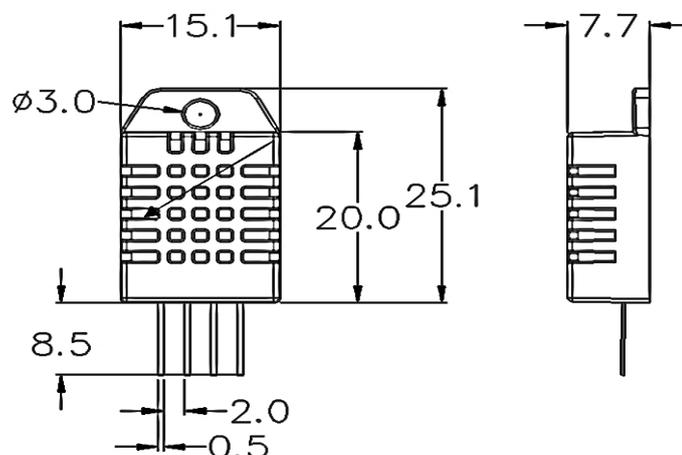
Parámetro	Descripción
Voltaje de operación	3.3V/5V
Corriente de operación	15 mA
Comparador	LM393
Salida digital	Nivel alto: 1 Nivel bajo: 0
LED señal indicadora	Integrado
Interface	Digital
Dimensiones	14 mm × 32 mm

Nota. Datos extraídos del datasheet SW-420. Fuente: (Digikey, 2023)

5.5.3 Módulo sensor DHT22

El DHT22 (AM2302) es un sensor digital de temperatura y humedad basado en el AM2302 con una interfaz digital de un solo cable. El sensor DHT22 puede ser usado en las plataformas Arduino/Raspberry Pi/Nodemcu. Su rango de medición de la temperatura es de 40 a +125 grados Celsius con una precisión de $\pm 0,5$ grados. Su rango de medición de la humedad es de 0 al 100% con una precisión del 2 al 5%. Utiliza una línea de datos para el intercambio y control de datos en el sistema. En la **Figura 37**, se presentan las dimensiones del módulo sensor DHT22.

Figura 37.
Módulo DHT22

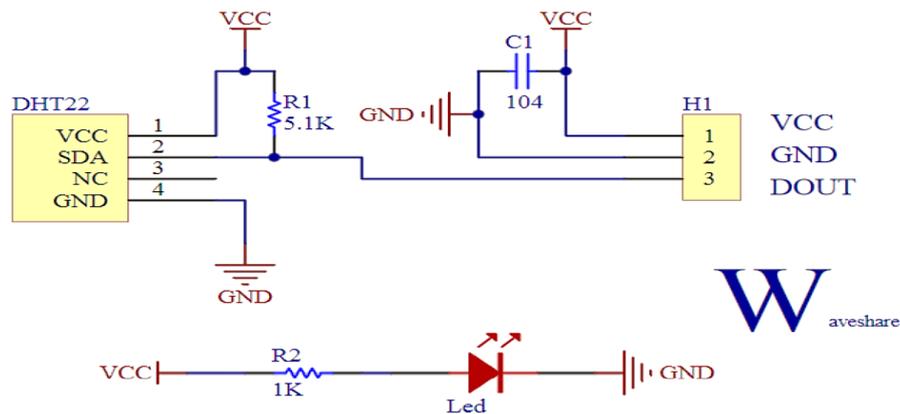


Nota. La figura muestra las dimensiones físicas del DHT22 (Aosong Electronics Co., n.d.).

La **Figura 38**, se muestra que el DHT22 está compuesto por 4 pines: VCC de alimentación a 3-5V, el pin GND a Tierra (0V), el pin de datos a un pin digital y Null (NC). El protocolo de comunicación entre el sensor y el microcontrolador emplea un único hilo, la distancia máxima recomendable de longitud de cable es de 20m.

Figura 38.

Diagrama sistemático del DHT22



Nota. La figura muestra el diagrama sistemático del DHT22. Fuente: (Rabesiranana et al., 2019).

En la **Tabla 24**, se presentan las especificaciones técnicas del sensor DHT22. Con voltaje de funcionamiento de 3.3V a 5.5.V y su rango de medición de temperatura es de -40°C a 125°C.

Tabla 24.

Especificaciones técnicas del sensor DHT22

Parámetro	Descripción
Voltaje de funcionamiento	3.3 V a 5.5 V
Rango de medición Temperatura	-40°C a 125°C
Precisión de medición de temperatura	<±0.5 °C
Resolución de Temperatura	0.1°C
Rango de medición Humedad	0% hasta 100% RH 0.5°C de variación
Precisión de medición de humedad	2% RH
Resolución Humedad	0.1% RH
Consumo de corriente	2.5 mA
Señal de salida	Digital
Distancia máxima de cable de datos	20m
Dimensiones	36 x 15 x 13 mm
Peso	4g

Nota. Datos obtenidos del datasheet DHT22. Fuente: (Aosong Electronics Co., n.d.)

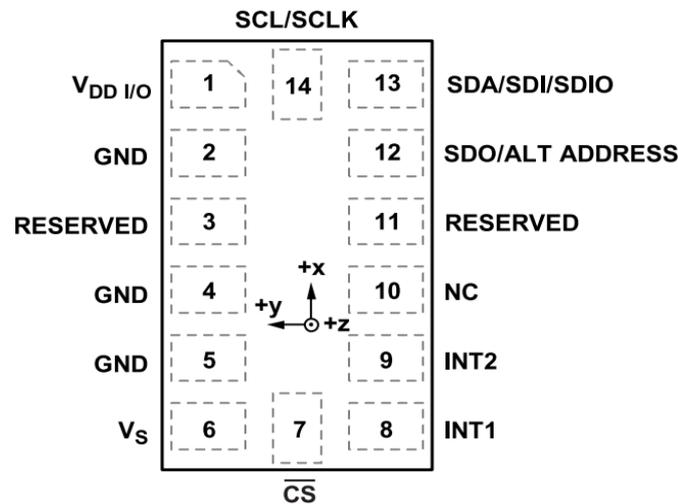
5.5.4 Acelerómetro ADXL345

El acelerómetro ADXL345 de 3 ejes se caracteriza por su bajo consumo de energía y por ser tan compacto. Los datos de salida digital están formateados como complemento de

dos de 16 bits y son accesibles a través de una interfaz digital SPI (3 o 4 hilos) o I2C. Incorpora una resolución de 4mg/LSB permitiendo detectar cambios de inclinación de hasta 1 grado. Además, detecta Tap Sensing de grifos simples y dobles y la detección de caída libre si el dispositivo se está cayendo.

Figura 39.

Configuración de pines del ADXL345

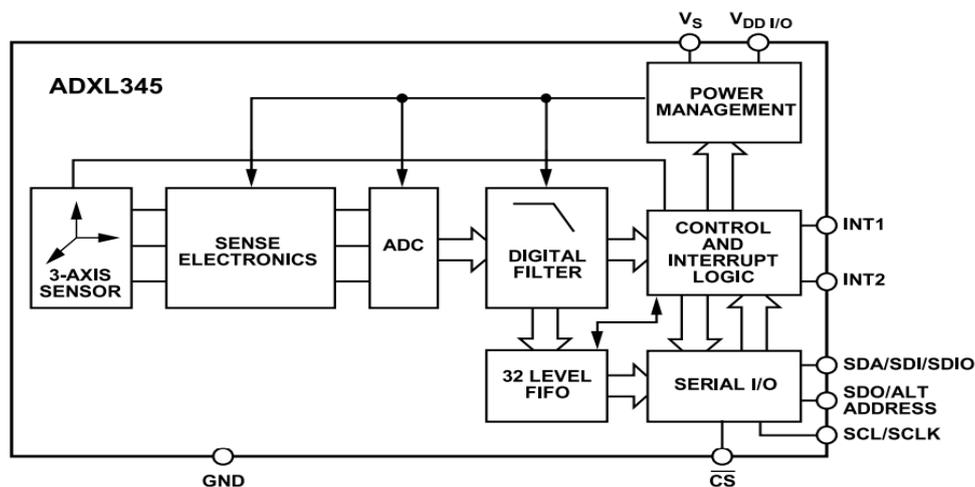


Nota. Figura obtenida del Datasheet ADXL345. Fuente: (Devices, n.d.).

En la **Figura 40**, se muestra el diagrama de bloques funcional del ADXL345 donde se dispone de un bloque de memoria FIFO que almacena hasta 32 conjuntos X, Y, Z. La comunicación puede realizarse a través de interfaces SPI y I2C. Adicionalmente, cuenta con un regulador de voltaje 6206A.

Figura 40.

Diagrama de bloques funcional del ADXL345



Nota. Figura obtenida del datasheet ADXL345. Fuente: (Devices, n.d.).

En la **Figura 39**, se ilustra la configuración de los pines del ADXL345, en total está compuesto por 14 pines que se describen a continuación en la **Tabla 25**.

Tabla 25.
Descripción de las funciones de los pines ADXL345

Pin No.	Termino	Descripción
1	V _{DD I/O}	Digital Interface Supply Voltage.
2	GND	Connected to ground
3	RESERVED	Reserved. This pin must be connected to VS or left open.
4	GND	Connected to ground
5	GND	Connected to ground
6	V _S	Supply Voltage
7	CS	Chip Select.
8	INT1	Interrupt 1 Output.
9	INT2	Interrupt 2 Output.
10	NC	Not Internally Connected
11	RESERVED	Reserved. This pin must be connected to ground or left open.
12	SDO/ALT ADDRESS	Serial Data Output (SPI 4-Wire)/Alternate I2C Address Select (I2C).
13	SDA/SDI/SDIO	Serial Data (I2C)/Serial Data Input (SPI 4-Wire)/Serial Data Input and Output (SPI 3-Wire).
14	SCL/SCLK	Serial Communications Clock. SCL is the clock for I2C, and SCLK is the clock for SPI

Nota. Datos extraídos del datasheet. Fuente: (Devices, n.d.)

En la **Tabla 26**, se presentan las especificaciones técnicas del sensor ADXL345. El sensor tiene un voltaje de operaciones en el rango de 3V a 5V, el consumo de corriente es de 23μA en funcionamiento y 0.1 μA en reposo. Su interfaz puede ser SPI o I2C.

Tabla 26.
Especificaciones técnicas del ADXL345

Parámetro	Descripción
Voltaje de operación	3V - 5V
Consumo de corriente	23 μA en medición y 0.1 μA en standby
Rango Acelerómetro	+/- 2g, 4g, 8g y 16g
Resolución	hasta 13 bits a ± 16 g (manteniendo 4 mg / LSB)
Grados de libertad	3 ejes
Regulador de voltaje	Integrado
Interfaz	SPI (3 o 4 hilos) a 5 MHz I2C (2 hilos) a 100kHz-400kHz
Dimensiones	3 mm × 5 mm × 1 mm
I/O rango de voltaje	1.7 V a 3.6V
Rango de temperatura	-40°C a +85°C

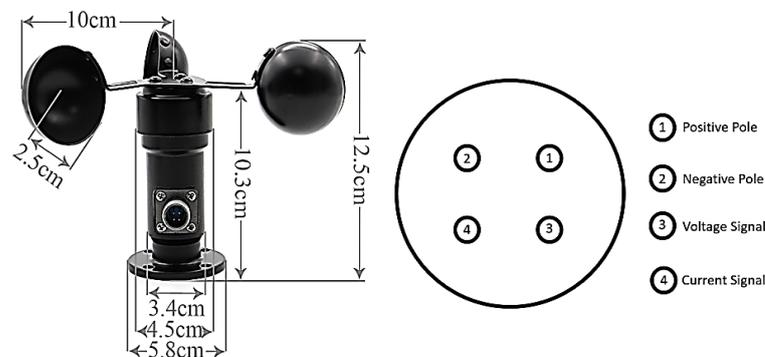
Nota. Datos extraídos del datasheet. Fuente: (Devices, n.d.)

5.5.5 Sensor JL -FS2

El anemómetro es un instrumento que puede medir la velocidad del viento. Se compone de la carcasa, la copa de viento y el módulo de circuito. Los módulos fotovoltaicos, el procesador de microcomputadora industrial, el generador de corriente, la corriente eléctrica, etc. están integrados internamente. Puede funcionar normalmente en $-20\text{ }^{\circ}\text{C} \sim +50\text{ }^{\circ}\text{C}$, humedad 35% ~ 85% (condensación).

En la **Figura 41**, se presenta las dimensiones y el diagrama del sensor compuesto por 4 pines: Positive Pole de color Rojo en rangos de +9-24V, Negative Polo de color Negro (GND), la Señal de voltaje de color amarilla y la señal de corriente de color azul.

Figura 41.
Sensor JL-FS2



Nota. La figura muestra las dimensiones físicas del JL-FS2 (Katranji, n.d.).

En la **Tabla 27**, se presentan las especificaciones técnicas del sensor de viento JL-FS2. Tiene una tensión de alimentación de 12 a 24 v DC, su temperatura de trabajo es de $-40\text{ }^{\circ}\text{C} \sim 80\text{ }^{\circ}\text{C}$.

Tabla 27.
Especificaciones técnicas JL-FS2

Parámetro	Descripción
Señal de salida	0-5
Resolución	0.1 m/s
Rango de medición	0-30 m/s
Medio de transmisión	transmisión por cable
Modo de conexión	sistema de tres hilos
Temperatura de trabajo	$-40\text{ }^{\circ}\text{C} \sim 80\text{ }^{\circ}\text{C}$
Tensión de alimentación	DC 12-24 V
Material	aleación de aluminio
Tamaño	18x10 cm/7.09x3.94 pulgadas
Longitud del cable	120 cm/47,2 pulgadas
Sensor	Tres tazas

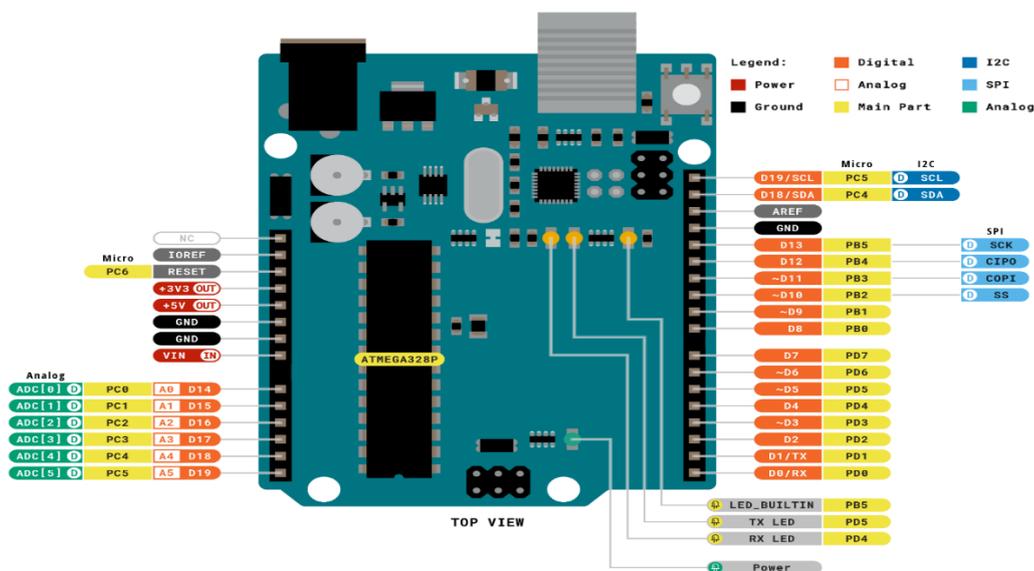
Nota. Datos extraídos del datasheet JL-FS2. Fuente: (Katranji, n.d.)

5.5.6 Arduino Uno R3

Arduino UNO es una placa de microcontrolador basada en el chip ATmega328P. Posee 14 pines digitales de entrada/salida donde 6 pines se pueden utilizar como salidas de modulación por ancho de pulso (PWM, del inglés *Pulse Width Modulation*), 6 entradas analógicas, un resonador cerámico de 16 MHz, una conexión USB, un conector de alimentación, un cabezal ICSP y un botón de reinicio.

En la **Figura 42** es interesante observar el mapa de pines del Arduino Uno integrado con el chip ATmega328P. En general, Arduino Uno tiene puertos de entrada/salida digitales y analógicos e integra pines de comunicación USB, I2C Y SPI.

Figura 42.
Pin mapping Arduino UNO



Nota. La figura exhibe el mapa de pines del Arduino Uno. Fuente:(Arduino, 2022)

En la **Tabla 28**, se presentan las especificaciones técnicas del Arduino Uno. Adicionalmente, se debe considerar otro aspecto muy importante en el apartado de la memoria. Para ello Arduino tiene tres tipos de memoria:

- **SRAM:** donde Arduino crea y manipula las variables cuando se ejecuta.
- **EEPROM:** memoria no volátil para mantener datos después de un reset o apagado.
- **Flash:** Memoria de programa. Usualmente desde 1 Kb a 4 Mb (controladores de familias grandes). Donde se guarda el sketch.

Tabla 28.
Especificaciones técnicas del Arduino Uno

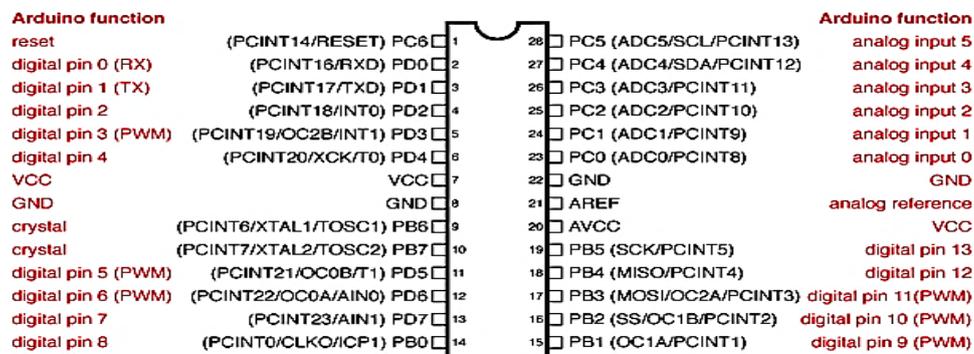
Parámetro	Descripción
Microcontroller & USB-to-serial converter	ATmega328P & Atmega16U2
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Nota. Datos extraídos del datasheet Arduino Uno. Fuente: (Arduino, 2022).

Arduino Uno integra el chip ATmega328P, se trata de un microcontrolador de alto rendimiento de 8 bits basado en la tecnología AVR RISC, que adopta una memoria flash ISP de 32 KB, EEPROM de 1 KB y SRAM de 2 KB.

En la **Figura 43** se presentan el mapa de pines del chip ATmega328P, tiene 23 líneas de E/S de propósito general, 32 registros de trabajo de propósito general, tres temporizadores/contadores flexibles con modos de comparación, interrupciones internas y externas, USART programable, una interfaz I2C, puerto en serie SPI, un convertidor A/D de 6 canales y 10 bits, watchdog programable con oscilador interno y cinco modos de ahorro de energía seleccionables por software.

Figura 43.
Chip ATmega328P



Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Nota. La figura muestra el mapa de pines del ATmega328P. Fuente: (Proyecto Arduino, 2020).

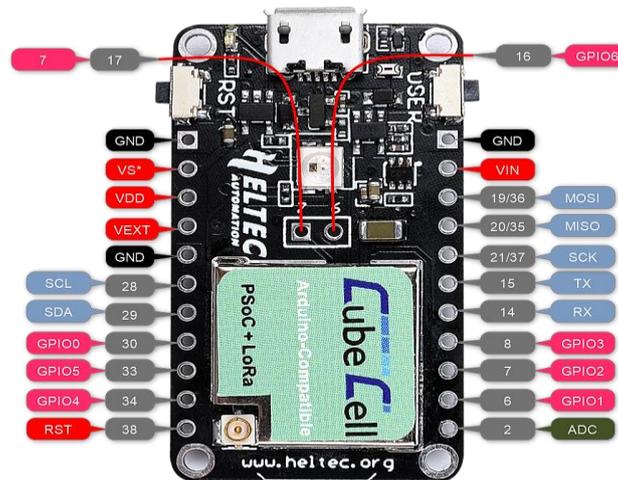
5.5.7 Módulo LoRa CubeCell HTCC-AB01

El módulo de comunicación inalámbrica se ha utilizado de la marca Heltec modelo CubeCell HTCC-AB01 basado en el chip ASR605x. El módulo Cubecell soporta LoRaWAN 1.0.2 diseñado con una potencia ultra baja con 3.5uA en reposo profundo.

Del mismo modo, en la

Figura 44 se presenta el mapa de pines del módulo de comunicación. En total se integra de 22 pines, 8 pines GPIO y una interfaz micro USB para la comunicación de datos y fuente de alimentación. Para mayor información acerca de la instalación del módulo Cubecell en Arduino Ide ver **Anexo 1**.

Figura 44.
Pin Mapping Heltec's Cubecell HTCC-AB01



Nota. La figura muestra el mapa de pines del cubecell. Fuente: (Heltec, 2022)

A continuación, en la **Tabla 29** se presentan las especificaciones técnicas del módulo LoRa. En ella se destacan sus interfaces, frecuencia de trabajo, consumo de energía, potencia de radiación y la compatibilidad del mismo con los entornos de desarrollo como Arduino IDE.

Tabla 29.
Especificaciones técnicas CubeCell HTCC-AB01

Parámetro		Descripción		
MCU		ASR6501		
LoRa chip		SX1262		
LoRa bands		470~510 MHz, 863~923 MHz		
LoRa maximum output power		22dB ± 1dB		
Frequency		470~510 MHz, 863~923 M		
Hardware resources		UART x 1; SPI x 1; I2C x 1; SWD x 1; 8-channel DMA engine GPIO x 8		
Display		None		
FLASH		128KB internal FLASH, 16KB internal SRAM		
Interface		Micro USB x 1; LoRa antenna interface (IPEX) x 1		
Size		50.2 x 25.5 x 9.74 mm		
USB to UART Bridge		CP2102		
Battery Type		3.7V Lithium (SH1.25 x 2 socket)		
Battery detection circuit		√		
External device power control (Vext)		√		
Low power design		Deep Sleep 3.5uA		
Electrical characteristics	Condition	Minimum	Typical	Maximum
Power Supply	USB powered (≥500mA)	4.7V	5V	6V
	Lithium powered (≥250mA)	3.3V	3.7V	4.2V
	3.3V (pin) powered (≥150mA)	2.7V	3.3V	3.5V
	5V (pin) powered (≥500mA)	4.7V	5V	6V
Power consumption(mA)	LoRa Rx output		10 mA	
	LoRa 10dB output		70 mA	
	LoRa 14 dB output		90 mA	
	LoRa 17 dB output		100 mA	
	LoRa 20 dB output		105 mA	
	Sleep Mode (USB powered)		9.6 mA	
	Sleep Mode (VBAT/battery powered)		11 μA	
	Sleep Mode (3.3V header powered)		2.5 μA	
Output	3.3V pin output			500mA
	5V pin output (USB powered only)		Equal to the input current	
	External device power control (Vext 3.3V)			350mA

Nota. Datos extraídos del datasheet CubeCell HTCC-AB01. Fuente:(Heltec, 2022).

6. Resultados

En el presente apartado se detallarán los resultados obtenidos en la simulación y las pruebas de funcionamiento de la WSN. De la misma forma, se evidenciarán las configuraciones de los dispositivos, diseño de los circuitos eléctricos y la integración con Ubidots.

6.1 Registro de dispositivos LoRa

En el proceso de registro de dispositivos LoRa en el servidor de red TTN, se siguen dos pasos principales. Primero, se registra el Gateway a través de la consola, y luego se procede a registrar el nodo final.

6.1.1 Registro del Gateway TTN

Para el registro del Gateway, se accede a la consola de TTN a través de tu página web oficial. En ella, se selecciona la unidad de “Go to gateways” para el registro del Gateway. Una vez dentro, hacer click en “register gateway” y llenar el formulario correspondiente.

En la **Figura 45** destaca el proceso de ingreso de variables esenciales, tales como el nombre, la frecuencia de trabajo, el identificador único global (EUI) y el ID del Gateway. Una vez que estos campos obligatorios se han completado y los datos se han verificado como correctos, es posible llevar a cabo con éxito el registro del Gateway.

Figura 45.

Formulario de registro de Gateway TTN

The image shows a screenshot of the TTN 'Register gateway' form. The form includes the following fields and options:

- Gateway EUI**: A text input field containing 'No gateway EUI' and a 'Reset' button. An orange arrow points from the label 'EUI de Gateway' to the 'Reset' button.
- Gateway ID**: A text input field containing 'gw1'. An orange arrow points from the label 'Identificador del Gateway' to this field.
- Gateway name**: A text input field containing 'Monitoreo estado de puentes'. An orange arrow points from the label 'Asignación de nombre del Gateway' to this field.
- Frequency plan**: A dropdown menu showing 'United States 902-928 MHz, FSB 2 (used by TTN)'. An orange arrow points from the label 'Plan de frecuencias' to this dropdown.
- Require authenticated connection**: A checkbox that is currently unchecked. Below it is a note: 'Choose this option eg. if your gateway is powered by LoRa Basic Station'.
- Share gateway information**: A section with two checked checkboxes: 'Share status within network' and 'Share location within network'.
- Register gateway**: A blue button at the bottom of the form.

Four orange callout boxes with arrows point to the 'Reset' button, 'Gateway ID' field, 'Gateway name' field, and 'Frequency plan' dropdown, each containing a label in Spanish: 'EUI de Gateway', 'Identificador del Gateway', 'Asignación de nombre del Gateway', and 'Plan de frecuencias'.

Nota. Obtenido de: Elaborado por el autor.

6.1.2 Registro del dispositivo TTN

Después de haber realizado exitosamente el registro del Gateway, el siguiente paso consiste en registrar el dispositivo final, conocido también como "end devices". Sin embargo, antes de proceder con dicho registro, es necesario crear la aplicación correspondiente. Esto se logra accediendo a la sección "Go to applications" en la consola de TTN.

La creación de la aplicación como se muestra en la **Figura 46**, involucra ingresar el ID de la aplicación, el nombre de la misma y proporcionar una breve descripción. Una vez que se hayan completado estos requisitos, se procede a crear la aplicación en TTN.

Figura 46.

Formulario de registro de aplicación TTN

monitoreo de estado estructural de puentes

Overview

End devices

Live data

Payload formatters

Integrations

MQTT

Webhooks

Storage Integration

AWS IoT

Azure IoT

LoRa Cloud

Collaborators

Hide sidebar

Applications > monitoreo de estado estructural de puentes

General settings

Application ID *

monitoreo-puente

Identificador único

Name

monitoreo de estado estructural de puentes

Nombre de la aplicación

Description

El sistema consiste en monitorear el estado estructural de puente utilizando sensores como anemómetro, termómetro y acelerómetro

Descripción de la aplicación

Skip payload encryption and decryption ⓘ

Use Application Layer Clock Synchronization ⓘ

Attributes

+ Add attributes

Attributes can be used to set arbitrary information about the entity, to be used by scripts, or simply for your own organization

Save changes

Delete application

Nota. Obtenido de: Elaborado por el autor.

Una vez creada la aplicación, en la sección de "End devices" se registran los dispositivos finales. El registro de los nodos implica ingresar datos importantes, como la frecuencia de trabajo, la versión de LoRaWAN, los parámetros regionales, el ID del nodo, el DevEUI y las claves de cifrado. Al completar todos estos requisitos, se registran exitosamente los nodos en el sistema (ver **Figura 47**).

Figura 47.
Registro del nodo TTN

Register end device

✓ QR code scanned successfully
[Reset QR code data](#) [Learn more](#)

End device type

Input Method

Select the end device in the LoRaWAN Device Repository

Enter end device specifics manually

Frequency plan

LoRaWAN version

Regional Parameters version

[Show advanced activation, LoRaWAN class and cluster settings](#)

Provisioning information

JoinEUI [Reset](#)

This end device can be registered on the network

DevEUI [Generate](#) 0/50 used

AppKey [Generate](#)

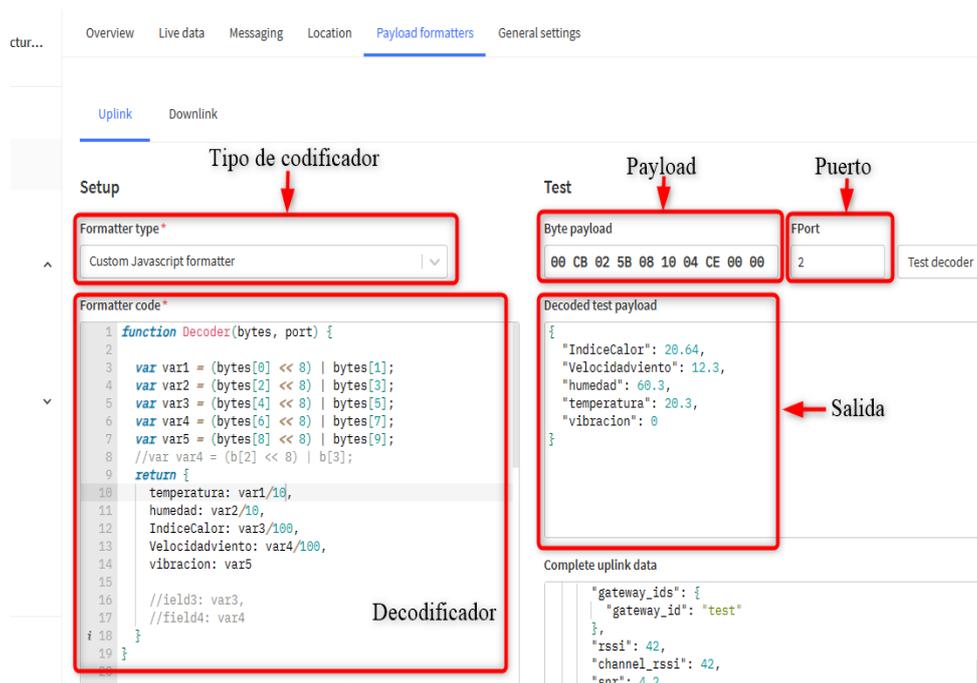
Nota. Obtenido de: Elaborado por el autor.

6.2 Decodificador payload TTN

TTN posee un decodificador de *payload* que permite procesar los datos que se envían hacia y desde dispositivos finales. El decodificador es útil para convertir el *payload* en un objeto JSON legibles por humanos u otro tipo de conversión de datos en enlaces ascendentes y descendentes.

En la versión 3 de Things Stack, se admiten tres tipos de formateadores de carga útil: JavaScript, CayenneLPP y GRPC service. En la **Figura 48**, se usa el decodificador personalizado de javascript para decodificar los mensajes que envía el dispositivo final donde el payload del enlace ascendente se transmite como carga útil binaria por el nodo. Para mayor detalle del código de programación del decodificador ver **Anexo 9** y **Anexo 10**.

Figura 48.
Decodificador payload TTN



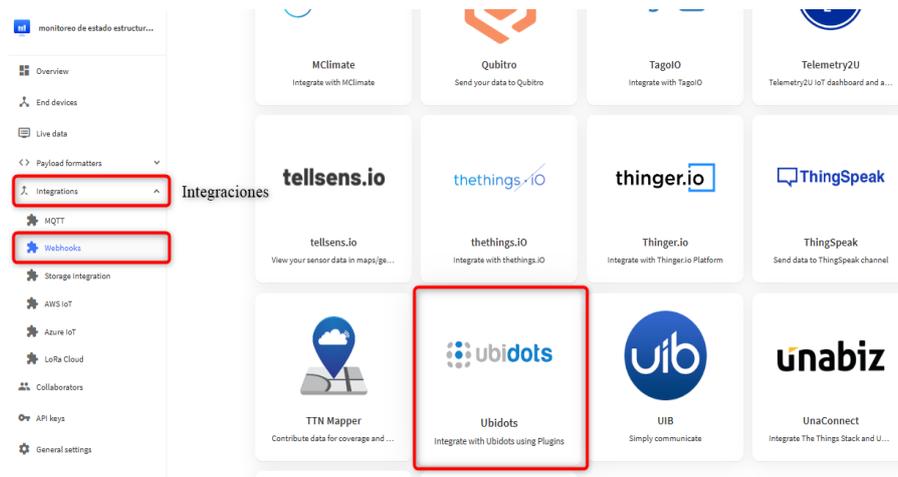
Nota. Obtenido de: Elaborado por el autor.

6.3 Integración TTN con Ubidots

Las Integraciones de TTN se refiere a las capacidades de integración por HTTP o MQTT e incluso por mediante una serie de APIs en diferentes lenguajes de programación, como Go, Java, Node-RED y Node.js. Con estas integraciones, es posible enviar datos desde los nodos y gateways hacia el servidor de TTN, y luego procesarlos y utilizarlos en la aplicación IoT.

A continuación, se detallará una de las partes importantes sobre la visualización de datos. Para ello, es necesario realizar una integración entre el servidor de red (TTN) y una API. En la **Figura 49**, se muestra la consola de TTN donde existe una sección de “Integrations” que permite una conexión con un servicio externo como Ubidots.

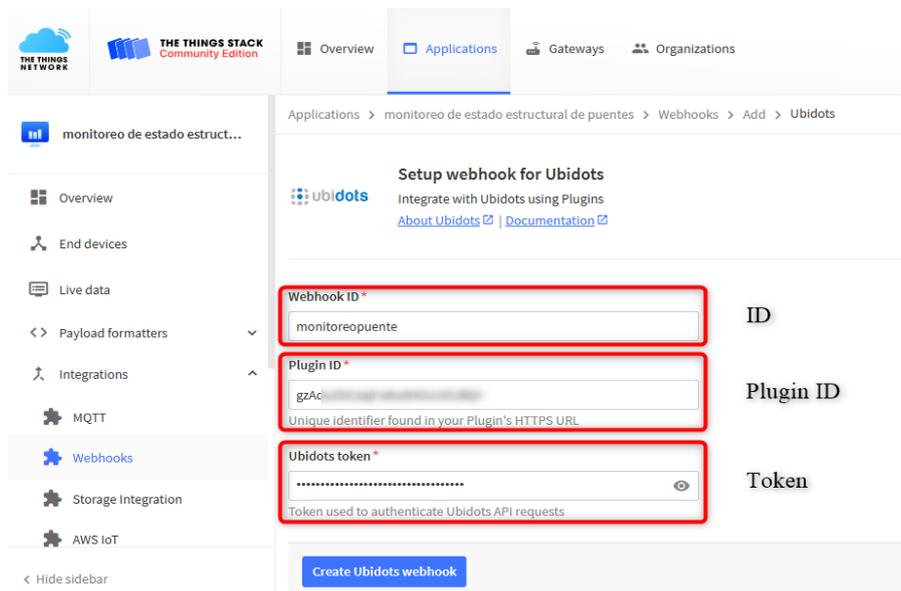
Figura 49.
Integraciones de TTN



Nota. Obtenido de: Elaborado por el autor.

Para realizar una integración con Ubidots es necesario ingresar las credenciales (*Plugin* y *token*) de acceso para que exista una comunicación entre el servidor de red TTN y Ubidots (ver **Figura 50**).

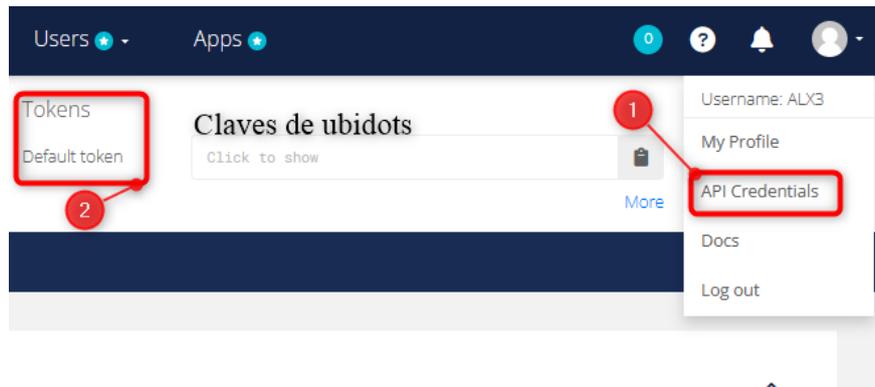
Figura 50.
Integración con Ubidots



Nota. Obtenido de: Elaborado por el autor.

Continuando con el proceso de integración, es necesario crear una cuenta en Ubidots para generar el *plugin* y el *token*. Los *plugins* de Ubidots permiten la integración con otros servicios y sistemas externos para obtener datos. Por otro lado, un *token* en Ubidots es una clave de acceso que se utiliza para autenticar y autorizar el acceso a la API de Ubidots. En la **Figura 51** se procede a crear el *token*.

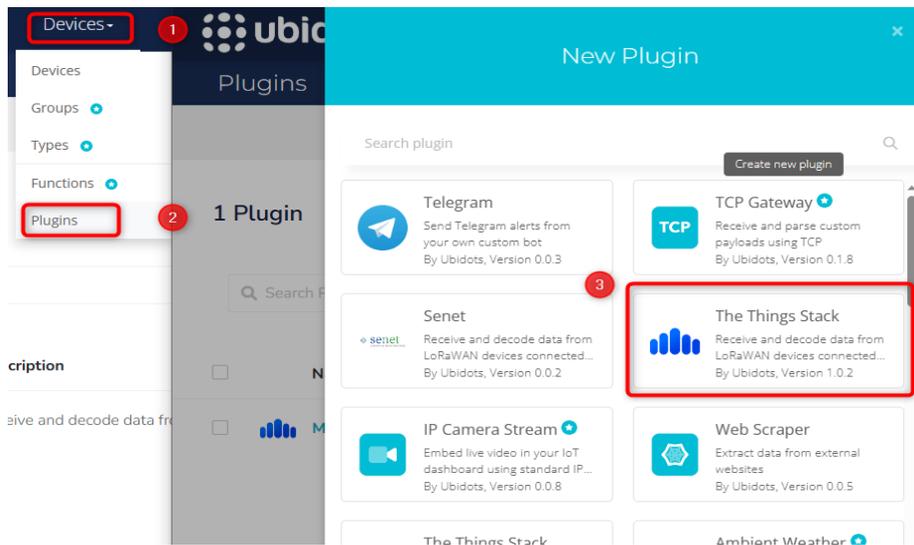
Figura 51.
Creación de token Ubidots



Nota. Obtenido de: Elaborado por el autor.

Seguidamente, se procede a crear el *plugin* como se visualiza en la **Figura 52**. Primeramente, se ingresa a la sección de “*Divices*” → “*Plugins*”. Luego en “*New Plugin*”, se integra con el servidor de red TTN. En la versión 3 de TTN pasó a llamarse “*The Things Stack*”.

Figura 52.
Creación del Plugin

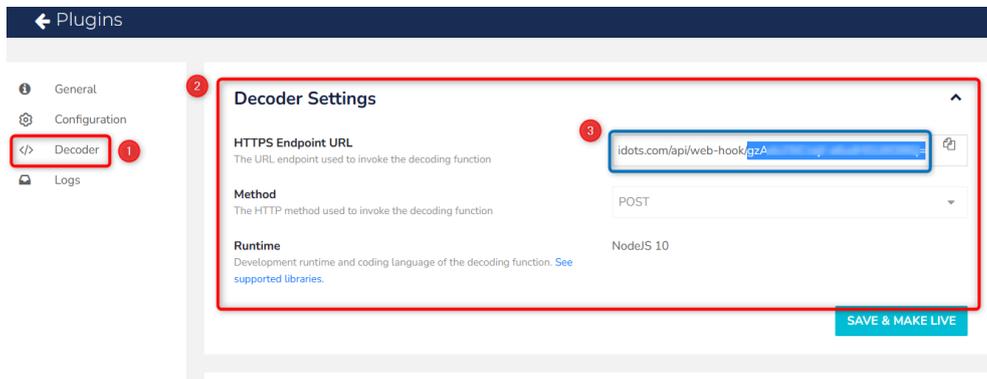


Nota. Obtenido de: Elaborado por el autor.

Concluido la creación del *plugin*, en la seccion de “*Decoder*” existen tres parametros importantes. Una de ellas es el “*HTTPS Endpoint URL*”, este se refiere a la dirección URL de un punto final (endpoint) seguro que utiliza el protocolo HTTPS para enviar datos a la plataforma Ubidots. Por su parte, el termino “*Method*”, se refiere al método HTTP utilizado para enviar la solicitud al servidor. En el caso de Ubidots, el método comúnmente utilizado es “*POST*” (ver **Figura 53**).

En conclusión, al utilizar el *HTTPS Endpoint URL*, se puede realizar una solicitud *POST* a esa URL con los datos que deseas enviar a Ubidots. Los datos deben estar en formato JSON para la integración.

Figura 53.
Decoder Settings

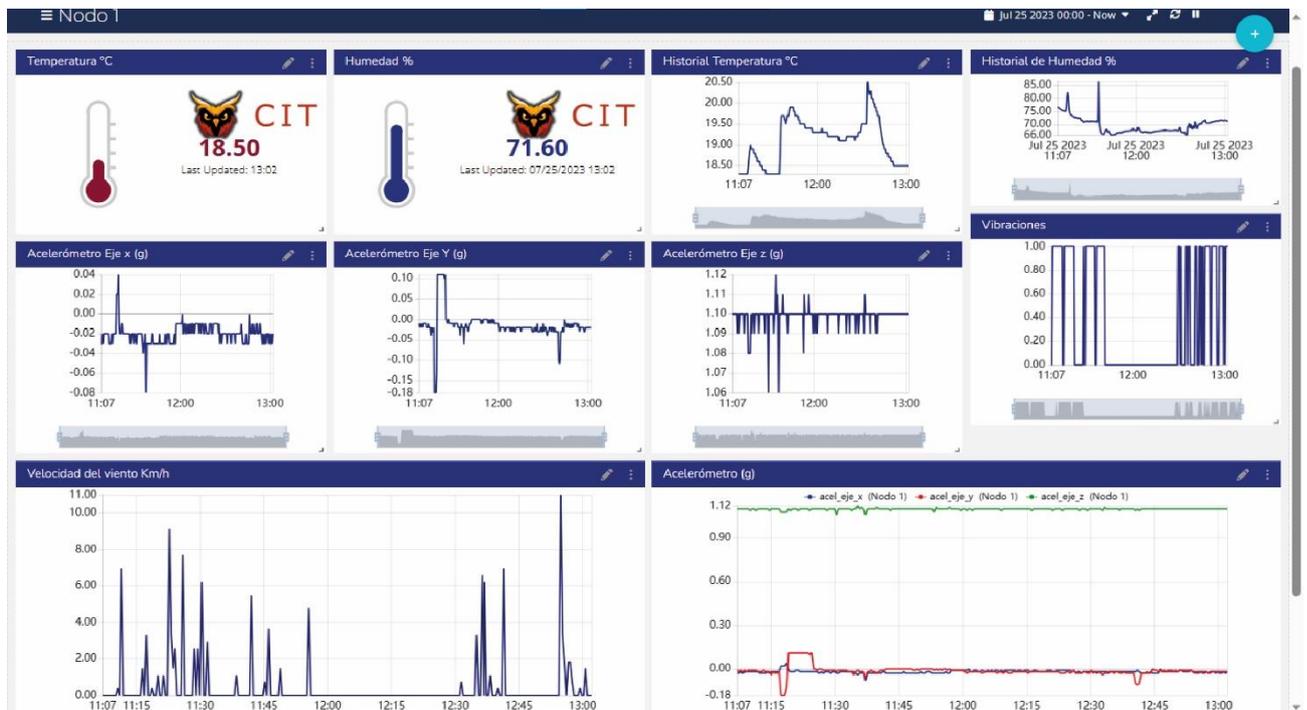


Nota. Obtenido de: Elaborado por el autor.

6.3.1 Dashboard Nodo 1

Concluido el proceso de integración de Ubidots, se procede a crear los *dashboards*. En la **Figura 54** se muestra el *dashboard* del nodo 1, consta de 7 *widets* que exhiben los datos obtenidos por los nodos finales.

Figura 54.
Dashboard del Nodo 1



Nota. Obtenido de: Elaborado por el autor.

6.3.2 Dashboard Nodo 2

Finalmente, en la **Figura 55** se visualiza el *dashboard* del nodo 2 integrado por 7 widgets que permitió mostrar los datos enviados por el nodo final. De esta forma, los nodos envían datos obtenidos de los sensores a TTN para ser visualizados en los *dashboards* de la plataforma Ubidots.

Figura 55.
Dashboard del Nodo 2

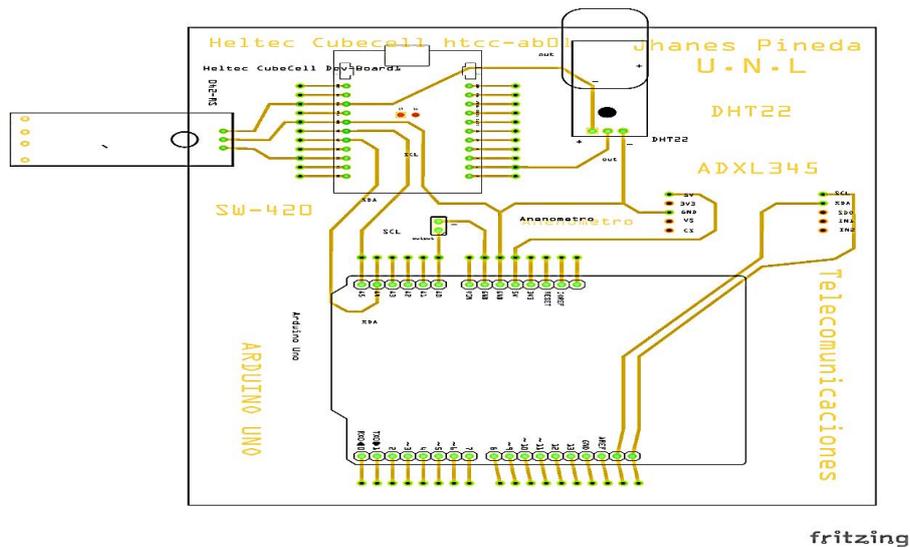


Nota. Obtenido de: Elaborado por el autor.

6.4 Diseño de los circuitos impresos

A continuación, en la **Figura 56** se presenta el diseño del circuito impreso del nodo sensor 1. El esquema del nodo 1 fue integrado por los sensores DHT22 y SW-420 conectados al módulo CubeCell. Por su parte, el sensor ADXL345 y el sensor de viento estaban conectados al Arduino Uno haciendo una comunicación I2C. Para mayor detalle ver **Anexo 17**.

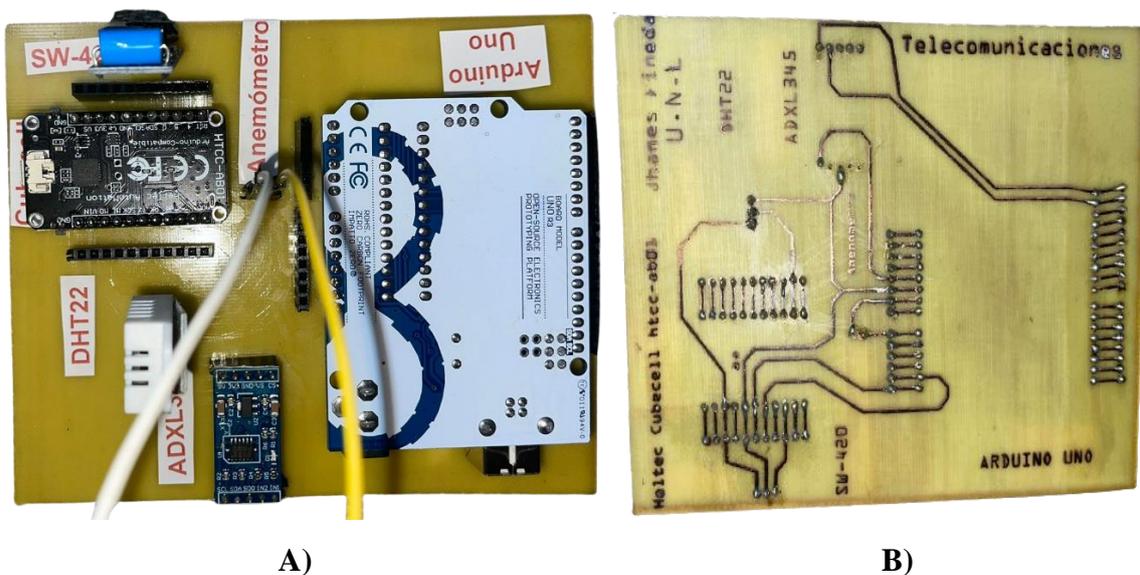
Figura 56.
Diseño del circuito impreso Nodo 1



Nota. Obtenido de: Elaborado por el autor.

En la **Figura 57**, se muestra el resultado del diseño del circuito impreso (PCB) del nodo 1 utilizando el software Fritzing. La placa PCB fue compuesta por los sensores DHT22 y SW-420, así como el módulo de comunicación LoRa CubeCell. Además, se integra el Arduino Uno, el cual envía datos de los sensores ADXL345 y el anemómetro al módulo CubeCell mediante una comunicación I2C. Esto permite la transferencia de datos desde los sensores conectados al Arduino hacia el módulo CubeCell para su posterior transmisión a través de la tecnología LoRa.

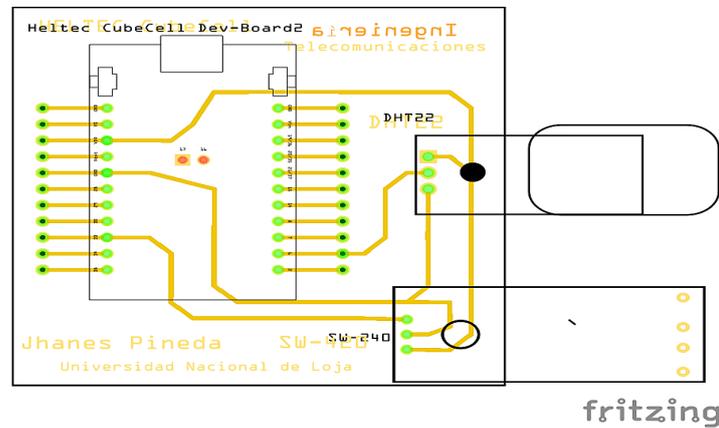
Figura 57.
Placa PCB del nodo 1



Nota. A) Placa PCB nodo 1 vista superior. B) Placa PCB nodo 1 vista inferior.

De la misma manera, la **Figura 58** representa el diseño del circuito impreso del nodo sensor 2. El nodo 2 integró los sensores DHT22 y SW-420 conectados al módulo de comunicación CubeCell HTCC-AB01. Para mayor detalle ver **Anexo 17**.

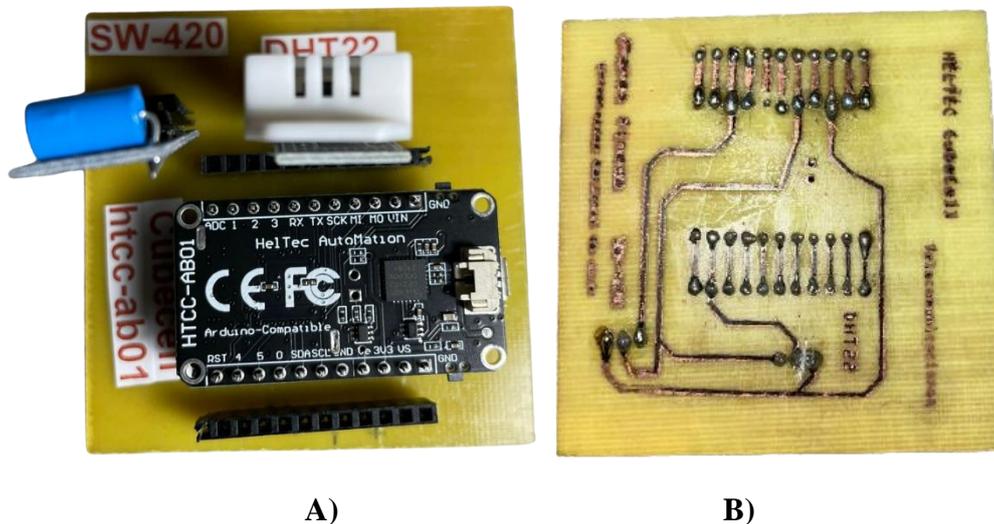
Figura 58.
Diseño del circuito impreso Nodo 2



Nota. Obtenido de: Elaborado por el autor.

En la **Figura 59**, se puede apreciar el resultado del circuito impreso del nodo 2 elaborado previamente. La placa PCB se encuentra compuesta los sensores DHT22 y SW-420, de la misma forma íntegra el módulo de comunicación LoRa CubeCell.

Figura 59.
Placa PCB del nodo 2



Nota. A) Placa PCB nodo 2 vista superior. B) Placa PCB nodo 2 vista inferior. Obtenido de: Elaborado por el autor.

6.5 Resultados de las pruebas de funcionamiento del sistema WSN

El funcionamiento del sistema WSN se realizó en base a pruebas del prototipo instalado sobre el escenario analizado previamente en la **Figura 16**. En primer lugar, las

pruebas incluyeron la verificación del correcto envío de datos desde los nodos finales hacia el Gateway y posteriormente al servidor de red TTN. Además, se confirmó la adecuada visualización de los datos en Ubidots. Estas pruebas aseguraron el correcto desempeño de la red WSN y su capacidad para recopilar, transmitir y visualizar datos de manera efectiva.

Por otra parte, en la **Figura 60** se presenta una representación visual de los nodos finales implementados dentro de una caja protectora diseñada para enfrentar las condiciones climáticas adversas del entorno. La inclusión de esta protección garantiza la integridad y el rendimiento óptimo de los módulos LoRa a lo largo del tiempo. Entre los elementos fundamentales que componen los nodos finales, se enumeran a continuación:

- | | |
|-------------------|-------------------------|
| 1. Módulo LoRa. | 5. Antena LoRa. |
| 2. Batería. | 6. Arduino Uno. |
| 3. Sensor DHT22. | 7. Acelerómetro ADXL345 |
| 4. Sensor SW-420. | 8. Anemómetro. |

Figura 60.

Nodos finales con sensores integrados en caja protectora



Nota. A) Primer nodo final. B) Segundo nodo final. Fuente: Elaborado por el autor.

En la **Figura 61** se visualiza la ubicación de los nodos finales en la estructura del puente seleccionado para el escenario analizado. Tanto el primer como el segundo nodo del prototipo se encuentran desplegados en sus ubicaciones correspondientes. También se destaca la colocación del anemómetro sobre la estructura para obtener datos relevantes sobre las condiciones ambientales.

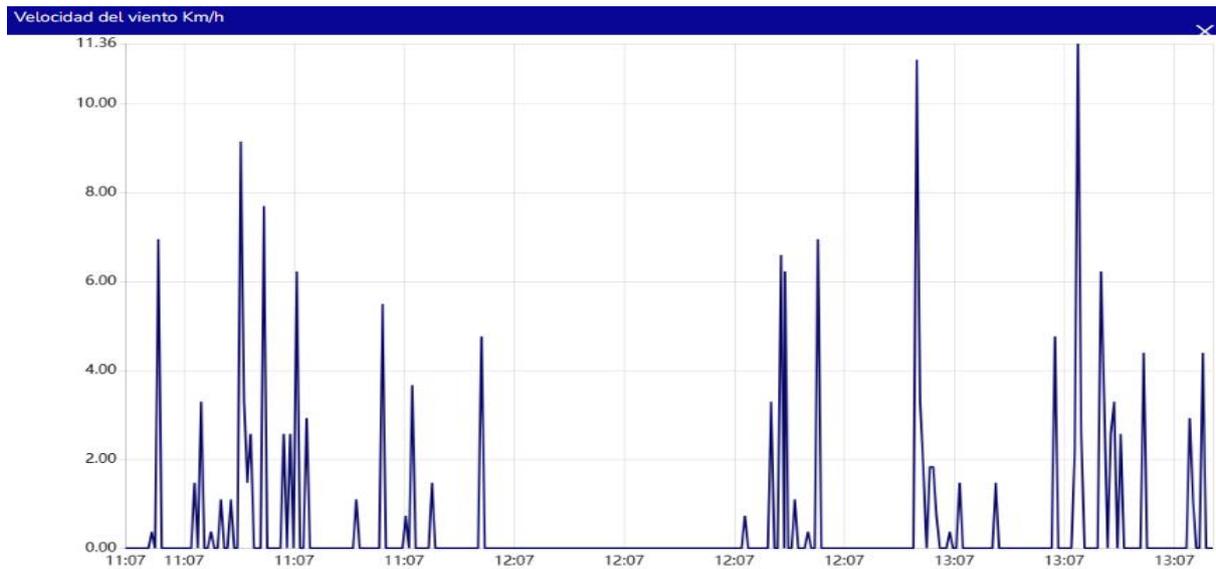
Figura 61.
Despliegue estratégico de los nodos finales



Nota. Obtenido de: Elaborado por el autor.

Durante la fase de pruebas del prototipo, se llevó a cabo una exhaustiva evaluación para garantizar su adecuado desempeño. Como resultado de estos ensayos, se pudo verificar la correcta operatividad del sistema WSN. En la **Figura 62** se presentan los datos del anemómetro del primer nodo final obtenidos durante las pruebas de funcionamiento. Los resultados muestran la velocidad del viento en kilómetros por hora (Km/h) registradas por el dispositivo.

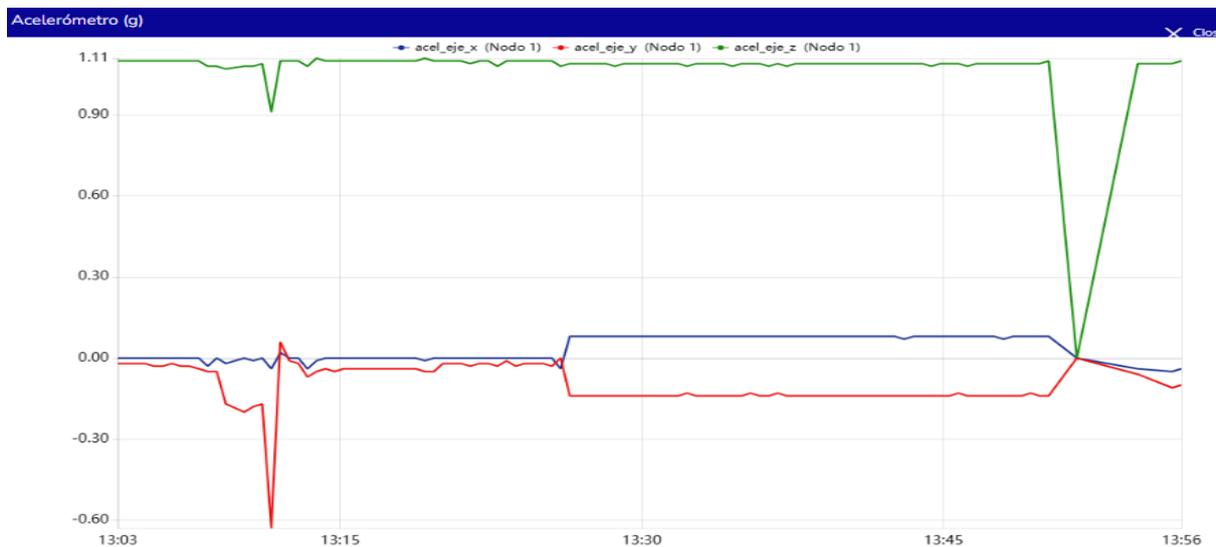
Figura 62.
Visualización de datos del anemómetro (Km/h)



Nota. Obtenido de: Elaborado por el autor.

Por otra parte, en la **Figura 63** se muestran los datos obtenidos a través del acelerómetro realizadas en el prototipo. Estos datos muestran las más pequeñas vibraciones producidas sobre la estructura del puente. Las unidades del acelerómetro fueron configuradas en gravedades (g) en las tres dimensiones “X” “Y” y “Z”.

Figura 63.
Visualización de datos del acelerómetro (g)



Nota. Obtenido de: Elaborado por el autor.

En la **Figura 64** se ilustran los datos de humedad recopilados durante las pruebas. La unidad de medida utilizada para expresar la humedad relativa es el porcentaje, representado

por el símbolo %. Esta escala puede variar desde el 0% para identificar el aire completamente seco hasta el 100% que indica que el aire está completamente saturado de humedad.

Figura 64.

Visualización de datos de humedad (%)



Nota. Obtenido de: Elaborado por el autor.

La **Figura 65** muestra los resultados de las mediciones de temperatura realizadas en el segundo nodo. Aunque cabe recalcar que ambos nodos integran un sensor de temperatura y humedad. Las unidades de la temperatura fueron configuradas en grados centígrados (°C).

Figura 65.

Visualización de datos de temperatura (°C)



Nota. Obtenido de: Elaborado por el autor.

Los resultados de las mediciones de sensación térmica se presentan en la **Figura 66**. Esta información es valiosa para comprender cómo perciben las personas la temperatura en

función de otros factores como la humedad y la velocidad del viento. Las unidades de esta variable fueron configuradas en grados centígrados (°C).

Figura 66.

Visualización de datos de sensación térmica (°C)



Nota. Obtenido de: Elaborado por el autor.

Por último, en la **Figura 67** se presentan los datos obtenidos a través del sensor de vibración SW-420 durante las pruebas de funcionamiento del prototipo. Este sensor es altamente sensible y se activa al detectar vibraciones en su entorno. Cuando se produce una vibración, el sensor emite una señal digital de "1", mientras que, en ausencia de vibraciones, la señal es "0". La unidad de medida asociada con este sensor es el valor binario (1 o 0), lo que permite una detección clara y directa de la presencia o ausencia de vibraciones en el área monitoreada.

Figura 67.

Visualización de datos de detecciones de vibraciones



Nota. Obtenido de: Elaborado por el autor.

6.6 Análisis de resultados del sistema WSN

En esta sección, se analizaron los resultados obtenidos a partir de los datos recopilados por los dos nodos finales. Cada uno de los nodos finales estaban equipado con varios sensores, incluyendo el DHT22, que permitía medir la temperatura y la humedad ambiental; el acelerómetro ADXL345, utilizado para detectar vibraciones y cambios en la aceleración; el sensor de vibración SW-420, que proporcionaba información sobre eventos sísmicos y movimientos indeseados; y, por último, el anemómetro, que medía la velocidad del viento en el área de interés.

6.6.1 Detecciones de vibraciones

El sensor de vibración SW-420 se sometió a un análisis de detecciones y no detecciones durante 6 días de monitoreo estructural del puente. Durante este periodo, se registraron un total de 5056 detecciones de vibraciones y 5332 ocasiones en las que no se detectaron vibraciones significativas.

Los resultados del primer nodo se resumen en la **Tabla 30**. Durante el periodo de pruebas, se contabilizaron un total de 2528 detecciones de vibraciones, mientras que se constató la ausencia de vibraciones significativas en 2666 ocasiones.

Tabla 30.
Resultados de las detecciones de vibraciones del nodo 1

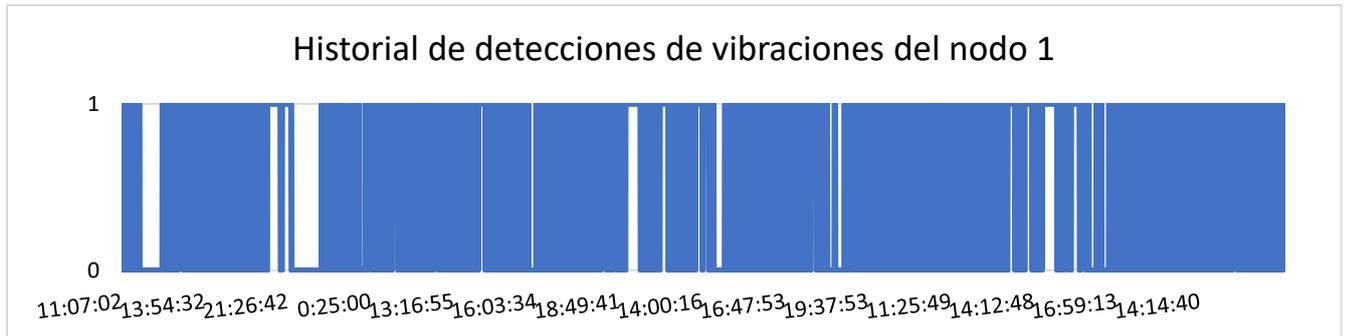
Nodo	Estadísticas	Valor
Nodo 1	Vibraciones detectadas	2528
	Vibraciones no detectadas	2666
	Total de eventos	5194
	Tiempo de ejecución	6 días

Nota. Obtenido de: Elaborado por el autor.

De la misma forma, la **Figura 68** se presenta el historial de las detecciones de vibraciones generadas por la API Ubidots. Estos datos resultaron fundamentales para el monitoreo estructural del puente, ya que permitieron identificar eventos que pudieran generar resonancias, vibraciones excesivas o cargas dinámicas inesperadas. Las detecciones con el sensor SW-420 proporcionaron información valiosa para evaluar la respuesta dinámica del puente ante cargas ambientales y tráfico vehicular, lo que ayudó a garantizar la integridad estructural y la seguridad del puente en todo momento.

Figura 68.

Historial de las detecciones de vibraciones del nodo 1



Nota. Obtenido de: Elaborado por el autor.

Por otra parte, en la **Tabla 31** se presentan los resultados del análisis de detecciones de vibraciones utilizando el sensor SW-420 en el segundo nodo. En este caso, se registraron 2156 de detecciones de vibraciones y 3007 ocasiones en las que no se presentaron anomalías en las vibraciones de la estructura.

Tabla 31.

Resultados de las detecciones de vibraciones del nodo 2

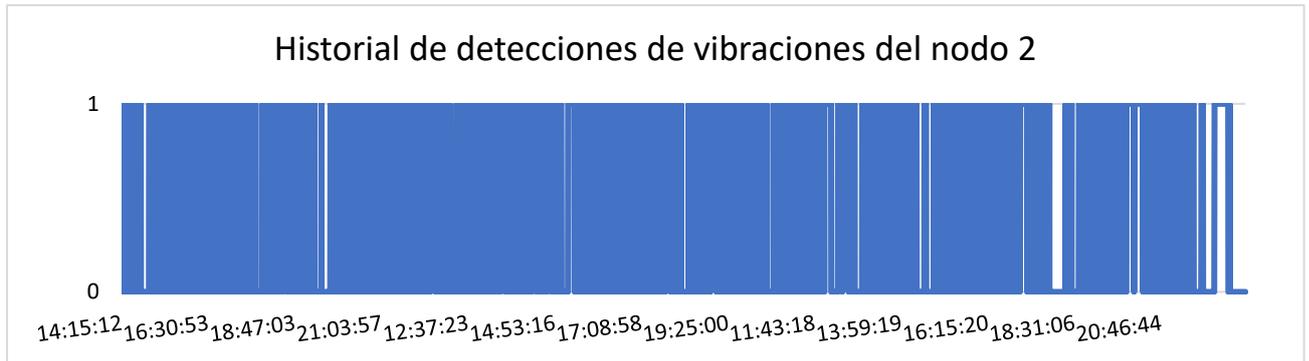
Nodo	Estadísticas	Valor
Nodo 2	Vibraciones detectadas	2156
	Vibraciones no detectadas	3007
	Total de eventos	5163
	Tiempo de ejecución	6 días

Nota. Obtenido de: Elaborado por el autor.

De la misma forma, en la **Figura 69** se muestran los resultados del SW-420 del segundo nodo. Las ocasiones en las que no se detectaron vibraciones también proporcionan información valiosa. Pueden indicar momentos de calma estructural, lo que sugiere que la estructura está funcionando dentro de rangos seguros y no está experimentando movimientos inesperados o comportamientos anómalos.

Figura 69.

Historial de las detecciones de vibraciones del nodo 2



Nota. Obtenido de: Elaborado por el autor.

6.6.2 Temperatura y humedad

Por otra parte, el sensor DHT22 ha sido desplegado para efectuar mediciones de temperatura y humedad ambiental. Los datos obtenidos han permitido calcular tanto el promedio, el valor máximo y el valor mínimo de las temperaturas registradas en el entorno de monitoreo.

De manera similar, se ha realizado el análisis para la humedad, revelando el promedio, el valor máximo y el valor mínimo de las mediciones. Adicionalmente, la sensación térmica, producto de la combinación de los datos de temperatura y humedad, también ha sido calculada para obtener una medida ajustada a la percepción real de la temperatura.

Para el caso del primer nodo, en la **Tabla 32** se presentan los resultados del análisis de la temperatura y humedad. Los resultados indican un promedio de temperatura de 19,85 °C alcanzando un valor máximo de 24,5 °C y un mínimo de 15,30 °C. Estos datos no resultan alarmantes ya que son temperaturas que no afectan al estado estructural del puente.

Tabla 32.

Resultados de temperatura y humeada del nodo 1

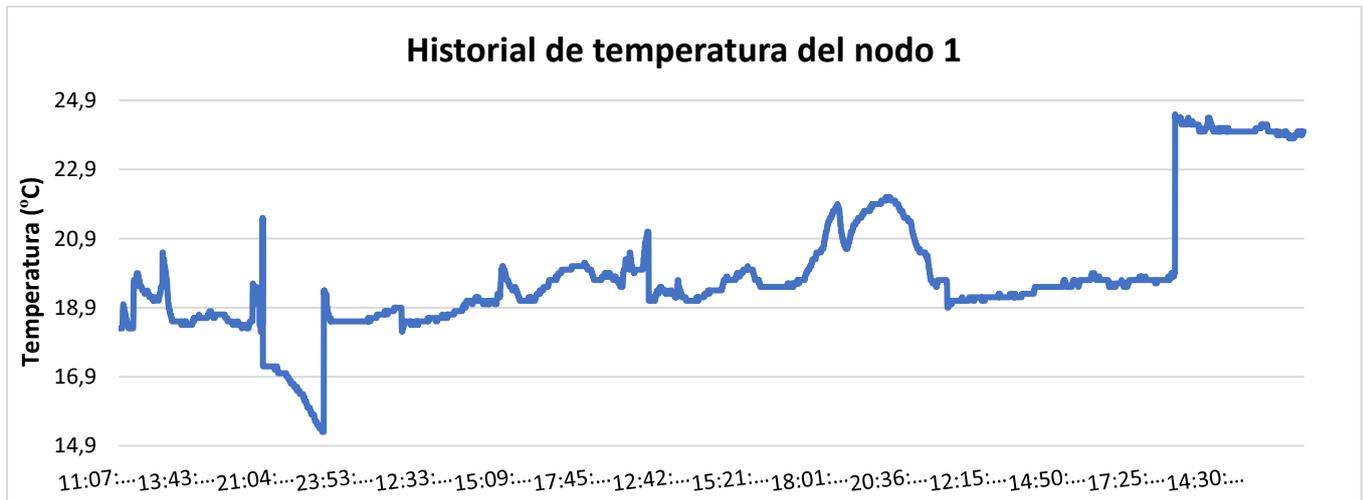
Nodo	Estadísticas	Temperatura °C	Humedad %
Nodo 1	Valor mínimo	15,30 °C	66 %
	Valor máximo	24,5 °C	87,3 %
	Promedio	19,85 °C	73,62 %
	Tiempo de ejecución	6 días	

Nota. Obtenido de: Elaborado por el autor.

En la **Figura 70** se muestra el historial de la temperatura del nodo 1. Los análisis de los datos revelaron que no se registraron temperaturas elevadas que pudieran indicar alertas sobre el estado estructural del puente. Este hecho resultó beneficioso para la integridad del puente durante el periodo de pruebas de funcionamiento del prototipo.

Figura 70.

Historial de temperatura del nodo 1

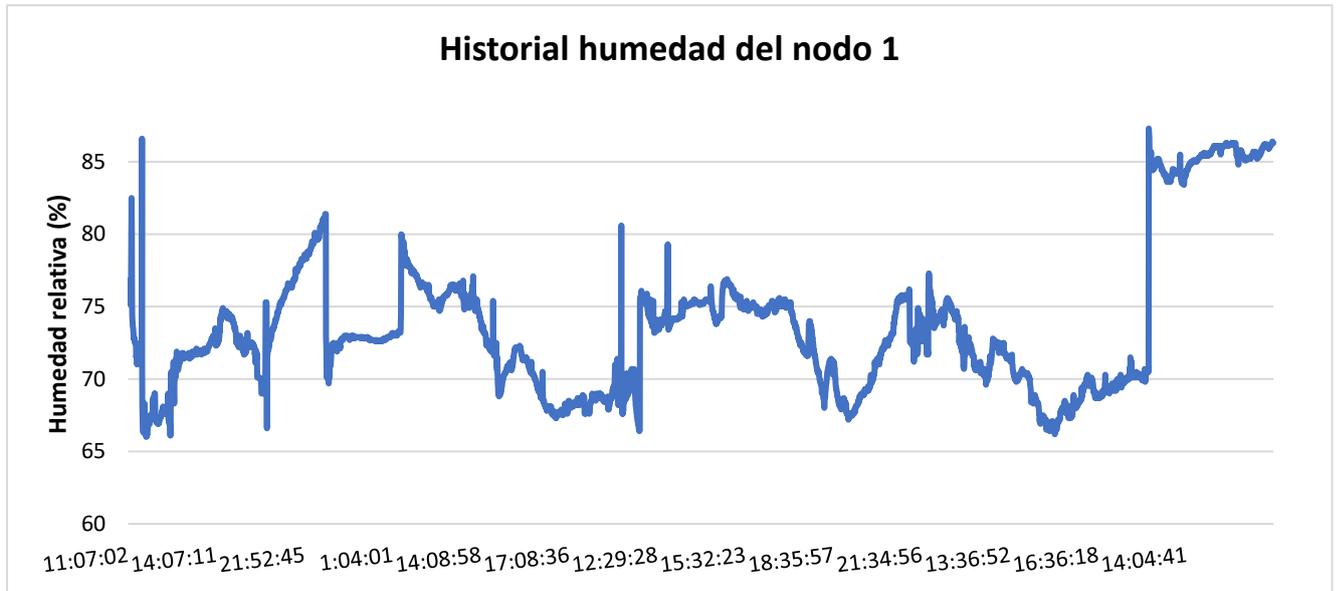


Nota. Obtenido de: Elaborado por el autor.

Con respecto a los análisis de datos de la humedad para el caso del primer nodo, como se observa en la **Figura 71**. Los datos indicaron que existió un promedio de humedad de 73,62% alcanzando un valor máximo de 87,3% y un mínimo de 66%. Los análisis de estos datos resultan llamativos ya que muestran que el puente está siendo afectado por la humedad provocando crecimiento de musgos y vegetación e incluso produciéndose la carbonatación.

De acuerdo con (Morán et al., 2017) indica que “La carbonatación se desarrolla a mayor velocidad en un rango de humedad comprendido entre el 55 y 85%. Esto es debido a que para que se produzca la carbonatación es necesario que exista agua, pero también es necesario que existan huecos de aire para que se difunda el CO₂. Sin embargo, una vez que se produce la carbonatación la velocidad de corrosión será significativa si la humedad relativa es superior al 80%” (p.02).

Figura 71.
Historial de humedad del nodo 1



Nota. Obtenido de: Elaborado por el autor.

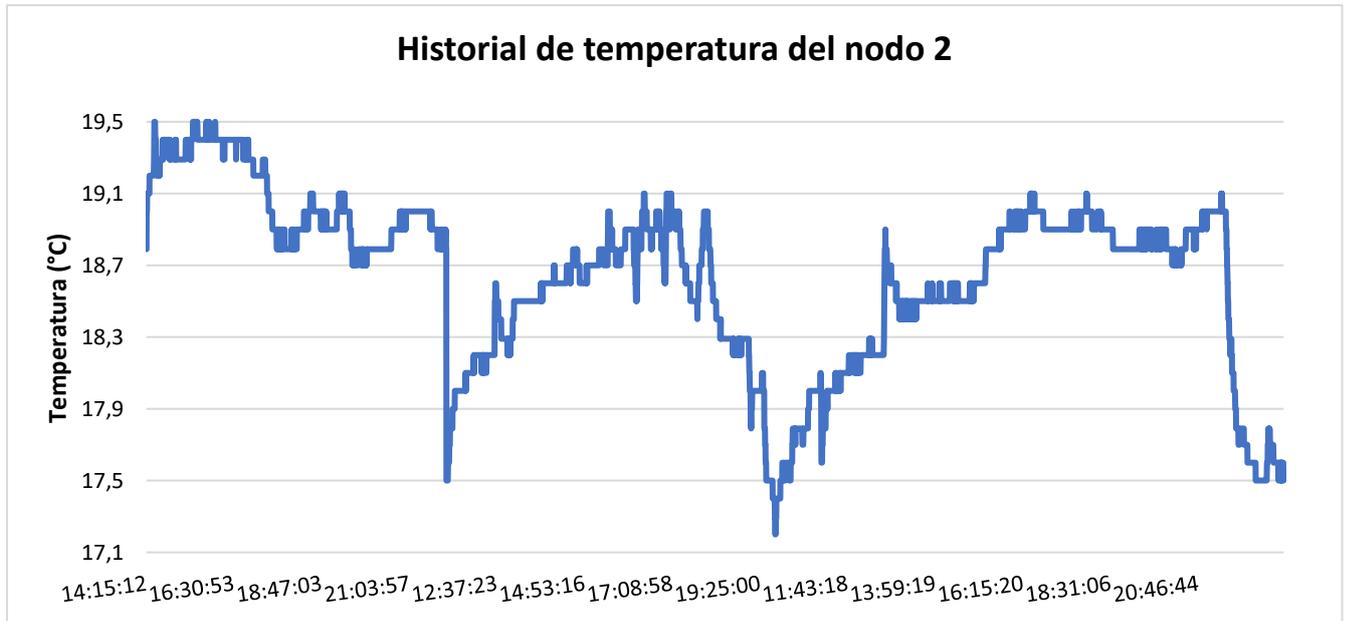
Por otra parte, en la **Tabla 33** se presentan los resultados de temperatura y humedad del segundo nodo. En primer lugar, los resultados de temperatura indican un promedio de 18,64 °C, alcanzando una temperatura máxima de 19,5 °C y una mínima de 17,2 °C. De forma similar al primer nodo, estos resultados no son alarmantes para la integridad del puente.

Tabla 33.
Resultados de temperatura y humedad del nodo 2

Nodo	Estadísticas	Temperatura °C	Humedad %	Sensación térmica °C
Nodo 2	Valor mínimo	17,2 °C	61,3 %	16,59 °C
	Valor máximo	19,5 °C	84,4 %	19,47 °C
	Promedio	18,64 °C	70,92 %	18,41 °C
	Tiempo de ejecución	6 días		

Nota. Obtenido de: Elaborado por el autor.

Figura 72.
Historial de temperatura del nodo 2



Nota. Obtenido de: Elaborado por el autor.

En cuanto análisis de humedad del segundo nodo, en la **Figura 73** se muestra el historial de la humedad relativa. Durante el proceso de pruebas, se observó un promedio de humedad del 70,92%, con un valor máximo de 84,4% y un mínimo de 61,3%. Estos resultados indican que existe la posibilidad de que se produzcan corrosiones en la estructura del puente debido a las condiciones tropicales de la zona, que propician el crecimiento de musgos u otras vegetaciones que podrían degradar el hormigón del puente.

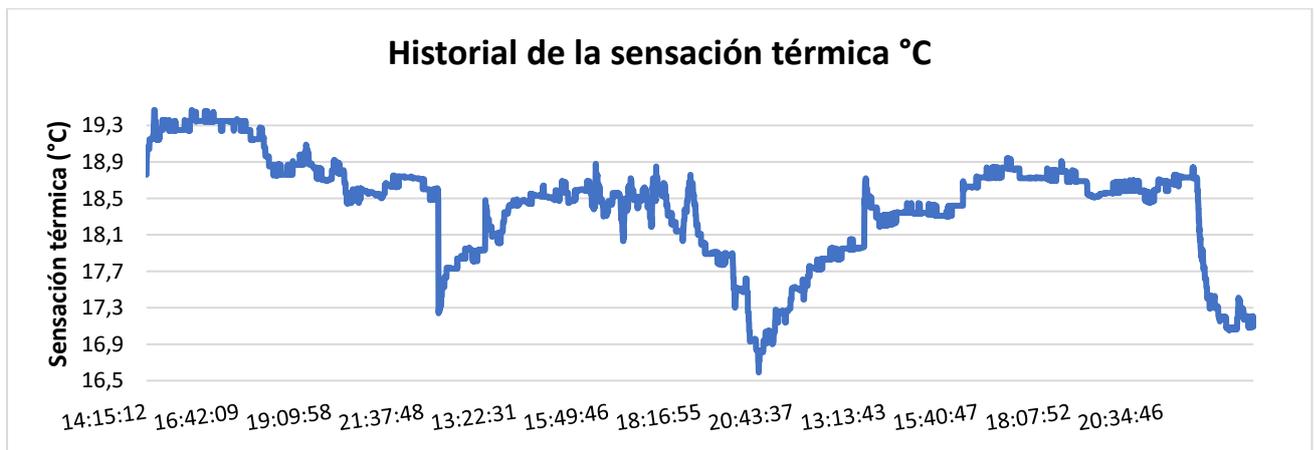
Figura 73.
Historial de humedad del nodo 2



Nota. Obtenido de: Elaborado por el autor.

Adicionalmente, también se obtuvieron resultados de la sensación térmica en función de la temperatura y la humedad relativa. Aunque esta variable no estaba inicialmente contemplada en los objetivos planteados, se utilizó para evaluar cómo el cuerpo humano percibe la combinación de la temperatura del aire y la humedad relativa. Los resultados se presentan en la **Figura 74**, donde se observó un promedio de 18,41°C, con un valor máximo de 19,74 °C y un mínimo de 16,59 °C, de acuerdo con los análisis realizados.

Figura 74.
Historial de la sensación térmica



Nota. Obtenido de: Elaborado por el autor.

6.6.3 Velocidad del viento

El anemómetro estuvo en funcionamiento durante 6 días para medir la velocidad del viento en las cercanías del puente. Los datos analizados incluyeron el promedio, el valor máximo y el valor mínimo de las velocidades del viento registradas en el área. Estos datos fueron vitales para evaluar el efecto del viento en la estabilidad y seguridad del puente. El monitoreo de la velocidad del viento contribuyó a tomar medidas preventivas y correctivas para garantizar la integridad y funcionalidad del puente en condiciones climáticas variables. Los resultados del análisis de la velocidad del viento se muestran en la **Tabla 34**.

Tabla 34.
Resultados de la velocidad del viento

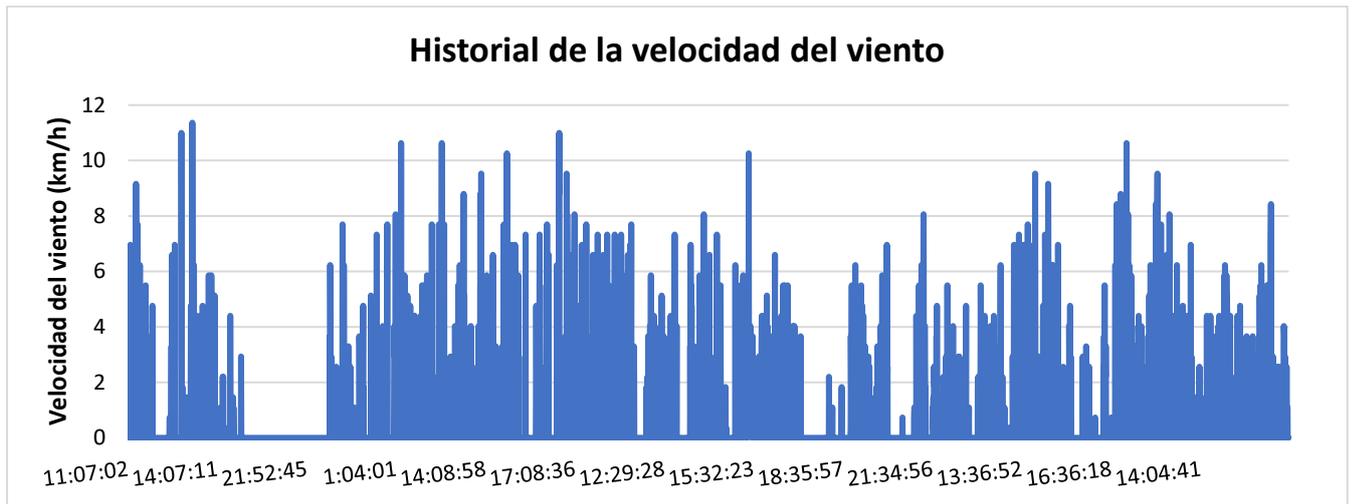
Nodo	Estadísticas	Velocidad del viento (km/h)
Nodo 1	Valor mínimo	0,37 km/h
	Valor máximo	11,36 km/h
	Promedio	2,91 km/h
	Tiempo de ejecución	6 días

Nota. Obtenido de: Elaborado por el autor.

De acuerdo al análisis de la **Figura 75**, se observó un promedio de 2,91Km/h, con un valor máximo de 11,36Km/h y un mínimo de 0,37Km/h. Estos resultados no resultados alarmantes ya que según (Romo, 2018) menciona que “La velocidad básica del viento de diseño para Ecuador debería ser tomada como 100 Km/h en puentes y viaductos” (p.68).

Figura 75.

Historial de la velocidad del viento



Nota. Obtenido de: Elaborado por el autor.

6.6.4 Acelerómetro

Por último, el acelerómetro ADXL345 se desplegó durante 6 días para medir las aceleraciones en los ejes X, Y y Z del puente. Los datos de aceleración registrados proporcionaron el promedio, el valor máximo y el valor mínimo de las aceleraciones experimentadas en cada eje. Estos datos fueron cruciales para el monitoreo estructural del puente, ya que permitieron detectar movimientos, vibraciones y cargas dinámicas que podrían indicar posibles deformaciones o daños en la estructura.

Tabla 35.

Resultados del acelerómetro

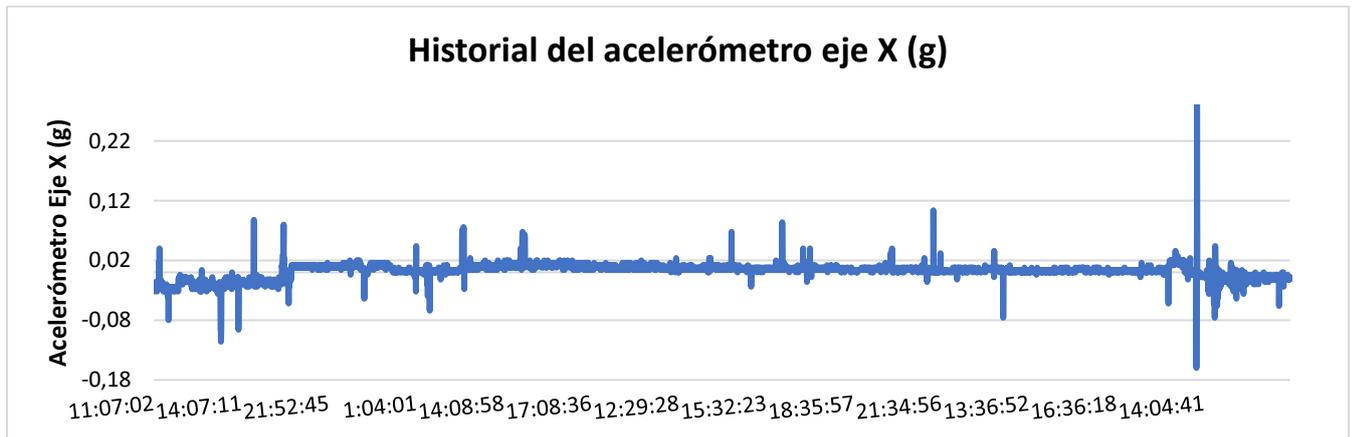
Acelerómetro ADXL345 (g)			
Estadísticas	Eje X	Eje Y	Eje Z
Promedio	0,00246431	-0,03493223	1,09747269
Mínimo	-0,159945369	-0,27590576	0,98366398
Máximo	0,283903033	0,11196175	1,45150411

Nota. Obtenido de: Elaborado por el autor.

En primer lugar, se analizaron los resultados de las aceleraciones del eje "X". Como se muestra en la **Figura 76** existió un promedio de 0,0024g con un valor máximo de 0,28g y un mínimo de -0,15g. Además, como se aprecia en la figura existen ligeras anomalías pero que no son relevantes para una alerta.

Figura 76.

Historial del acelerómetro eje x (g)

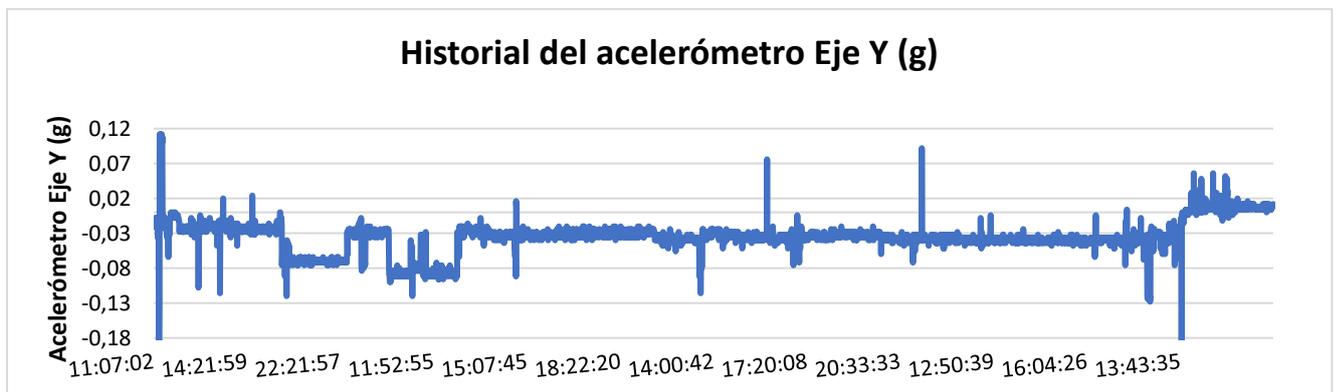


Nota. Obtenido de: Elaborado por el autor.

Por otra parte, en la **Figura 77** correspondiente al eje "Y", se obtuvo un promedio de -0,034g, con un valor máximo de -0,27g y un mínimo de 0,11g. Al igual que en el eje "X", se observaron ligeras vibraciones debidas a las cargas vehiculares y, en ocasiones, a las fuerzas del viento. Sin embargo, estas vibraciones no representan un motivo de preocupación significativo.

Figura 77.

Historial del acelerómetro eje y (g)



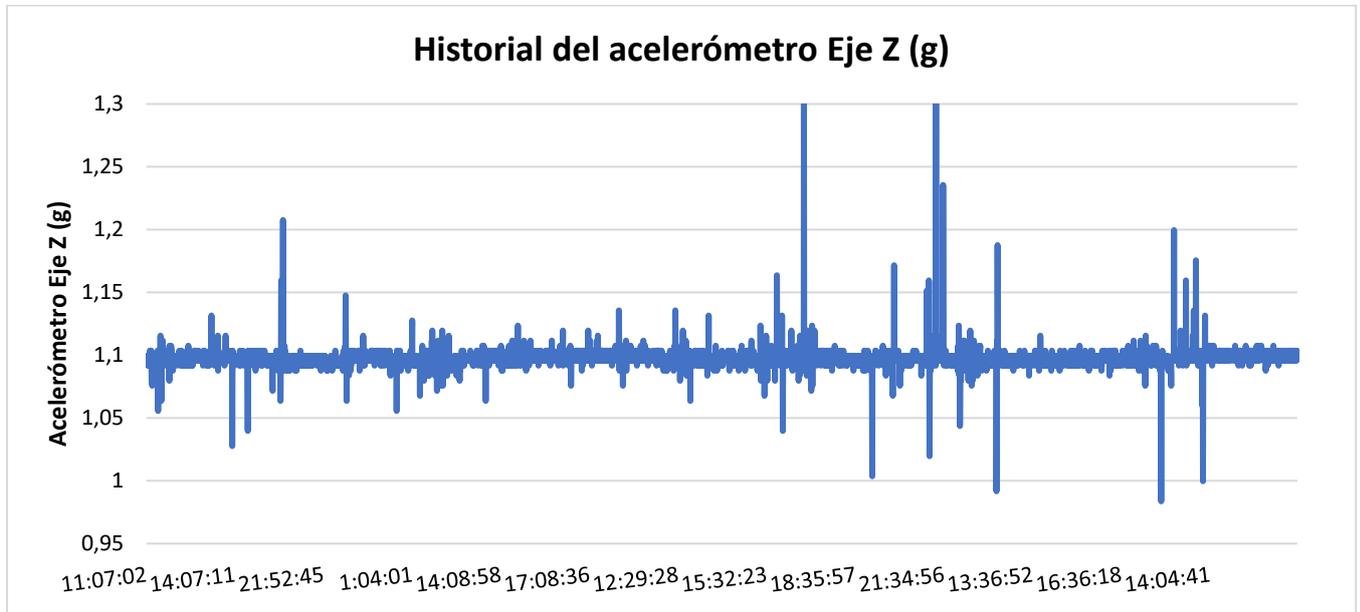
Nota. Obtenido de: Elaborado por el autor.

En relación al eje "Z", se llevaron a cabo análisis exhaustivos de las aceleraciones obtenidas. Como se muestra en la **Figura 78**, se obtuvieron un promedio de 1,10g con un valor máximo de 1,45g y un mínimo de 0,98g. Al igual que en los ejes "X" e "Y", se

detectaron ligeras vibraciones causadas principalmente por las cargas vehiculares y las fuerzas ambientales. Sin embargo, al igual que en los ejes anteriores, estas vibraciones no representan un riesgo significativo para la estabilidad estructural del puente.

Figura 78.

Historial del acelerómetro eje z (g)



Nota. Obtenido de: Elaborado por el autor.

6.7 Resultados de simulación en Omnet++

Con la ayuda del software OMNeT++ es posible observar la arquitectura y cómo funciona la comunicación entre los distintos dispositivos finales y el Gateway. En este caso, el número de nodos es 2. En la **Tabla 36**, es posible observar la energía consumida por el nodo sensor 1 y nodo sensor 2, de tal manera se aprecia que el nodo 1 tiene un mayor consumo de energía mientras que el nodo 2 posee menos consumo de energía.

Tabla 36.

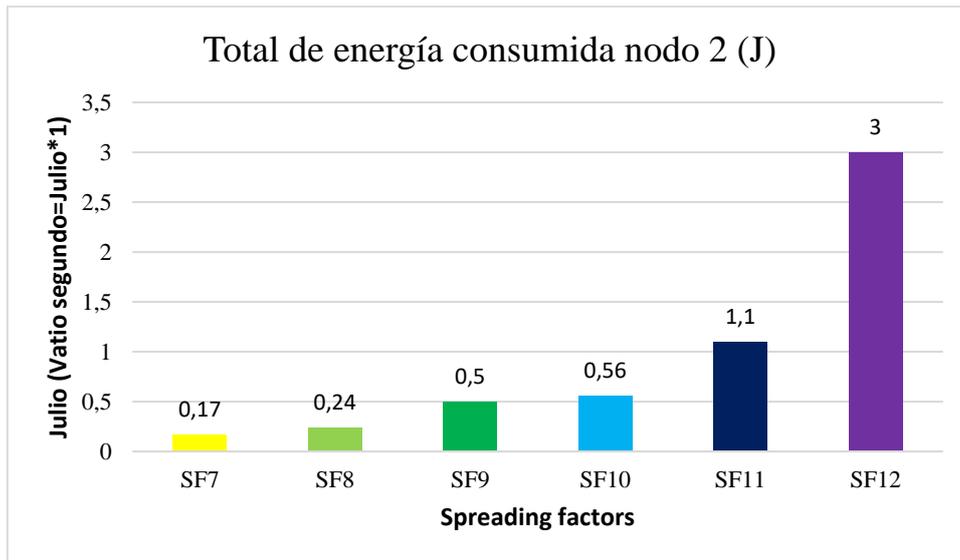
Total de energía consumida por nodo 1 y nodo 2

Spreading factors	Nodo 2 (J)	Nodo 1 (J)
SF7	0,17	0,159
SF8	0,24	0,31
SF9	0,5	0,46
SF10	0,56	0,9
SF11	1,1	1,8
SF12	3	3,1

Nota. Obtenido de: Elaborado por el autor.

En la **Figura 79**, se observa que el consumo de energía del nodo 2 es menor en el factor de dispersión SF7, mientras que el factor de dispersión de SF12 tiene un mayor consumo de energía. Lógicamente, a mayor SF, menor velocidad de transferencia tendremos, pero mayor inmunidad a interferencias.

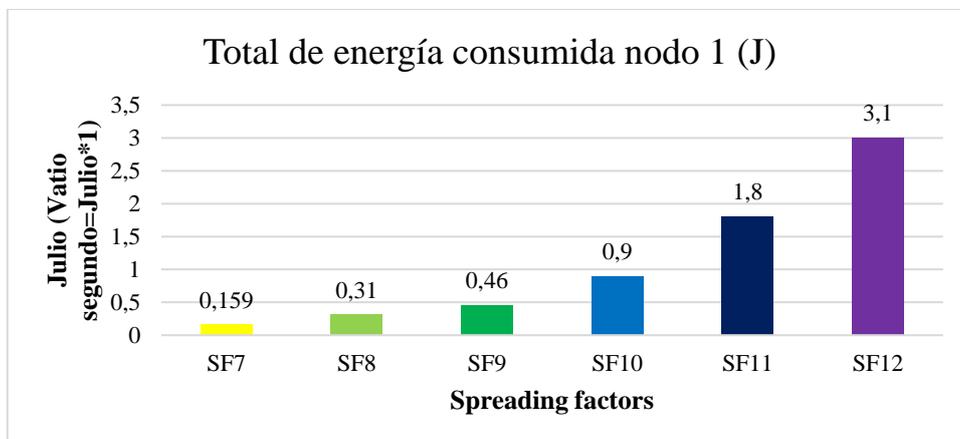
Figura 79.
Energía consumida nodo 2



Nota. Obtenido de: Elaborado por el autor.

Observando la **Figura 80**, el consumo de energía del nodo 1 es mucho mayor que el consumo de energía del nodo 2, la razón por la que se aprecian estas diferencias incide en la distancia que se encuentran con respecto al Gateway. Las distancias pueden ser la causa de que el consumo de energía sea mayor con los nodos más lejanos y menos con los nodos más cercanos al Gateway.

Figura 80.
Energía consumida nodo 1

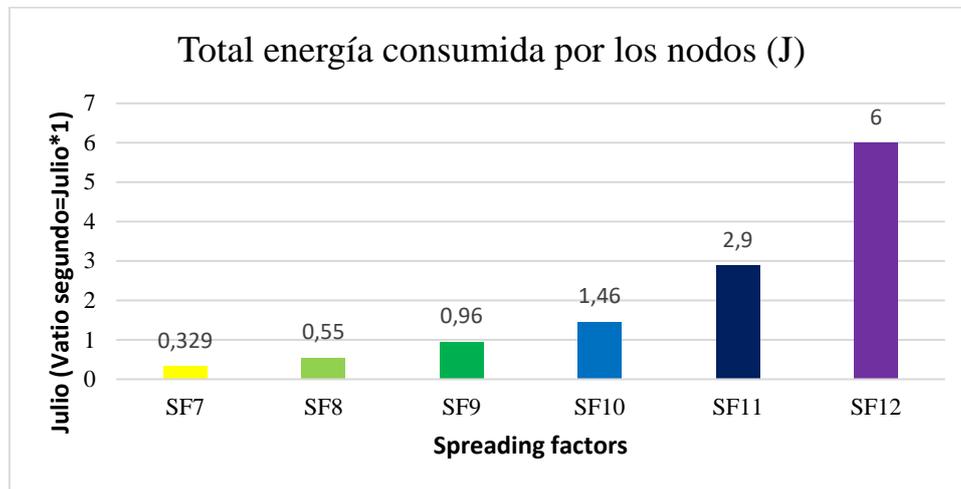


Nota. Obtenido de: Elaborado por el autor.

Por su parte, en la **Figura 81**, se muestra el total de energía consumida por el sistema de monitoreo con los dos nodos sensores en diferentes factores de dispersión. Este resultado de la simulación evidencia que a mayor factor de dispersión mayor consumo de energía.

Figura 81.

Total de energía consumida del sistema WSN



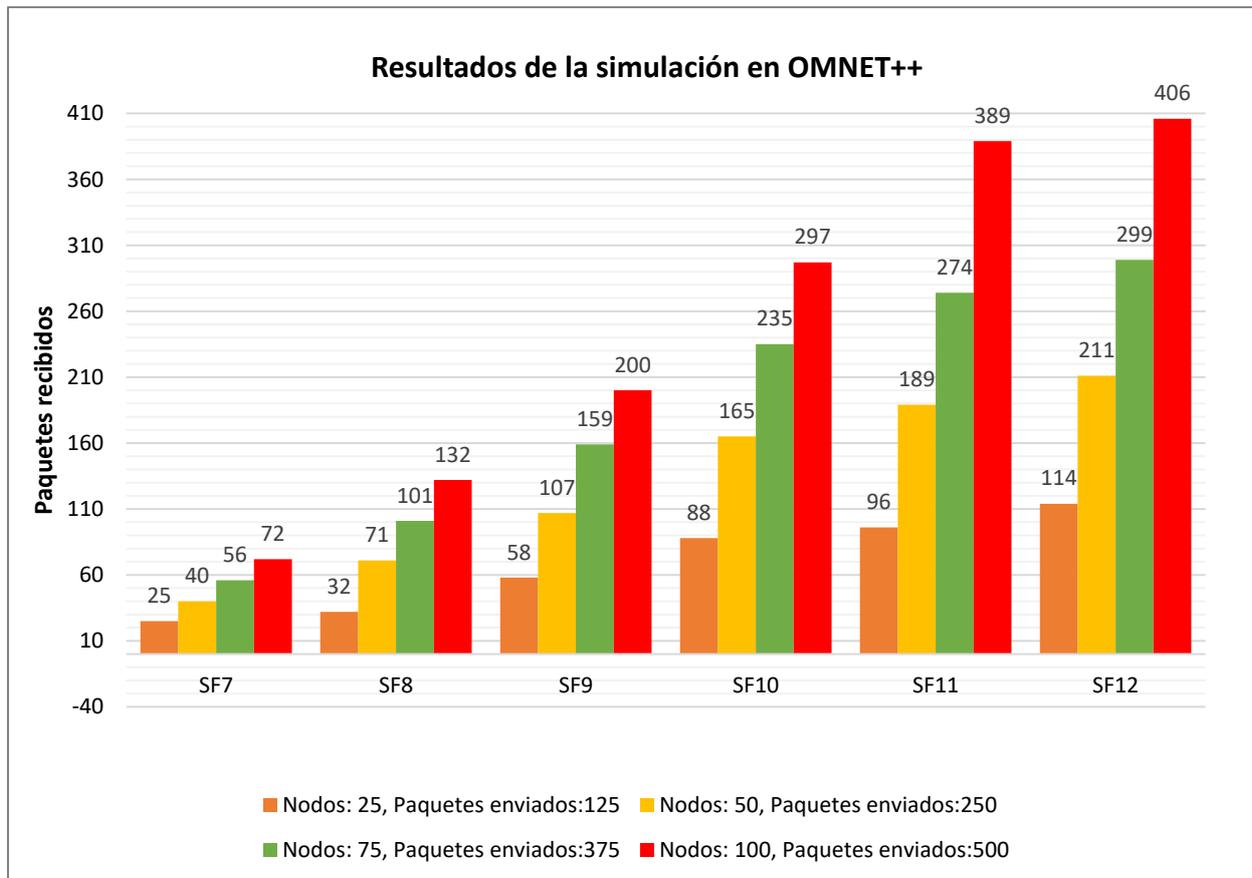
Nota. Obtenido de: Elaborado por el autor.

Adicionalmente, se llevaron a cabo simulaciones con un conjunto específico de nodos (25, 50, 75, 100). Cada nodo enviaba un total de 5 paquetes a un único Gateway. El propósito de estas simulaciones fue determinar la tasa de éxito en la recepción de paquetes por parte del Gateway. De esta manera, se obtuvieron los resultados deseados según el factor de dispersión aplicado en las simulaciones.

En la **Figura 82** se muestran los resultados obtenidos de las simulaciones realizadas. Inicialmente, se llevaron a cabo pruebas con diferentes factores de dispersión (SF) y distintas cantidades de nodos (25, 50, 75 y 100). Se puede observar que a medida que aumenta el SF, el Gateway recibe una mayor cantidad de paquetes. Esto se debe a que un SF más alto resulta en una mayor área de cobertura por parte de los nodos. Sin embargo, es importante destacar que este aumento en la recepción de paquetes también puede estar influenciado por la ubicación espacial de los nodos con respecto al Gateway. Es decir, la disposición física de los nodos en relación con la posición del Gateway puede tener un impacto significativo en la cantidad de paquetes recibidos con éxito.

Figura 82.

Resultados de las simulaciones en OMNET++



Nota. Obtenido de: Elaborado por el autor.

7. Discusión

Las infraestructuras como carreteras, aeropuertos, puertos marítimos, redes de transporte, sistemas de suministro de agua, energía eléctrica, comunicaciones y en especial los puentes proporcionan el soporte necesario para el desarrollo económico, social y urbano de una región o país. Su importancia radica en que forman parte del patrimonio vial en infraestructura de un país, son parte de una red de carreteras que permite la comunicación entre ciudades y reduce el tiempo de traslado para la articulación de actividades comerciales, recreativas, etcétera.

Los riesgos que se corren al no prestar atención al estado estructural de puentes pueden desprenderse grandes consecuencias como posibles fallas estructurales y deterioro prematuro poniendo en peligro la seguridad de las personas y afectar la vida útil de los puentes. Por ende, la necesidad de un sistema de monitoreo de puentes que involucre una tecnología de inspección constante que ayude a obtener datos relevantes para dar los mantenimientos necesarios según sea su deterioro.

La tecnología LoRa ofrece beneficios significativos para el monitoreo estructural de puentes al proporcionar una conectividad de largo alcance, bajo consumo de energía y alta penetración de señal lo que significa que puede superar obstáculos físicos como edificios, vegetación densa o terrenos accidentados. Esto es especialmente importante en el monitoreo de puentes, ya que los sensores pueden estar ubicados en diferentes puntos del puente, algunos de los cuales pueden tener obstrucciones físicas. Estas características permiten una supervisión efectiva y confiable de los puentes, asegurando una detección temprana de problemas y una toma de decisiones basada en datos sólidos para mejorar la seguridad y prolongar la vida útil de las estructuras.

Una de las fortalezas destacadas en este proyecto fue la exitosa conexión de los nodos finales con el servidor de red TTN, encargado de la recolección de datos. Además, esta integración se complementó con la plataforma IoT Ubidots, la cual permitió la visualización de los datos obtenidos por los sensores.

El uso de Ubidots resultó beneficioso para potenciar la red LoRaWAN. Al utilizar los *Dashboards*, se logró una visualización clara y estructurada de la información recopilada por los sensores, lo que facilitó su interpretación y análisis. Esto proporcionó una mayor comprensión de los datos, permitiendo identificar patrones, tendencias y anomalías en tiempo real. Asimismo, la capacidad de personalización de los *Dashboards* en Ubidots brindó

flexibilidad para adaptar la visualización a las necesidades específicas del proyecto y los usuarios finales.

Durante la ejecución del proyecto se presentaron ciertas dificultades, siendo una de ellas la programación de compatibilidad de librerías de los sensores especialmente con el sensor ADXL345 con el módulo LoRa CubeCell, por lo que fue necesario utilizar un microcontrolador como el Arduino Uno para hacer una comunicación I2C e enviar los datos a través de este medio al módulo LoRa. Por otra parte, también se presentaron errores de envío de datos del anemómetro debido a que el módulo CubeCell a pesar de tener una entrada analógica presentaba datos incorrectos siendo una solución utilizar el mismo Arduino Uno para enviar los datos del anemómetro por medio de la comunicación I2C.

Otra de las dificultades que presentaron en la ejecución del proyecto fue hacer una programación adecuada del decodificar del *payload* ya que los sensores enviaban datos tanto negativos y positivos como el ADXL345. Después haber realizado varias pruebas en el código de programación del decodificar se pudo hacer una solución adecuada para decodificar los datos enviados por los nodos finales. Además, resaltar que la API Ubidots permite enviar un máximo de 400 mensajes por día, esto significa encontrar una optimización en los tiempos de envíos de datos de los nodos finales.

8. Conclusiones

En el presente trabajo de integración se ha logrado alcanzar con éxito los objetivos planteados, centrados en el diseño de una Red de Sensores Inalámbricos (WSN) de bajo consumo energético para el monitoreo estructural de puentes basada en la tecnología LoRa. A través de una exhaustiva revisión del estado del arte, el análisis detallado del puente seleccionado y el desarrollo de un prototipo funcional, se ha logrado avanzar significativamente en la implementación de una solución tecnológica efectiva para mejorar la seguridad y prolongar la vida útil mediante la recolección y análisis de datos en tiempo real.

La revisión del estado del arte proporcionó una valiosa comprensión de las últimas tendencias y avances en el campo del monitoreo estructural de puentes mediante redes de sensores inalámbricos. Esta revisión fue fundamental para adoptar enfoques innovadores y tecnologías emergentes que han enriquecido el diseño de la solución propuesta.

El desarrollo del prototipo funcional fue un logro destacado en este proyecto. El desarrollo y despliegue exitoso del prototipo permitió validar la viabilidad y eficacia de la solución propuesta. Se obtuvo información en tiempo real sobre las variables físicas del puente, incluyendo temperatura, humedad, vibraciones y otros parámetros críticos para su monitoreo.

Se logró implementar con éxito la red de sensores inalámbricos, compuesta por sensores DHT22 para medir la temperatura y la humedad, ADXL345 para detectar vibraciones, SW-420 para monitorear movimientos sísmicos y un anemómetro para medir la velocidad. Estos sensores proporcionaron mediciones de las variables físicas claves necesarias para evaluar el comportamiento estructural de los puentes.

La elección de la tecnología LoRa para la transmisión de datos demostró ser acertada, ya que ofreció beneficios significativos para el monitoreo estructural de puentes. Su capacidad de proporcionar una conectividad de largo alcance permitió la cobertura de áreas extensas. Además, su alta penetración de señal garantizó una transmisión de datos confiable, incluso a través de obstáculos físicos.

El análisis del puente seleccionado resultó crucial para determinar el estado estructural y definir los puntos estratégicos donde se instalarían los sensores. Al comprender las condiciones específicas del puente, se aseguró una distribución efectiva de los nodos, optimizando la supervisión y generando datos relevantes para una evaluación precisa de su integridad.

La implementación de la plataforma The Things Network (TTN) y su integración con la API Ubidots facilitaron el procesamiento, almacenamiento y visualización de los datos en tiempo real. Esto permitió obtener información actualizada del escenario y tomar decisiones informadas para prevenir problemas estructurales potenciales.

Durante la investigación, se llevaron a cabo pruebas en condiciones reales, lo que permitió validar la precisión y confiabilidad de los datos recopilados por la red de sensores. Los resultados obtenidos demostraron que el monitoreo continuo de variables físicas como temperatura, humedad, velocidad del viento y vibraciones es crucial para detectar tempranamente cualquier cambio en el comportamiento estructural de los puentes.

Las simulaciones realizadas en OMNeT++ añadieron un valor significativo a la tesis, ya que proporcionaron una perspectiva adicional sobre el rendimiento y la eficiencia de la red de sensores. Estas simulaciones permitieron evaluar diferentes escenarios y condiciones operativas, lo que ayudó a optimizar aún más el diseño y a identificar posibles áreas de mejora.

A pesar de los logros alcanzados, se identificaron algunas limitaciones. La precisión de los sensores utilizados podría mejorarse en futuras investigaciones para aumentar la exactitud de las mediciones. Además, se reconoció la importancia de llevar a cabo pruebas en una variedad de ubicaciones geográficas y condiciones climáticas para evaluar la adaptabilidad del sistema.

En definitiva, esta tesis ha contribuido significativamente al campo del monitoreo estructural de puentes utilizando la tecnología LoRa y el IoT. La implementación exitosa de la red de sensores inalámbricos y la integración con plataformas en la nube han demostrado que este enfoque ofrece una solución efectiva y rentable para mejorar la seguridad y prolongar la vida útil de las infraestructuras críticas. Las recomendaciones para futuras investigaciones y aplicaciones proporcionan una base sólida para continuar avanzando en esta área y desarrollar sistemas de monitoreo aún más sofisticados y precisos para la protección de puentes y otras estructuras

9. Recomendaciones

En el presente estudio, se aborda el diseño de una red de sensores inalámbricos (WSN) de bajo consumo energético para el monitoreo estructural de puentes utilizando la tecnología LoRaWAN. La tecnología LoRaWAN ha demostrado ser una solución eficiente y escalable para la conectividad de dispositivos IoT, permitiendo la transmisión de datos de manera inalámbrica a largas distancias con un consumo de energía reducido. A continuación, se presentan las recomendaciones necesarias para futuras líneas de investigación.

- Se recomienda adquirir los conocimientos teóricos necesarios en base al proyecto planteado antes del comienzo del mismo, con objetivo que no exista problemas a futuro sobre su implementación. Para ello se aconseja utilizar documentos científicos, debido a que es donde se aloja información relevante y verídica.
- Es recomendable tener en cuenta las especificaciones técnicas de los materiales/equipos a utilizar en el proyecto planteado con la finalidad de evitar daños del mismo de esta manera realizar la correcta instalación y configuración.
- Es importante realizar pruebas de campo o calibración de los sensores que se utilizaran en el proyecto ya que los equipos pueden presentar errores de fábrica por ende enviar información incorrecta. Se puede adicionar herramientas como Excel con el complemento COM que integra el software para obtener los datos de los sensores para posteriormente visualizar de la mejor manera la información mediante gráficas.
- Se recomienda diseñar y desplegar una red LoRaWAN con una arquitectura adecuada según las necesidades de aplicación. Esto implica planificar la ubicación y densidad del Gateway para garantizar una cobertura óptima en el área de despliegue, minimizando así los puntos ciegos y las áreas de baja señal. Además, se deben considerar aspectos de escalabilidad y capacidad de la red para soportar un crecimiento futuro.
- Se sugiere ajustar los parámetros de configuración de los dispositivos finales para optimizar el rendimiento de la red. Esto puede incluir la selección adecuada de las frecuencias y canales disponibles, la configuración del factor de esparcimiento (*spreading factor*) y el tamaño de las tramas de datos. Una configuración óptima permitirá mejorar la eficiencia espectral, aumentar la vida útil de la batería de los dispositivos y optimizar la capacidad de la red.

- Es absolutamente necesario seguir correctamente las guías de usuarios proporcionadas por los fabricantes de los dispositivos LoRa. Una mala configuración de los dispositivos podría resultar en fallos en las conexiones o en el envío de datos, e incluso podría causar daños en el hardware de los dispositivos LoRa.
- Además de las variables físicas mencionadas (temperatura, humedad, velocidad del viento y vibraciones), considere la posibilidad de incluir otros parámetros relevantes para el monitoreo estructural de puentes, como la deformación, la presión, la carga, la corrosión y hacer una escalabilidad de la red como en el **Anexo 18**. Esto proporcionará una visión más completa del estado de los puentes y permitirá una evaluación más precisa de su integridad estructural.
- Utilizar los resultados del análisis de datos para establecer un sistema de alerta temprana que notifique a los responsables de mantenimiento y gestión de los puentes cuando se detecten desviaciones significativas en las variables monitoreadas. Esto permitirá una intervención rápida y oportuna para abordar cualquier problema potencial antes de que se convierta en una amenaza para la integridad del puente.

10. Bibliografía

- Aakvaag, N., & Frey, J.-E. (2010). Redes de Sensores inalámbricos, Nuevas soluciones de interconexión para la automatización industrial. *Aab*, 2, 4. http://interelectricas.co/pdf/ABB/02-2006/39-42%25202M631_SPA72dpi.pdf
- Adamu, M. J., Qiang, L., Zakariyya, R. S., Nyatega, C. O., Kawuwa, H. B., & Younis, A. (2021). An Efficient Turbo Decoding and Frequency Domain Turbo Equalization for LTE Based Narrowband Internet of Things (NB-IoT) Systems. *Sensors*, 21(16), 5351. <https://doi.org/10.3390/s21165351>
- Aguilar, D., León, P., Álvarez, J., Aguilar, C., & Martínez, S. (2016). *Evaluation and analysis of a prototype wireless sensor for monitoring critical infrastructures in bridges*. <https://doi.org/10.1109/ICA-ACCA.2016.7778426>
- Alkhatib, A. (2011). *Wireless Sensor Network Architecture*. <https://www.researchgate.net/publication/227352986>
- Almuhaya, M. A. M., Jabbar, W. A., Sulaiman, N., & Abdulmalek, S. (2022). A Survey on LoRaWAN Technology: Recent Trends, Opportunities, Simulation Tools and Future Directions. In *Electronics (Switzerland)* (Vol. 11, Issue 1). MDPI. <https://doi.org/10.3390/electronics11010164>
- Al-Radaideh, A., Bheiry, A., Alawnah, S., & Al-Ali. (2015). *A Wireless Sensor Network Monitoring System for Highway Bridges*.
- Aosong Electronics Co., L. (n.d.). *Digital-output relative humidity & temperature sensor/module DHT22 (DHT22 also named as AM2302) Capacitive-type humidity and temperature module/sensor*.
- Arduino. (2022). *Arduino® UNO R3*. <https://docs.arduino.cc/hardware/uno-rev3>
- Borah, S., Al-Habaibeh, A., & Kromanis, R. (2021). *The Effect of Temperature Variation on Bridges—A Literature Review* (pp. 207–212). https://doi.org/10.1007/978-3-030-63916-7_26
- C. R Farrar, & G. H James. (1997). System identification from ambient vibration measurements on a bridge. In *Journal of Sound and Vibration* (Issue 1).
- Clarence W. de Silva. (n.d.). *Vibration: Fundamentals and Practice, Second Edition*. Retrieved October 25, 2022, from

https://books.google.es/books?id=rU0ANd3X_HIC&lpg=PP1&ots=or3dYtziE-&dq=vibration&lr&hl=es&pg=PR4#v=onepage&q=vibration&f=false

Cristian Camilo Vargas Duran. (2020). *Evaluación de la temperatura superficial puente vehicular de terreros en el municipio de soacha, cundinamarca.*

Daniel, I., & Pernia-Márquez, A. (2004). *Introducción a la medición de vibración.* https://www.academia.edu/29108458/INTRODUCCI%C3%93N_A_LA_MEDICI%C3%93N_DE_VIBRACI%C3%93N?bulkDownload=thisPaper-topRelated-sameAuthor-citingThis-citedByThis-secondOrderCitations&from=cover_page

Devices, A. (n.d.). *ADXL345 (Rev. G).*

Digikey. (2023). *SW-420.* https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/Grove_Vibration_Sensor_SW-420_Web.pdf

Ekanayake, J. C., & Hedley, C. B. (2018). Advances in Information Provision from Wireless Sensor Networks for Irrigated Crops. *Wireless Sensor Network*, 10(04), 71–92. <https://doi.org/10.4236/wsn.2018.104004>

Electronics Hub. (2019, March 25). *Basics of Wireless Sensor Networks (WSN) / Classification, Topologies, Applications.* <https://www.electronicshub.org/wireless-sensor-networks-wsn/>

Farrell, S. (2018). *Low-Power Wide Area Network (LPWAN) Overview.* <https://doi.org/10.17487/RFC8376>

Harold E. Price, N. (1995). *Spread Spectrum - It's not just for breakfast anymore!* <https://www.qsl.net/n9zia/ss.qexss.html>

He, J., Wang, N. K. K., & Muraleetharan, P. E. (n.d.). *Temperature Effects in Bridge Condition Evaluation and Capacity Rating in Oklahoma.* www.sptc.org

He, J., Xin, H., Wang, Y., & Correia, J. A. F. O. (2021). Effect of temperature loading on the performance of a prestressed concrete bridge in Oklahoma: Probabilistic modelling. *Structures*, 34, 1429–1442. <https://doi.org/10.1016/J.ISTRUC.2021.08.007>

Heltec. (2022). *CubeCell.* <https://heltec.org/project/htcc-ab01/>

- ITU. (2022). *Sobre la Unión Internacional de Telecomunicaciones (ITU)*.
<https://www.itu.int/es/about/Pages/default.aspx>
- Katranji. (n.d.). *Wind Sensor (Pulse-Type) Model: JL-FS2 Features: Specifications*.
- Kurata, M., Kim, J., Lynch, J. P., van der Linden, G. W., Sedarat, H., Thometz, E., Hipley, P., & Sheng, L.-H. (2013). Internet-Enabled Wireless Structural Monitoring Systems: Development and Permanent Deployment at the New Carquinez Suspension Bridge. *Journal of Structural Engineering*, 139(10), 1688–1702.
[https://doi.org/10.1061/\(asce\)st.1943-541x.0000609](https://doi.org/10.1061/(asce)st.1943-541x.0000609)
- Logitek, & Fernando Campos. (2020, July 21). *LPWAN*.
<https://www.m2mlogitek.com/lpwan-que-son-y-para-que-se-utilizan/>
- LoRa Alliance. (2018). *LoRaWAN™ 1.0.3 Specification*.
- LoRa Alliance. (2019). *LoRaWAN® Is Secure (but Implementation Matters)*.
- LoRa Alliance. (2021). *LoRaWAN Architecture*. <https://lora-alliance.org/about-lorawan/>
- LoRa Alliance. (2022). *LoRa Alliance*. <https://lora-alliance.org/about-lora-alliance/>
- Madeleine Renom. (2011). *ANEMOMETRIA: Principios básicos de las mediciones atmosféricas*.
http://www.meteorologia.edu.uy/wp-content/uploads/2019/principios_bas_medicones_atms/Bolilla5-Anemometria.pdf
- Marín, G., & Maldonado, N. (2022). Study of the causes of bridge collapse in Ecuador (2000-2022). *MQR Investigar*, 6(4), 368–395.
<https://doi.org/10.56048/mqr20225.6.4.2022.368-395>
- Mejía, Á., & Salazar, J. (2019). *Diseño e implementación de un prototipo para el análisis de salud estructural de un puente a través de una red de sensores inalámbricos utilizando el software Matlab*.
- Milesight. (2022). *UG63 Datasheet*. <https://www.milesight-iot.com/lorawan/gateway/ug63/>
- Morán, E. C., Ramos, N. R., Pinto, F. T., & Capacchione, M. (2017). *Measurement of moisture content in existing concrete bridge decks*.
- Naik, N. (2018). LPWAN Technologies for iot systems: choice between ultra narrow band and spread spectrum. *4th IEEE International Symposium on Systems Engineering, ISSE 2018 - Proceedings*. <https://doi.org/10.1109/SYSENG.2018.8544414>

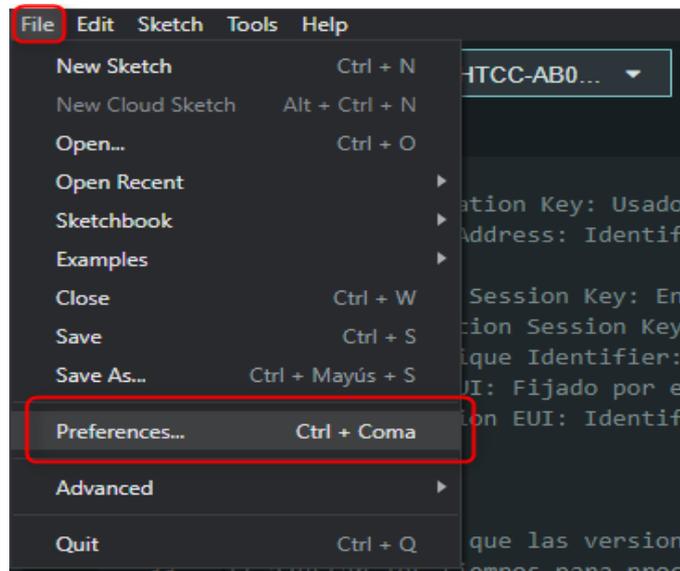
- Naranjo, G. L. (2021). *Sistema de monitoreo de salud estructural (SHM) de puentes de hormigón mediante redes WSN*.
- Ospina Rojas, E. (2021). *Diseño e implementación de un anemómetro usando sensores ultrasónicos*.
<https://repositorio.itm.edu.co/handle/20.500.12622/5560#.Y1oMakYPwoY.mendeley>
- Petäjäjärvi, J., Mikhaylov, K., Yasmin, R., Hämäläinen, M., & Iinatti, J. (2017). Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring. *International Journal of Wireless Information Networks 2017 24:2, 24(2)*, 153–165.
<https://doi.org/10.1007/S10776-017-0341-8>
- Poddaeva, O., Fedosova, A., & Gribach, J. (2019). The study of wind effects on the bridge constructions. *E3S Web of Conferences*, 97, 03030.
<https://doi.org/10.1051/e3sconf/20199703030>
- Proyecto Arduino. (2020). *Arduino Uno R3*. <https://proyectoarduino.com/arduino-uno-r3/>
- Quezada, P., Chango, P., Benavides, V., Ramirez, R., & Enciso, L. (2017). *System Monitoring for bridges structure*. <https://www.researchgate.net/publication/319312309>
- Rabesiranana, N., Rajaobelison, J., Francisco Pascal Elias, T., Jean Baptiste, R., Hery, A., Naivo, R., & Joël, R. (2019). IoT-based Environmental and Ionizing Radiation Monitoring System. *International Journal of Innovative Research in Science*, 8.
<https://doi.org/10.15680/IJIRSET.2019.0803258>
- Reynders, B., Wang, Q., & Pollin, S. (2018). A LoRaWAN module for ns-3: Implementation and evaluation. *ACM International Conference Proceeding Series*, 61–68.
<https://doi.org/10.1145/3199902.3199913>
- Romo, M. (2018). *Capítulo III. Cargas en puentes y viaductos*.
<https://www.researchgate.net/publication/327103995>
- Salgado, C., & Carranza Geomayra. (2016). *Diseño de un prototipo para la implementación de un sistema de monitoreo de vibración e inclinación de un puente mediante una red de sensores inalámbricos*.
- SEMTECH. (2015). *AN1200.22 LoRa™ Modulation Basics*.

- SEMTECH. (2022). *LoRa y LoRaWAN: Descripción general / Portal para desarrolladores*.
<https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan>
- Semtech Corporation. (2020). *LoRa and LoRaWAN: A Technical Overview LoRa® and LoRaWAN®: A Technical Overview*.
- Sharma, D., Verma, S., & Sharma, K. (2013). *Network Topologies in Wireless Sensor Networks: A Review 1. 4*. <http://www.iject.org/vol4/spl3/c0116.pdf>
- Sinchi, N. (2022). *Monitoreo de la salud estructural de un puente utilizando la técnica PP-GNSS*.
- Sinha, R. S., Wei, Y., & Hwang, S. H. (2017). A survey on LPWA technology: LoRa and NB-IoT. In *ICT Express* (Vol. 3, Issue 1, pp. 14–21). Korean Institute of Communications Information Sciences. <https://doi.org/10.1016/j.icte.2017.03.004>
- The things network. (2021). *Message Types*.
- UIT-T. (2012). *UIT-T Rec. Y.2060 (06/2012) Descripción general de Internet de las Cosas*.
- Valeria Quinde. (2016). *Monitoreo en tiempo real de la salud estructural de puentes (MSE) para la detección temprana de fallas mediante un prototipo de red de sensores inalámbricos ubicados en el puente del Río Boquerón-Catamayo*.
- Windcrane. (2019, December 8). *Wind monitoring importance in bridge construction and operation*. <https://www.windcrane.com/blog/construction/wind-monitoring-importance-bridge-construction-and-operation>
- Zambrano, C., & Zambrano, D. (2020). *Monitoreo de puentes en regiones andinas remotas*.

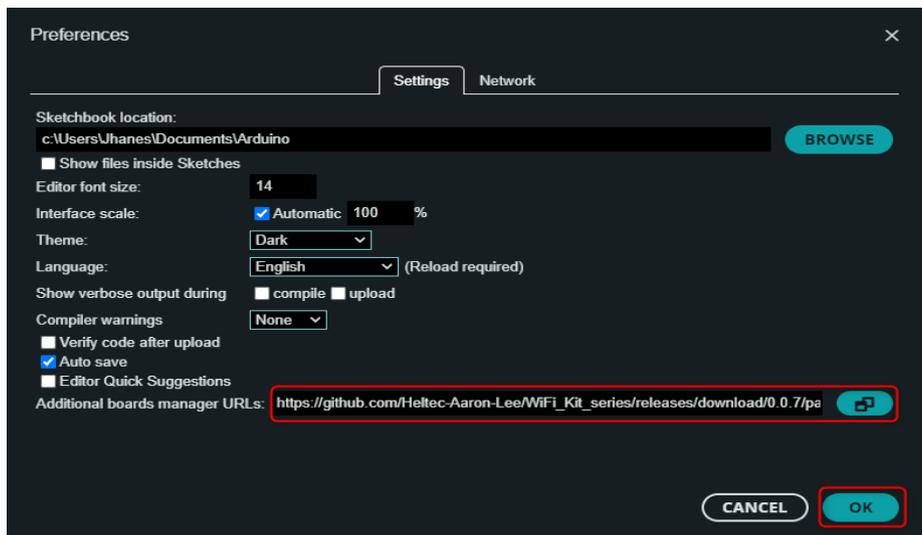
11. Anexos

Anexo 1. Guía de instalación HELTEC serie CubeCell

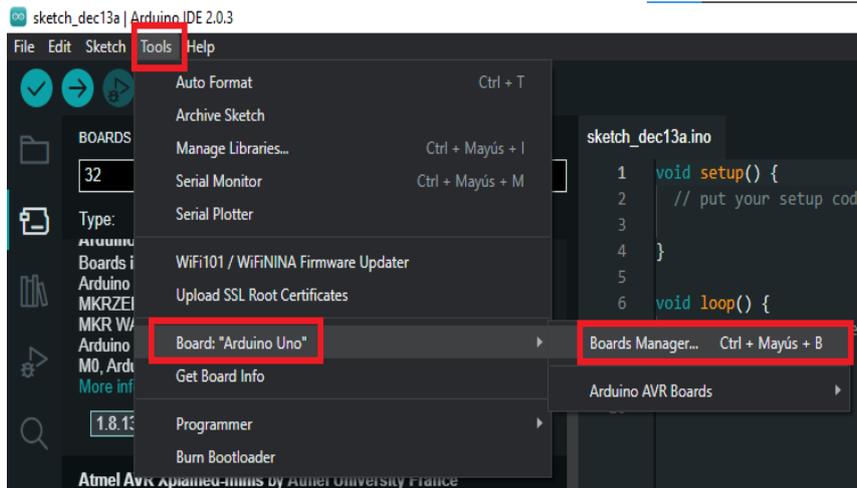
Paso 1: Descargar el soporte de Arduino-CubeCell. Para ello abrir el IDE de Arduino y hacer clic en File-Preferenes.



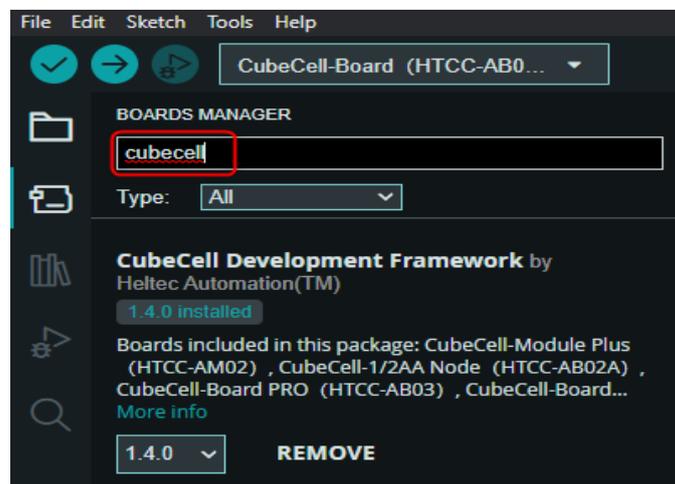
Paso 2: Heltec proporciona un paquete cubecell, en el cual en el menú de preferenes introducimos el URL https://resource.heltec.cn/download/package_CubeCell_index.json y le damos en aceptar.



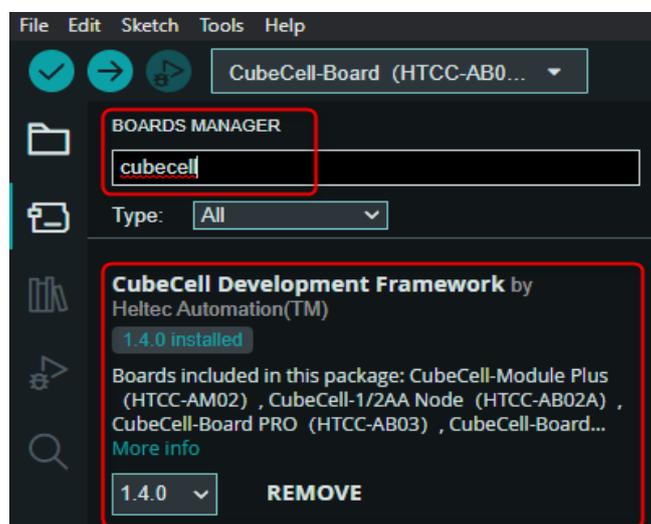
Paso 3: Abrimos la gestión de tarjetas de Arduino IDE y buscamos CubeCell.



Paso 4: Instalamos el Heltec CubeCell Series Dev-boards realizado por Heltec Automation.



Paso 5: finalmente, instalamos la placa CubeCell, Heltec CubeCell Development Framework.



Anexo 2. Código fuente *LoRaNetworkTest.ned*

```
package inet.examples.flora;
import inet.applications.udpapp.UdpBasicApp;
import inet.node.inet.Router;
import inet.node.internetcloud.InternetCloud;
import inet.flora.loraphy.LoRaMedium;
import inet.flora.loranode.LoRaNode;
import inet.flora.loranode.LoRaGW;
import inet.node.inet.StandardHost;
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;
import inet.node.ethernet.Eth1G;
import inet.visualizer.contract.IIntegratedVisualizer;
import inet.environment.common.PhysicalEnvironment;
@license(LGPL);
network LoRaNetworkTest{
  parameters:
    int numberOfNodes = default(5);
    int numberOfGateways = default(1);
    int networkSizeX = default(500);
    int networkSizeY = default(500);
    @display("bgb=355.824,356.832");
  submodules:
    visualizer: <default(firstAvailableOrEmpty("IntegratedVisualizer"))> like
    IIntegratedVisualizer if typename != "" {
      parameters:
        @display("p=100,400;is=s");}
    physicalEnvironment: PhysicalEnvironment {
      parameters:
        @display("p=100,200;is=s");}
    loRaNodes[numberOfNodes]: LoRaNode {
      @display("p=83.664,233.856"); }
    loRaGW[numberOfGateways]: LoRaGW {
      @display("p=83.664,150.192;is=s"); }
    LoRaMedium: LoRaMedium {
      @display("p=228,97");}
    networkServer: StandardHost {
      parameters:
        @display("p=314,27");}
    configurator: Ipv4NetworkConfigurator {
      parameters:
        assignDisjunctSubnetAddresses = false;
        @display("p=314,98");}
    internetCloud: InternetCloud {
      @display("p=172,27");}
    gwRouter[numberOfGateways]: Router {
      @display("p=71,23"); }
    nsRouter: Router {
      @display("p=242,27"); }
  connections:
    networkServer.ethg++ <--> Eth1G <--> nsRouter.ethg++;
    nsRouter.pppg++ <--> Eth1G <--> internetCloud.pppg++;
    for i=0..numberOfGateways-1 {
      internetCloud.pppg++ <--> Eth1G <--> gwRouter[i].pppg++;
      gwRouter[i].ethg++ <--> Eth1G <--> loRaGW[i].ethg++; }
}
```

Anexo 3. Código fuente n2-gw1.ini

```
network = inet.examples.flora.LoRaNetworkTest
**.maxTransmissionDuration = 4s
#cmdenv-autoflush = true
#cmdenv-status-frequency = 10000000s
**.vector-recording = false
#configuración básica del concentrador LoRa
rng-class = "cMersenneTwister"
**.loRaGW[*].numUdpApps = 1
**.loRaGW[0].packetForwarder.localPort = 2000
**.loRaGW[0].packetForwarder.destPort = 1000
**.loRaGW[0].packetForwarder.destAddresses = "networkServer"
**.loRaGW[0].packetForwarder.indexNumber = 0
#configuración básica del servidor de red LoRa
**.networkServer.numApps = 1
**.networkServer.**.evaluateADRinServer = false
**.networkServer.app[0].typename = "NetworkServerApp"
**.networkServer.app[0].destAddresses = "loRaGW[0]"
**.networkServer.app[0].destPort = 2000
**.networkServer.app[0].localPort = 1000
**.networkServer.app[0].adrMethod =>{"avg"}
**.numberOfPacketsToSend = 0 #${numero de paquetes = 200..5000 step 200} #100 #
obviamente 0 significa número infinito de paquetes
sim-time-limit = 1d #tiempo de ejecución de la simulación (en días)
simtime-resolution = -11 #resolución de simulación
repeat = 2 //2 #número de repeticiones
#define la frecuencia de envío de los paquetes
**.timeToFirstPacket = exponential(1000s)
**.timeToNextPacket = exponential(1000s)
**.alohaChannelModel = false
#Características de nodos (dispositivos finales)
**.loRaNodes[*]**.initFromDisplayString = false
**.loRaNodes[*]**.evaluateADRinNode = false
**.loRaNodes[*]**.initialLoRaBW = 125 kHz #bandwidth
**.loRaNodes[*]**.initialLoRaCR = 4 #code rate
#Características del gateway
**.LoRaGNic.radio.iAmGateway = true
**.loRaGW[*]**.initFromDisplayString = false
#características de consumo de energía
**.loRaNodes[*].LoRaNic.radio.energyConsumer.typename = "LoRaEnergyConsumer"
**.loRaNodes[*]**.energySourceModule = {"^}.IdealEpEnergyStorage"
**.loRaNodes[*].LoRaNic.radio.energyConsumer.configFile =
xmldoc("../energyConsumptionParameters.xml")
#Características generales
#establecer los valores mínimos para el tamaño del área en la que se #encuentra
la red simulada se encuentra
**.sigma = 3.57
**.constraintAreaMinX = 0m
**.constraintAreaMinY = 0m
**.constraintAreaMinZ = 0m
**.constraintAreaMaxZ = 0m
LoRaNetworkTest.**.radio.separateTransmissionParts = false
LoRaNetworkTest.**.radio.separateReceptionParts = false
# configuración sitio de backhaul
**.ipv4Delayer.config = xmldoc("../cloudDelays.xml")
# Medios de transmisión
**.radio.radioMediumModule = "LoRaMedium"
```

```

**.LoRaMedium.pathLossType = "LoRaLogNormalShadowing"
#**.LoRaMedium.pathLossType = "LoRaPathLossOulu"
#intervalo de tiempo mínimo en el que pueden considerarse dos señales
superpuestas como corriendo el uno hacia el otro
**.minInterferenceTime = 0s
**.displayAddresses = false
**.numberOfNodes = 2 #numero de nodos
**.numberOfGateways = 1 #numero de GW
**.constraintAreaMaxX = 2376.0m
**.constraintAreaMaxY = 1090.0m
**.loRaGW[0]**.initialX = 250m
**.loRaGW[0]**.initialY = 255m
output-scalar-file = ./results/n100-gw1-s${runnumber}.ini.sca #guardar resultados
**.loRaNodes[0]**.initialX = 450m
**.loRaNodes[0]**.initialY = 450m
**.loRaNodes[1]**.initialX = 488.29m
**.loRaNodes[1]**.initialY = 498m
#La potencia de transmisión es de al menos 2 dBm y un máximo de 14 dBm
**.loRaNodes[0]**.initialLoRaSF = 12 #factor de dispersion
**.loRaNodes[0]**.initialLoRaTP = 14dBm #potencia de transmision
**.loRaNodes[1]**.initialLoRaSF = 12 #factor de dispersion de
**.loRaNodes[1]**.initialLoRaTP = 14dBm #potencia de transmisión

```

Anexo 4. Código fuente energyConsumptionParameters.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<root>
  <supplyVoltage value="3.3"/>
  <receiverReceivingSupplyCurrent value="9.7"/>
  <receiverBusySupplyCurrent value="9.7"/>
  <idleSupplyCurrent value="0.0001"/>
  <txSupplyCurrents>
    <txSupplyCurrent txPower="2" supplyCurrent="24"/>
    <txSupplyCurrent txPower="3" supplyCurrent="24"/>
    <txSupplyCurrent txPower="4" supplyCurrent="24"/>
    <txSupplyCurrent txPower="5" supplyCurrent="25"/>
    <txSupplyCurrent txPower="6" supplyCurrent="25"/>
    <txSupplyCurrent txPower="7" supplyCurrent="25"/>
    <txSupplyCurrent txPower="8" supplyCurrent="25"/>
    <txSupplyCurrent txPower="9" supplyCurrent="26"/>
    <txSupplyCurrent txPower="10" supplyCurrent="31"/>
    <txSupplyCurrent txPower="11" supplyCurrent="32"/>
    <txSupplyCurrent txPower="12" supplyCurrent="34"/>
    <txSupplyCurrent txPower="13" supplyCurrent="35"/>
    <txSupplyCurrent txPower="14" supplyCurrent="44"/>
  </txSupplyCurrents>
</root>

```

Anexo 5. Código fuente cloudDelays.xml

```

<internetCloud symmetric="true">
  <parameters name="good">
    <traffic src="*" dest="*" delay="10ms" datarate="1Gbps" drop="uniform(0,1)
    &lt; 0" />
  </parameters>
</internetCloud>

```

Anexo 6. Código de programación del Nodo 1

```
#include "LoRaWan_APP.h" //libreria lorawan
#include <Wire.h> //libreria para hacer una comunicacion i2c
#include "Arduino.h" //libreria arduino
#include <DHT.h> //libreria dht
#include <DHT_U.h>
#define DHTPIN GPIO1 // se define el pin de entrada
#define DHTTYPE DHT22 // se define la version del sensor
//=====
DHT dht(DHTPIN, DHTTYPE);
float val=0; // almacenar la lectura (val es 1 cuando detecta vibración)
int vibration = GPIO5; // variable para conectar la salida del sensor D0
//=====
/*Se define una estructura llamada TransmitData que almacenará los datos enviados
por
el esclavo (arduino uno) en cuatro valores flotantes relacionados.*/
struct TransmitData
{
    float ANE1; // datos del anenometro
    float EJEX; // dato del acelerometro eje x
    float EJEY; // dato del acelerometro eje y
    float EJEZ; // dato del acelerometro eje z
};
// Se declara una variable global llamada 'data' de tipo TransmitData, que
almacenará los datos
TransmitData data;
// Declaración de variables de configuración para la red LoRaWAN
//-----Modo de activacion-----
// OTAA (Over-The-Air Activation)
uint8_t devEui[] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0xF7, 0xA4 };
uint8_t appEui[] = { 0xA5, 0x69, 0xB7, 0xA9, 0xB2, 0xF8, 0xA3, 0x8B };
uint8_t appKey[] = { 0x70, 0x9D, 0xB0, 0x4E, 0x73, 0x0E, 0x59, 0x09, 0xFD, 0x27,
0x72, 0x98, 0x96, 0xEF, 0x70, 0x35 };
// ABP (Activation By Personalization)
uint8_t nwkSKey[] = { 0xB4, 0x8F, 0xEA, 0xA1, 0x0E, 0xFD, 0x39, 0x18, 0x0A, 0x65,
0x51, 0x5D, 0xC8, 0x00, 0xF5, 0x15 };//
uint8_t appSKey[] = { 0xCD, 0x67, 0x4F, 0xD3, 0x95, 0x03, 0xEC, 0xB7, 0xAA, 0x43,
0xB0, 0xAC, 0x35, 0xC1, 0xCF, 0x35 };//
uint32_t devAddr = ( uint32_t )0x260C9bba;

/*Máscara de canales LoraWan, canales por defecto 0-7*/
uint16_t userChannelsMask[6]={ 0xFF00,0x0000,0x0000,0x0000,0x0000,0x0000 };

/*Región LoraWan, seleccionar en las herramientas IDE de arduino*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;

/*Se admiten la Clase LoraWan, la Clase A y la Clase C*/
DeviceClass_t loraWanClass = LORAWAN_CLASS;

/*ciclo de trabajo de transmisión de datos de la aplicación. valor en [ms].*/
uint32_t appTxDutyCycle = 25000; // 25 segundos

/*OTAA o ABP*/
bool overTheAirActivation = LORAWAN_NETMODE;

/*Habilitar Adaptive Data Rate (ADR)*/
bool loraWanAdr = LORAWAN_ADR;
```

```

/*Poner LORAWAN_Net_Reserve en ON, el nodo puede guardar la información de red en
flash, cuando el nodo se reinicia no necesita unirse de nuevo */
bool keepNet = LORAWAN_NET_RESERVE;

/*Indica si el nodo está enviando mensajes confirmados o sin confirmar */
bool isTxConfirmed = LORAWAN_UPLINKMODE;

/*Puerto de aplicación*/
uint8_t appPort = 2;

uint8_t confirmedNbTrials = 4; // Número de retransmisiones en caso de no recibir
una confirmación

/* Prepara la carga útil de la trama (payload)*/
static void prepareTxFrame( uint8_t port )
{
    //=====Lecturas de los sensores=====
    int val = digitalRead (vibration); // leer valor del sensor SW-420
    float temperature = dht.readTemperature(); //leer valor del DHT22
    float int_temp = temperature ; // asignar el valor de temperatura
    float int_vel = data.ANE1 ; // asignar valor del anemometro
    int vibra = val*10; // asignar valor de las vibraciones
    float int_X= data.EJEX ;// asignar valor del acelerometro eje x
    float int_Y= data.EJEY ;// asignar valor del acelerometro eje y
    float int_Z= data.EJEZ;// asignar valor del acelerometro eje z
    //=====ENVIO DE DATOS=====
    /*signed char se utiliza para representar valores de -128 a 127 sea float o int
    Se asigna a puc la dirección de memoria donde se encuentra almacenada la
    variable */
    signed char *puc;
    puc = (signed char *)&int_temp;
    appDataSize = 22;
    appData[0] = puc[0];
    appData[1] = puc[1];
    appData[2] = puc[2];
    appData[3] = puc[3];
    puc = (signed char *)&int_vel;
    appData[4] = puc[0];
    appData[5] = puc[1];
    appData[6] = puc[2];
    appData[7] = puc[3];
    puc = (signed char *)&int_X;
    appData[8] = puc[0];
    appData[9] = puc[1];
    appData[10] = puc[2];
    appData[11] = puc[3];
    puc = (signed char *)&int_Y;
    appData[12] = puc[0];
    appData[13] = puc[1];
    appData[14] = puc[2];
    appData[15] = puc[3];
    puc = (signed char *)&int_Z;
    appData[16] = puc[0];
    appData[17] = puc[1];
    appData[18] = puc[2];
    appData[19] = puc[3];

    appData[20] = vibra >> 8;

```

```

    appData[21] = vibra;
}
void setup() {
    Wire.begin(); // Inicializa la comunicación I2C
    dht.begin(); // Inicializa el sensor DHT (DHT22)
    Serial.begin(115200);
    pinMode (vibration, INPUT); // sensor como entrada (op)(SW-420)
#ifdef AT_SUPPORT //habilitar comandos AT
    enableAt();
#endif
    deviceState = DEVICE_STATE_INIT;
    LoRaWAN.ifskipjoin();
}

void loop()
{
    // Se utiliza la función Wire.requestFrom para solicitar datos desde el
    dispositivo con dirección 2 (esclavo).
    // sizeof data devuelve el tamaño en bytes de la estructura 'data'.
    if (Wire.requestFrom(2, sizeof data)) {
        // Si la solicitud fue exitosa y se recibieron los datos solicitados:
        // Se utiliza la función Wire.readBytes para leer los datos recibidos desde el
        dispositivo y almacenarlos en la variable 'data'.
        // (byte*) &data se utiliza para convertir el puntero de la estructura 'data'
        en un puntero de bytes para leer los datos en el formato adecuado.
        // sizeof data devuelve nuevamente el tamaño en bytes de la estructura 'data',
        lo que indica la cantidad de datos que se deben leer desde el bus I2C.
        Wire.readBytes((byte*) &data, sizeof data);
    }

    switch( deviceState )
    {
        case DEVICE_STATE_INIT:
        {
#ifdef LORAWAN_DEVEUI_AUTO // Generar el DevEUI automáticamente a partir del ID del
chip si está habilitado
            LoRaWAN.generateDeveuiByChipID();
#endif
#ifdef AT_SUPPORT
            getDevParam();
#endif
            printDevParam();
            LoRaWAN.init(loraWanClass,loraWanRegion); // Inicializar la
librería LoRaWAN
            deviceState = DEVICE_STATE_JOIN;
            break;
        }
        case DEVICE_STATE_JOIN: // Unirse a la red LoRaWAN
        {
            LoRaWAN.join();
            break;
        }
        case DEVICE_STATE_SEND: // Preparar y enviar el mensaje.
        {
            prepareTxFrame( appPort );
            LoRaWAN.send();
            deviceState = DEVICE_STATE_CYCLE; //En este caso, se programa
la transmisión del siguiente paquete estableciendo el tiempo de espera
            break;
        }
    }
}

```

```

    }
    case DEVICE_STATE_CYCLE:
    {
        // Programar la transmisión del siguiente paquete
        txDutyCycleTime = appTxDutyCycle + randr( 0,
APP_TX_DUTYCYCLE_RND );
        LoRaWAN.cycle(txDutyCycleTime);
        deviceState = DEVICE_STATE_SLEEP;
        break;
    }
    case DEVICE_STATE_SLEEP:
    {
        LoRaWAN.sleep();// Poner el dispositivo en modo de suspensión
        break;
    }
    default:
    {
        deviceState = DEVICE_STATE_INIT; //reiniciar el ciclo.
        break;
    }
}
delay(1000);
}

```

Anexo 7. Código de programación del esclavo

```
//-----Esclavo-----
//Esclavo (Arduino Uno), Maestro(Cubecell)
#include <Adafruit_ADXL345_U.h> // se incluye las librerías del acelerómetro
ADXL345
#include <Adafruit_Sensor.h>
#include <Arduino.h>
#include <Wire.h> // Se declara la librería estándar de arduino le
permite comunicarse con dispositivos por bus I2C
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);
/* Variables para el envío de los datos mediante I2C*/
//-----
#define INTERVAL_COUNTDOWN 1000 // se define el intervalo de 1 segundo de cuenta
regresiva
#define INTERVAL_READ 30000

// Se declara una estructura llamada TransmitData para almacenar los datos a
enviar mediante I2C
struct TransmitData {
    float ANE1;
    float EJEX;
    float EJEY;
    float EJEZ;
};

unsigned long timer = 0;

TransmitData data;

void setup() {

    Serial.begin(115200); //iniciar comunicación serial
    // Se verifica si el sensor ADXL345 está disponible y funcional
    if(!accel.begin()) //iniciar lecturas del acelerómetro
    {
        Serial.println("No se pudo encontrar el sensor ADXL345");
        while(1);
    }
    // Configurar el rango de medición del acelerómetro a ±4g (gravidades)
    accel.setRange(ADXL345_RANGE_4_G); // Configurar el rango de ±4g
(ADXL345_RANGE_2_G, ADXL345_RANGE_4_G, ADXL345_RANGE_8_G y ADXL345_RANGE_16_G)
    // Se inicia la comunicación I2C en el pin 2 del Arduino
    Wire.begin(2);
    // Se establece una función (requestEvent) que será llamada cuando otro
dispositivo realice una solicitud I2C a este Arduino
    Wire.onRequest(requestEvent);
}

void loop() {
    // Se realiza una lectura analógica del pin A0 para obtener el valor medido
(voltaje) del viento
    float valorMedido = analogRead(A0);
    // Se convierte el valor medido de voltaje a velocidad del viento en Km/h
    // ahora convertimos la lectura analógica (que va de 0 a 1023) a voltaje (0 -
5V):
    float voltaje = valorMedido * (5.0/1023.0);
    float velocidad= voltaje*25;
```

```

float velocidad1=velocidad*(3600/1000);//convertimos unidades de ms a Km/H
// Se imprime la velocidad del viento en Km/h en el monitor serial
Serial.print("Velocidad del viento Km/h: ");
Serial.print(velocidad1);
Serial.println();
// Se obtienen las lecturas de aceleración en cada eje del acelerómetro (en
unidades g, donde 1g = 9.81 m/s^2)
sensors_event_t event;
accel.getEvent(&event);
Serial.print("Aceleración (g): ");
Serial.print(event.acceleration.x/ 9.81);
Serial.print(", ");
Serial.print(event.acceleration.y/ 9.81);
Serial.print(", ");
Serial.println(event.acceleration.z/ 9.81);
// Se obtienen las lecturas de aceleración en cada eje en g (gravedades,
1g=9.8m/s^2) y se almacenan en variables 'x', 'y' y 'z'
float x= event.acceleration.x / 9.81;
float y= event.acceleration.y / 9.81;
float z= event.acceleration.z / 9.81;

// Se actualizan los valores de la estructura TransmitData con los datos
actuales de velocidad del viento y aceleración
data.ANE1 = velocidad1;
data.EJEX = x;
data.EJEY = y;
data.EJEZ = z;
// Se establece un retraso de 2 segundos antes de repetir el ciclo
delay(4000);
}
void requestEvent() {
// Se envían los datos almacenados en la estructura 'data' al dispositivo que
hizo la solicitud I2C
Wire.write((byte *)&data, sizeof data);
delay(5000); // Se agrega un pequeño retraso para permitir una transmisión
adecuada de los datos
}

```

Anexo 8. Código de programación del nodo 2

```
#include "LoRaWan_APP.h" //libreria lorawan
#include "Arduino.h" //libreria arduino
#include <DHT.h> //libreria dht
#include <DHT_U.h>
#define DHTPIN GPIO1 // se define el pin de entrada
#define DHTTYPE DHT22 // se define la version del sensor DHT (DHT11, DHT11)
DHT dht(DHTPIN, DHTTYPE);
int vibration = GPIO5; // variable para conectar la salida del sensor D0
//=====
//nodo 2
/*
 * set LoraWan_RGB to Active,the RGB active in loraWan
 * RGB rojo significa enviando
 * RGB púrpura significa unido hecho;
 * RGB azul significa RxWindow1;
 * RGB amarillo significa RxVentana2;
 * RGB verde significa recibido hecho;
 */
// Declaración de variables de configuración para la red LoRaWAN
//-----Modo de activacion-----
// OTAA (Over-The-Air Activation)
uint8_t devEui[] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0xF7, 0x9F };
uint8_t appEui[] = { 0x15, 0xDE, 0x20, 0x8A, 0xB6, 0x34, 0xA6, 0xA1 };
uint8_t appKey[] = { 0x5A, 0x38, 0xBD, 0x0A, 0x3F, 0x24, 0x84, 0xF8, 0x03, 0x5A,
0xA0, 0x7E, 0x03, 0xC4, 0xF5, 0x22 };

// ABP (Activation By Personalization)
uint8_t nwkSKey[] = { 0x37, 0x73, 0x73, 0x94, 0xE2, 0xC1, 0xDF, 0xC1, 0xED, 0x7C,
0xC7, 0xEB, 0xF4, 0x22, 0x2B, 0x9F };
uint8_t appSKey[] = { 0xF7, 0x13, 0xA5, 0xB4, 0x62, 0x0F, 0x7A, 0x42, 0xCA, 0xE5,
0xBA, 0x05, 0xC5, 0xC0, 0x55, 0x67 };
uint32_t devAddr = ( uint32_t )0x260C9505;

/*Máscara de canales LoraWan, canales por defecto 0-7*/
uint16_t userChannelsMask[6]={ 0xFF00,0x0000,0x0000,0x0000,0x0000,0x0000 };

/*Región LoraWan, seleccionar en las herramientas IDE de arduino*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;

/*Se admiten la Clase LoraWan, la Clase A y la Clase C*/
DeviceClass_t loraWanClass = LORAWAN_CLASS;

/*ciclo de trabajo de transmisión de datos de la aplicación. valor en [ms].*/
uint32_t appTxDutyCycle = 20000; // 20 segundos

/*OTAA o ABP*/
bool overTheAirActivation = LORAWAN_NETMODE;

/*Habilitar Adaptive Data Rate (ADR)*/
bool loraWanAdr = LORAWAN_ADR;

/*ADR mecanismo para optimizar las velocidades de datos, el tiempo aire y el
consumo de energía en la red.
-SF
-BW
```

```

-PTx*/

/*Poner LORAWAN_Net_Reserve en ON, el nodo puede guardar la información de red en
flash, cuando el nodo se reinicia no necesita unirse de nuevo */
bool keepNet = LORAWAN_NET_RESERVE;

/*Indica si el nodo está enviando mensajes confirmados o sin confirmar */
bool isTxConfirmed = LORAWAN_UPLINKMODE;

/*Puerto de aplicación*/
uint8_t appPort = 2;
/*!
* Número de intentos para transmitir la trama, si la capa LoRaMAC no
* recibido un acuse de recibo. La MAC realiza una adaptación de la tasa de datos,
* de acuerdo con la Especificación LoRaWAN V1.0.2, capítulo 18.4, según
* la tabla siguiente:
*
* Transmission nb | Data Rate
* -----|-----
* 1 (first)      | DR
* 2              | DR
* 3              | max(DR-1,0)
* 4              | max(DR-1,0)
* 5              | max(DR-2,0)
* 6              | max(DR-2,0)
* 7              | max(DR-3,0)
* 8              | max(DR-3,0)
*
* Tenga en cuenta que si NbTrials se establece en 1 o 2, la MAC no disminuirá
* el datarate, en caso de que la capa LoRaMAC no recibiera un acuse de recibo
*/
uint8_t confirmedNbTrials = 4; // Número de retransmisiones en caso de no recibir
una confirmación

/* Prepara la carga útil de la trama (payload)*/
static void prepareTxFrame( uint8_t port )
{
  /*appData size es LORAWAN_APP_DATA_MAX_SIZE que se define en "commissioning.h".
  *appDataSize valor máximo es LORAWAN_APP_DATA_MAX_SIZE.
  *Si se activa AT, no modifique LORAWAN_APP_DATA_MAX_SIZE, puede causar que el
  sistema se cuelgue o falle.
  *Si se desactiva AT, LORAWAN_APP_DATA_MAX_SIZE puede modificarse, el valor
  máximo hace referencia a la región lorawan y SF.
  *Por ejemplo, si se utiliza REGION_CN470,
  *el valor máximo para diferentes DR se puede encontrar en
  MaxPayloadOfDatarateCN470 consulte DataratesCN470 y BandwidthsCN470 en
  "RegionCN470.h".
  */
  //=====Lectura de los sensores=====
  int val = digitalRead (vibration); // leer valor del SW-420
  //*****
  float temperature = dht.readTemperature(); //lectura de temperatura
  float humidity = dht.readHumidity(); //lectura de humedad
  float hic= dht.computeHeatIndex (temperature,humidity,false); //indice de calor
  //-----Impresion en la consola serial-----
  //=====SW-420=====
  Serial.print("Vibracion= ");
  Serial.println(val);
  //=====DHT22=====

```

```

Serial.print("temperatura= ");
Serial.println(temperature);
Serial.print("humedad= ");
Serial.println(humidity);
Serial.print("Indice de calor= ");
Serial.println(hic);

//=====
/*int int_temp = 20.30 * 100; // Remover coma
int int_hum = 60.30 * 10; // Remover coma
int int_hic = 22.12*100;
int int_vel = 12.3 * 10; // Remover coma
*/
int int_temp = temperature * 100; // Remover coma
int int_hum = humidity * 100; // Remover coma
int int_hic = hic*100; //remover coma (conversion float a int)
int vibra = val*10; //val

// Asignación de los valores al appData
appDataSize = 8; // cantidad de bytes (byte=8bits)
appData[0] = int_temp >> 8;
appData[1] = int_temp;

appData[2] = int_hum >> 8;
appData[3] = int_hum;

appData[4] = int_hic >> 8;
appData[5] = int_hic;

appData[6] = vibra >> 8;
appData[7] = vibra;

/*
-appData[0] = int_temp >> 8;
desplazar hacia la derecha (>>) la variable int_temp en 8 bits al elemento en
la posición 0 del arreglo appData.
El desplazamiento hacia la derecha en 8 bits divide int_temp entre 256 (2^8) y
toma la parte entera del resultado.
Esto se utiliza para extraer los bits más significativos de int_temp y asignar
ese valor a appData[0]
Ejemp, si int_temp es igual a 512, después de esta operación, appData[0] tendrá
el valor 2 (512/256=2).
-appData[1] = int_temp;: Esta línea asigna el valor actual de int_temp al
elemento en la posición 1 del arreglo
*/

// Impresión de los valores en la consola serial
Serial.println(appData[0],HEX);
Serial.println(appData[1], HEX);
Serial.println(appData[2], HEX);
Serial.println(appData[3], HEX);
Serial.println(appData[4],HEX);
Serial.println(appData[5], HEX);
Serial.println(appData[6], HEX);
Serial.println(appData[7], HEX);
}

```

```

void setup() {
  Serial.begin(115200);
  dht.begin();
  pinMode (vibration, INPUT); // sensor como entrada (op)(SW-420)
#if(AT_SUPPORT) //habilitar comandos AT
  enableAt();
#endif
  deviceState = DEVICE_STATE_INIT;
  LoRaWAN.ifskipjoin();
}

void loop()
{
  switch( deviceState )
  {
    case DEVICE_STATE_INIT:
    {
      #if(LORAWAN_DEVEUI_AUTO) // Generar el DevEUI automáticamente a partir del ID del
      chip si está habilitado
        LoRaWAN.generateDeveuiByChipID();
      #endif
      #if(AT_SUPPORT)
        getDevParam();
      #endif
      printDevParam();
      LoRaWAN.init(loraWanClass,loraWanRegion); // Inicializar la librería
      LoRaWAN con la clase y región configuradas
      deviceState = DEVICE_STATE_JOIN;
      break;
    }
    case DEVICE_STATE_JOIN: // Unirse a la red LoRaWAN
    {
      LoRaWAN.join();
      break;
    }
    case DEVICE_STATE_SEND: // Preparar y enviar el mensaje.
    {
      prepareTxFrame( appPort );
      LoRaWAN.send();
      deviceState = DEVICE_STATE_CYCLE; //En este caso, se programa la
      transmisión del siguiente paquete estableciendo el tiempo de espera
      break;
    }
    case DEVICE_STATE_CYCLE:
    {
      // Programar la transmisión del siguiente paquete
      txDutyCycleTime = appTxDutyCycle + randr( 0, APP_TX_DUTYCYCLE_RND ); //
      intervalo de transmisión periódica de un dispositivo LoRa
      LoRaWAN.cycle(txDutyCycleTime);
      deviceState = DEVICE_STATE_SLEEP;
      break;
    }
    case DEVICE_STATE_SLEEP:
    {
      LoRaWAN.sleep(); // Poner el dispositivo en modo de suspensión
      break;}
    default:{
      deviceState = DEVICE_STATE_INIT; //reiniciar el ciclo.
      break; }}}

```

Anexo 9. Código de programación del decodificador del Nodo 1

```
// La función Decoder toma dos argumentos: 'bytes' (un array de bytes) y 'port'
// (un número de puerto).
function Decoder(bytes, port) {
// Se extrae un valor entero de 16 bits de los bytes en las posiciones 20 y 21
//y se almacena en la variable 'varv'.
    var varv = (bytes[20] << 8) | bytes[21];

// Se define la función 'bytesToFloat' que toma como argumento un array de
//bytes y devuelve un valor de punto flotante.
    function bytesToFloat(bytes) {
// Se convierten los bytes en un entero de 32 bits utilizando operaciones de
//desplazamiento y operaciones OR.
        var bits = bytes[3] << 24 | bytes[2] << 16 | bytes[1] << 8 | bytes[0];
// Se verifica el bit de signo para determinar si el valor es positivo o
//negativo.
        var sign = (bits >>> 31 === 0) ? 1.0 : -1.0;
// Se extrae el exponente y el mantisa del valor de 32 bits.
        var e = bits >>> 23 & 0xff;
        var m = (e === 0) ? (bits & 0x7fffffff) << 1 : (bits & 0x7fffffff) | 0x800000;
// Se calcula el valor de punto flotante utilizando la fórmula (sign * mantisa
//* 2^(exponente - 150)).
        var f = sign * m * Math.pow(2, e - 150);
        // Se devuelve el valor de punto flotante resultante.
        return f;
    }

// Se devuelve un objeto que contiene los resultados decodificados.
return {
// Se decodifica la temperatura tomando los primeros 4 bytes y se almacena
//en la propiedad 'temperatura'.
    temperatura: bytesToFloat(bytes.slice(0, 4)),
// Se decodifica la velocidad del viento tomando los siguientes 4 bytes y
//se almacena en la propiedad 'velocidadViento'.
    velocidadViento: bytesToFloat(bytes.slice(4, 8)),
// Se decodifica la aceleración en el eje X tomando los siguientes 4 bytes
//y se almacena en la propiedad 'Acel_EJE_X'.
    Acel_EJE_X: bytesToFloat(bytes.slice(8, 12)),
// Se decodifica la aceleración en el eje Y tomando los siguientes 4 bytes
//y se almacena en la propiedad 'Acel_EJE_Y'.
    Acel_EJE_Y: bytesToFloat(bytes.slice(12, 16)),
// Se decodifica la aceleración en el eje Z tomando los siguientes 4 bytes
//y se almacena en la propiedad 'Acel_EJE_Z'.
    Acel_EJE_Z: bytesToFloat(bytes.slice(16, 20)),
// Se utiliza el valor 'varv' (valor entero de 16 bits extraído al
principio) y se multiplica por 0.1 para obtener las vibraciones, que se
almacenan en la propiedad 'Vibraciones'.
    Vibraciones: varv * 0.1
};
}
```

Anexo 10. *Código de programación del decodificador del Nodo 2*

```
// Esta función toma un array de bytes y un número de puerto como parámetros
function Decoder(bytes, port) {
  // Se combinan los dos primeros bytes (bytes[0] y bytes[1]) para obtener el
  valor
  var var1 = (bytes[0] << 8) | bytes[1];
  // Se combinan los siguientes dos bytes (bytes[2] y bytes[3]) para obtener el
  valor
  var var2 = (bytes[2] << 8) | bytes[3];
  // Se combinan los siguientes dos bytes (bytes[4] y bytes[5]) para obtener el
  valor
  var var3 = (bytes[4] << 8) | bytes[5];
  // Se combinan los últimos dos bytes (bytes[6] y bytes[7]) para obtener el
  valor
  var var4 = (bytes[6] << 8) | bytes[7];
  // Se devuelve un objeto que contiene las variables calculadas con sus
  nombres
  return {
    // El valor de "temperatura" es igual a var1 multiplicado por 0.01
    temperatura: var1 * 0.01,
    // El valor de "humedad" es igual a var2 multiplicado por 0.01
    humedad: var2 * 0.01,
    // El valor de "IndiceCalor" es igual a var3 multiplicado por 0.01
    IndiceCalor: var3 * 0.01,
    // El valor de "vibracion" es igual a var4 multiplicado por 0.1
    vibracion: var4 * 0.1,
  };
}
```

Anexo 11. Datasheet SW-420

Specification

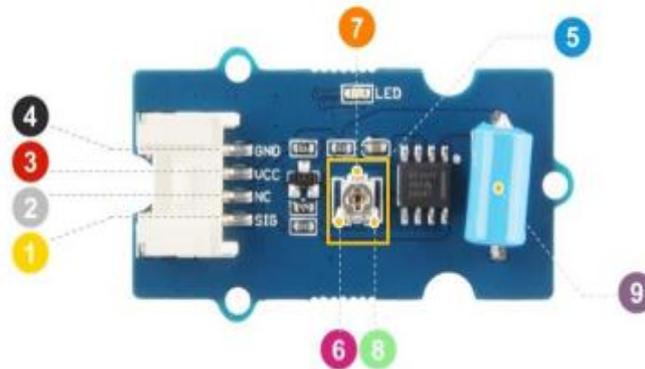
Item	Value
Operating voltage	3.3V / 5V
Interface	Digital

Applications

- Car, bicycle, motorcycle burglar alarm
- Game control
- Vibration detection

Hardware Overview

Pin Map



- ➊ GND: connect this module to the system GND
- ➋ VCC: you can use 5V or 3.3V for this module
- ➌ NC: none connected in this module
- ➍ SIG: output the Vout voltage signal
- ➎ Potentiometer: you can use a screwdriver to rotate the potentiometer which controls the sensitivity of this sensor.
- ➏ GND: the GND pin of the potentiometer.
- ➐ Vsen: the lower the voltage of Vsen, the higher the sensitivity
- ➑ VCC: the VCC pin of the potentiometer.
- ➒ SW-420: the Vibration Sensor

Preliminary Technical Data	ADXL345
----------------------------	----------------

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 2.5\text{ V}$, $V_{DDIO} = 1.8\text{ V}$, Acceleration = 0 g, unless otherwise noted.

Table 1. Specifications¹

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
	Each axis				
Measurement Range	User Selectable		±2, 4, 8, 16		g
Nonlinearity	Percentage of full scale		±0.5		%
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity ²			±1		%
OUTPUT RESOLUTION					
	Each axis				
All g-ranges	10-bit mode		10		Bits
±2 g range	Full-Resolution		10		Bits
±4 g range	Full-Resolution		11		Bits
±8 g range	Full-Resolution		12		Bits
±16 g range	Full-Resolution		13		Bits
SENSITIVITY					
	Each axis				
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 2.5\text{ V}$, ±2 g 10-bit or Full-Resolution	232	256	286	LSB/g
Scale Factor at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 2.5\text{ V}$, ±2 g 10-bit or Full-Resolution	3.5	3.9	4.3	mg/LSB
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 2.5\text{ V}$, ±4 g 10-bit mode	116	128	143	LSB/g
Scale Factor at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 2.5\text{ V}$, ±4 g 10-bit mode	7.0	7.8	8.6	mg/LSB
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 2.5\text{ V}$, ±8 g 10-bit mode	58	64	71	LSB/g
Scale Factor at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 2.5\text{ V}$, ±8 g 10-bit mode	14.0	15.6	17.2	mg/LSB
Sensitivity at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 2.5\text{ V}$, ±16 g 10-bit mode	29	32	36	LSB/g
Scale Factor at X_{OUT} , Y_{OUT} , Z_{OUT}	$V_S = 2.5\text{ V}$, ±16 g 10-bit mode	28.1	31.2	34.3	mg/LSB
Sensitivity Change due to Temperature			±0.02		%/°C
0 g BIAS LEVEL					
	Each axis				
0 g Output (X_{OUT} , Y_{OUT} , Z_{OUT})	$V_S = 2.5\text{ V}$, $T_A = 25^\circ\text{C}$	-150	0	+150	mg
0 g Offset vs. Temperature			<±1		mg/°C
NOISE PERFORMANCE					
Noise (x-, y-axes)	Data Rate = 100 Hz, ±2 g 10-bit or Full-Res.		<1		LSB RMS
Noise (z-axis)	Data Rate = 100 Hz, ±2 g 10-bit or Full-Res.		<1.5		LSB RMS
OUTPUT DATA RATE / BANDWIDTH					
	User Selectable				
Measurement Rate ³		0.1		3200	Hz
SELF TEST					
Output Change X		+0.31		+1.02	g
Output Change Y		-0.31		-1.02	g
Output Change Z		+0.46		+1.64	g
POWER SUPPLY					
Operating Voltage Range (V_S)		2.0	2.5	3.6	V
Interface Voltage Range (V_{DDIO})		1.7	1.8	V_S	V
Supply Current	Data Rate > 100 Hz		130	150	μA
Supply Current	Data Rate < 10 Hz		25		μA
Standby Mode Leakage Current			0.1	2	μA
Turn-On Time ⁴	Data Rate = 3200 Hz		1.4		ms
TEMPERATURE					
Operating Temperature Range		-40		85	°C
WEIGHT					
Device Weight			20		mgrams

¹ All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

² Cross-axis sensitivity is defined as coupling between any two axes.

³ Bandwidth is half the output data rate.

⁴ Turn-on and wake-up times are determined by the user defined bandwidth. At 100 Hz data rate the turn-on/wake-up time is approximately 11.1 ms. For additional data rates the turn-on/wake-up time is approximately $\tau + 1.1$ in milliseconds, where τ is $1/(\text{Data Rate})$.

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

1. Feature & Application:

- * Full range temperature compensated * Relative humidity and temperature measurement
- * Calibrated digital signal *Outstanding long-term stability *Extra components not needed
- * Long transmission distance * Low power consumption *4 pins packaged and fully interchangeable

2. Description:

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.

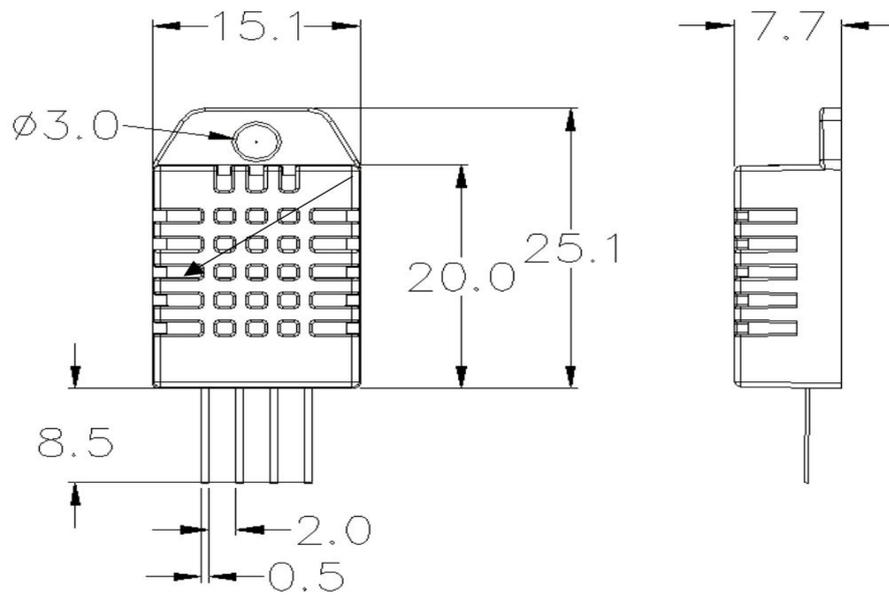
Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +2%RH(Max +5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+0.3%RH
Long-term Stability	+0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

4. Dimensions: (unit----mm)

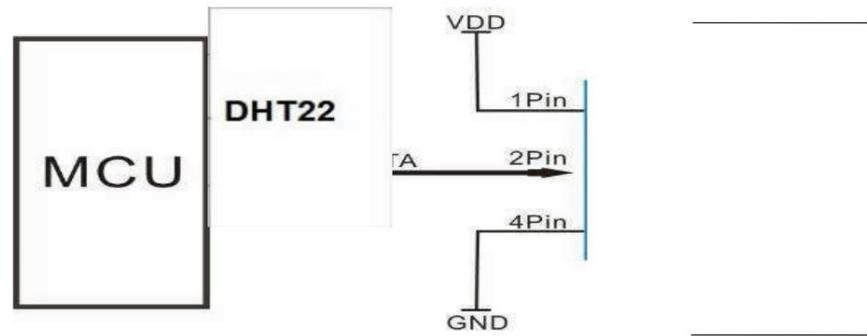
1) Small size dimensions: (unit----mm)



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD—power supply
2	DATA—signal
3	NULL
4	GND

5. Electrical connection diagram:



3Pin---NC, AM2302 is another name for DHT22

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-6V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

Single-bus data is used for communication between MCU and DHT22, it costs 5mS for single time communication.

Data is comprised of integral and decimal part, the following is the formula for data.

DHT22 send out higher data bit firstly!

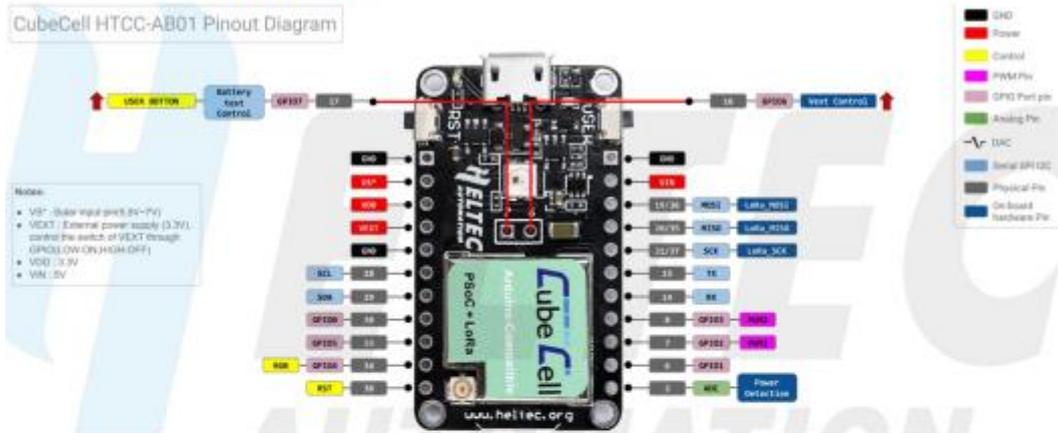
DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data+8 bit check-sum
If the data transmission is right, check-sum should be the last 8 bit of "8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data".

When MCU send start signal, DHT22 change from low-power-consumption-mode to running-mode. When MCU finishes sending the start signal, DHT22 will send response signal of 40-bit data that reflect the relative humidity

Anexo 14. Datasheet Heltec's CubeCell

2. Pin Definition

2.1 Pin assignment



2.2 Pin description

- Header J2

Table 2-2-1 Pin description

No.	Name	Type	Function
1	GND	P	Ground.
2	5V	P	5V Power Supply.
3	Ve	P	Output 3.3V, power supply for external sensor.
4	Ve	P	Output 3.3V, power supply for external sensor.
5	RX	I/O	GPIO44, U0RXD, connected to CP2102 TXD.
6	TX	I/O	GPIO43, U0RXD, connected to CP2102 RXD.
7	RST	I	CHIP_PU, connect to RST switch.
8	0	I/O	GPIO0, connect to PRG switch.
9	36	I/O	GPIO36, SPIIO7, FSPICLK, SUBSPICLK, Vext Ctrl.
10	35	I/O	GPIO35, SPIIO6, FSPID, SUBSPID, LED Write Ctrl.
11	34	I/O	GPIO34, SPIIO5, FSPICSO, SUBSPICSO.

12	33	I/O	GPIO33, SPIIO4, FSPiHD, SUBSPiHD.
13	47	I/O	GPIO47, SPiCLK_P_DIFF, SUBSPiCLK_P_DIFF.
14	48	I/O	GPIO48, SPiCLK_N_DIFF, SUBSPiCLK_N_DIFF.
15	26	I/O	GPIO26, SPiCS1.
16	21	I/O	GPIO21, OLED RST.
17	20	I/O	GPIO20, U1CTS, ADC2_CH9, CLK_OUT1, USB_D+ ¹ .
18	19	I/O	GPIO19, U1RTS, ADC2_CH8, CLK_OUT2, USB_D- ² .

Header J3

Table 2-2-2 Pin description

No.	Name	Type	Function
1	GND	P	Ground.
2	3V3	P	3.3V Power Supply.
3	3V3	P	3.3V Power Supply.
4	37	I/O	GPIO37, SPiDQS, FSPiQ, SUBSPiQ.
5	46	I/O	GPIO46.
6	45	I/O	GPIO45.
7	42	I/O	GPIO42, MTMS.
8	41	I/O	GPIO41, MTDI.
9	40	I/O	GPIO40, MTDO.
10	39	I/O	GPIO39, MTCK.
11	38	I/O	GPIO38, FSPiWP, SUBSPiWP.
12	1	I/O	GPIO1, ADC1_CH0 ³ , TOUCH1, Read VBAT Voltage.
13	2	I/O	GPIO2, ADC1_CH1, TOUCH2.
14	3	I/O	GPIO3, ADC1_CH2, TOUCH3.
15	4	I/O	GPIO4, ADC1_CH3, TOUCH4.
16	5	I/O	GPIO5, ADC1_CH4, TOUCH5.
17	6	I/O	GPIO6, ADC1_CH5, TOUCH6.

3.1 General specifications

Table 3-1: General specifications

Parameters	Description
Master Chip	ASR6501 (48 MHz ARM® Cortex® M0+ MCU)
LoRa Chipset	SX1262
USB to Serial Chip	CP2102
Frequency	470~510 MHz, 863~923 MHz
Max TX Power	22dB ± 1dB
Receiving sensitivity	-135 dBm
Solar Energy	5.5~7V solar panel
Low Power	Deep Sleep 3.5 μ A
Hardware Resource	UART x 1; SPI x 1; I2C x 1; SWD x 1; 12-bits ADC input x 1; 8 - channel DMA engine; GPIO x 8
Memory	128KB internal FLASH; 16KB internal SRAM
Interface	Micro USB x 1; LoRa Antenna interface(IPEX) x 1; 11 x 2.54 pin x 2 + 2 x 2.54 pin x 1
Battery	3.7V Lithium(SH1.25 x 2 socket)
Operating temperature	-20 ~ 70 ℃
Dimensions	41.5 x 25 x 7.6 mm

3.2 Power supply

Except when USB or 5V Pin is connected separately, lithium battery can be connected to charge it. In other cases, only a single power supply can be connected.

Table 3-2: Power supply

Power supply mode	Minimum	Typical	Maximum	Company
USB powered($\geq 500\text{mA}$)	4.7	5	6	V
Lithium battery($\geq 250\text{mA}$)	3.3	3.7	4.2	V
5V pin($\geq 500\text{mA}$)	4.7	5	6	V
3V3 pin($\geq 150\text{mA}$)	2.7	3.3	3.5	V

3.3 Power output

Table 3-3: Power output

Output Pin	Minimum	Typical	Maximum	Company
3.3V Pin			500	mA
5V Pin (USB Powered only)		Equal to the input current		
Vext Pin			350	mA

3.5.2 Receiving sensitivity

The following table gives typically sensitivity level of the HTCC-AB01-(L/H).

Table3-6: Receiving sensitivity

Signal Bandwidth/[KHz]	Spreading Factor	Sensitivity/[dBm]
125	SF12	-135
125	SF10	-130
125	SF7	-124

3.6 Operation Frequencies

HTCC-AB01(F) supports LoRaWAN frequency channels and models corresponding table.

Table3-7: Operation Frequencies

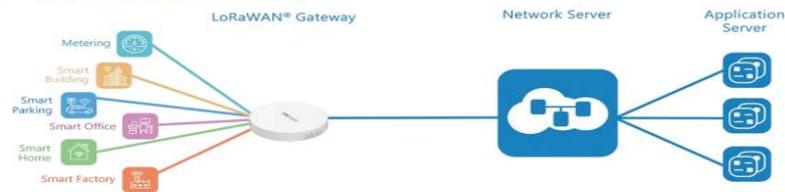
Region	Frequency (MHz)	Model
EU433	433.175~434.665	HTCC-AB01-L
CN470	470~510	HTCC-AB01-L
IN868	865~867	HTCC-AB01-F
EU868	863~870	HTCC-AB01-F
US915	902~928	HTCC-AB01-F
AU915	915~928	HTCC-AB01-F
KR920	920~923	HTCC-AB01-F
AS923	920~925	HTCC-AB01-F

Anexo 15. Datasheet Gateway UG63.



UG63 is a 8-channel lightweight indoor LoRaWAN® gateway. Adopting, SX1302 chip and high-performance CPU, UG63 can not only set up packet forwarding connection between more than 2000 end nodes and mainstream network servers (such as TTN, Chirpstack, etc.), but also cover up to 2 km in urbanized environment as a blind-filling gateway. With compact size, it is an ideal supplement for UG65 or UG67 for blind-filling in complex environment or wide indoor areas such as office, parking lots, campuses, etc. Besides, UG63 also has built-in network server and is compatible with Milesight IoT for dealing with data for a few end-nodes, suitable for LoRaWAN® small deployments.

◆ Application Example



LoRaWAN	
Antenna	2 × Internal Antennas
Channel	8 (Half/Full-duplex)
Frequency Band	CN470/IN865/EU868/RU864/US915/AU915/KR920/AS923-1&2&3&4
Sensitivity	-140dBm Sensitivity @292bps
Output Power	27dBm Max
Protocol	V1.0 Class A/Class B/Class C and V1.0.2 Class A/Class B/Class C
LBT	Support
Ethernet Interface	
Port	1 × RJ45 (PoE PD supported)
Physical Layer	10/100 Base-T (IEEE 802.3)
Data Rate	10/100 Mbps (Auto-Sensing)
Interface	Auto MDI/MDIX
Mode	Full or Half Duplex (Auto-Sensing)
Others	
Reset Button	1 × RST
Console Port	1 × Type-C
LED Indicators	1 × SYS, 1 × LoRa
Built-in	Watchdog, RTC, Timer
Software	
Network Protocols	PPPoE, SNMP v1/v2c/v3, TCP, UDP, DHCP, DDNS, HTTP, HTTPS, DNS, ARP, SNTP, Telnet, SSH, MQTT, etc.
VPN Tunnel	OpenVPN/IPsec/PPTP/L2TP/GRE/DMVPN
Management	Web, CLI, DeviceHub, Milesight IoT Cloud
Power Supply and Consumption	
Power Input	1. 1 × 802.3af standard PoE input 2. 5V by Type-C Port
Power Consumption	Max 3.3 W
Physical Characteristics	
Ingress Protection	IP30
Dimensions	Φ 115 x 21 mm (Φ 4.52 x 0.83 in)
Color & Material	White, PC + ABS

Anexo 16. Wind Sensor (Pulse -Type) JL -FS2.

**Wind Sensor (Pulse - Type)
Model: JL -FS2**



Features:

Wind sensor made of aluminum alloy material, the use of special precision die casting process , dimensional tolerances very small surface accuracy is very high, both through internal circuit protective treatment , the entire sensor with high strength, weather resistance , anti- corrosion and water resistance . Plug the cable connector for the military, has a good anti-corrosion, anti- corrosion properties, to ensure the long-term use of the instrument , in conjunction with internal bearings system to ensure the accuracy of the wind speed acquisition .

Circuit PCB using military grade A grade materials to ensure the quality and stability and electrical performance parameters ; electronic components are imported industrial-grade chips , making the whole has a very reliable anti-electromagnetic interference capability , to ensure that the host -20C ~ 60C, humidity can work properly within 10% -95% range.

Wind sensor compact , flange base, carrying , convenient installation , beautiful appearance , high accuracy , wide range, good stability , low power consumption , data of good, long distance signal transmission , anti- interference ability The signal output forms, light aluminum alloy material quality , high strength.

Wiring:

Three-wire:

Red LineSupply positive
Black line Power negative
Yellow Line Signal line

Specifications:

Type	Cable and outlet		Waterproof air plug type
	Model		
	Model		JL-FS2
	Name		Universal wind sensor
Sensor	Signal output		Pulse signal output
	Sensor Style		Three cups of formula
	Start wind speed		0. 4-0.8 m / s
	Resolution		0.1m / s
	Effective wind speed measurement range		0-30m / s or 0-60 m / s

General parameters	System error	± 3%
	Transmission distance	Greater than 1000m
	Transmission medium	Cable transmission
	Wiring	Three-wire
	Operating Temperature	-40 °C ~ 80 °C
	Supply Voltage	DC12-24V can be generic
	Power Consumption	MAX ≤ 0.2W
	Size	See size chart
	Weight	<1Kg

Dimensions:



Scope:

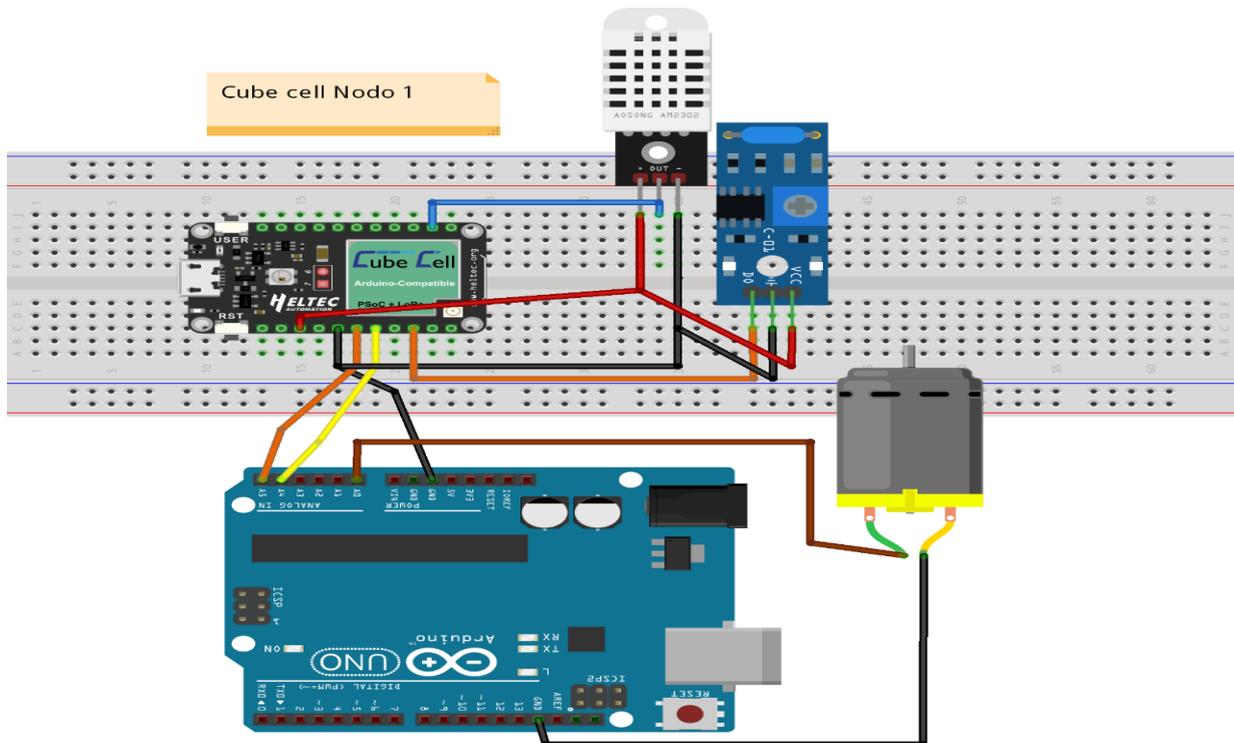
Can be widely used in greenhouses, environmental protection, weather stations, ships, docks, heavy machinery cranes, ports, terminals, cable cars, no need to measure wind speed and direction in place.

Made in China

Anexo 17. Diseño final para la placa base de los nodos finales.

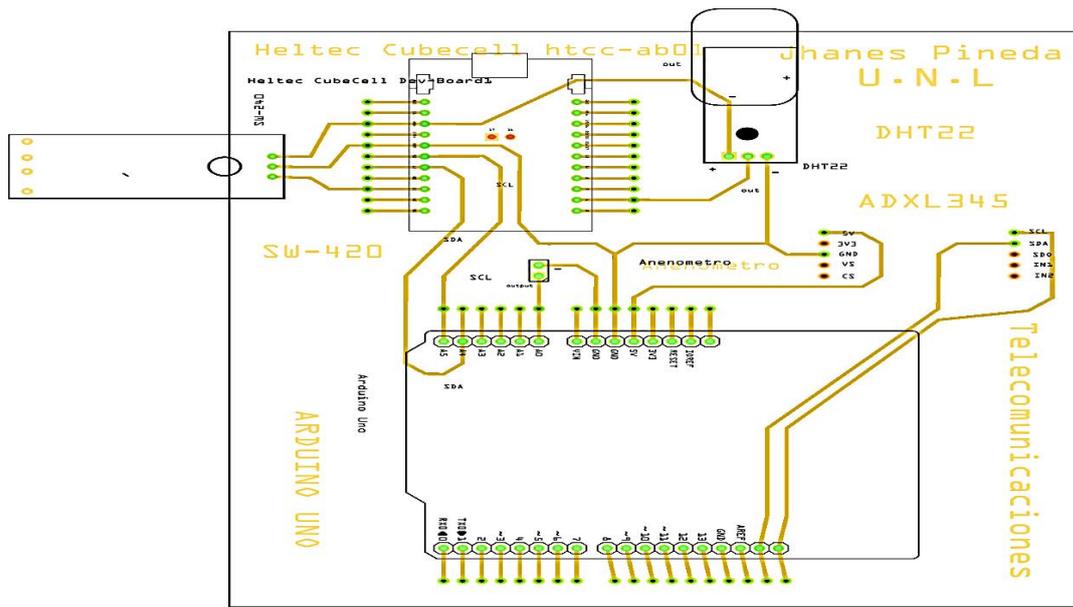
A) Primer nodo final.

Diseño del primer nodo en protoboard en el software Fritzing:



fritzing

Diseño de la placa PCB del segundo nodo en el software Fritzing:



fritzing

Primer nodo en caja protectora:

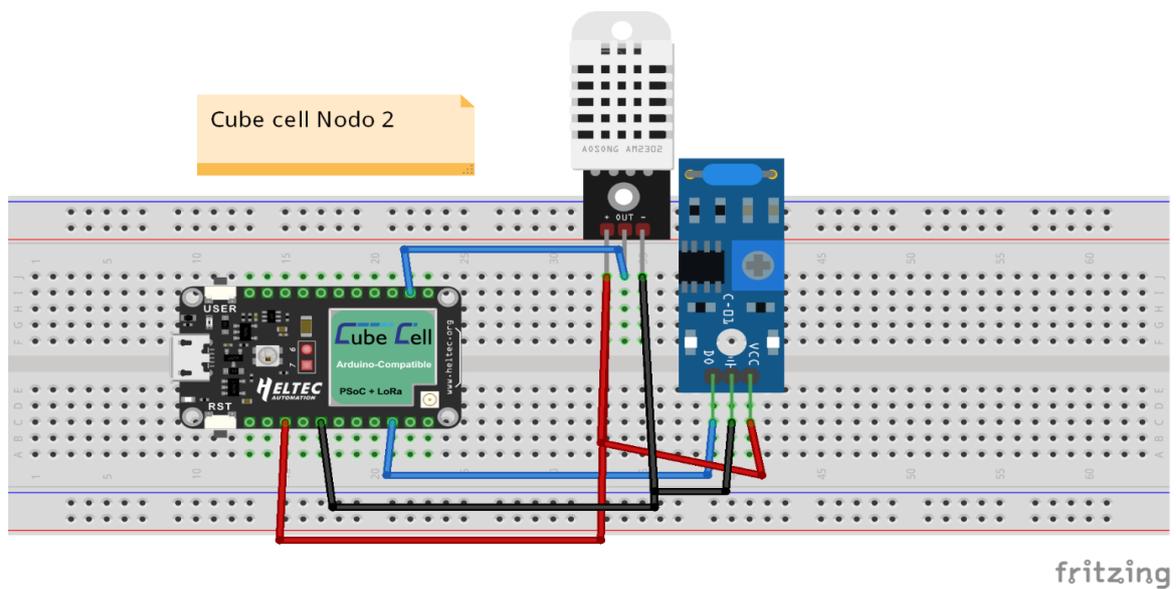


Anemómetro:

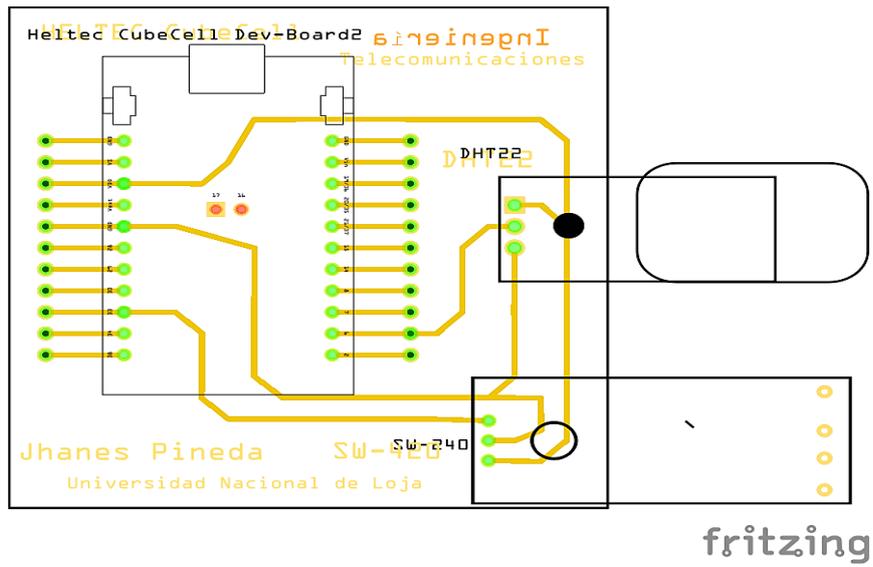


B) Segundo nodo final.

Diseño del segundo nodo en protoboard en el software Fritzing:



Diseño de la placa PCB del segundo nodo en el software Fritzing:



Segundo nodo en caja protectora:



Instalación del prototipo:



Colocación del nodo 1



Colocación del nodo 2



Vista al puente desde el Gateway

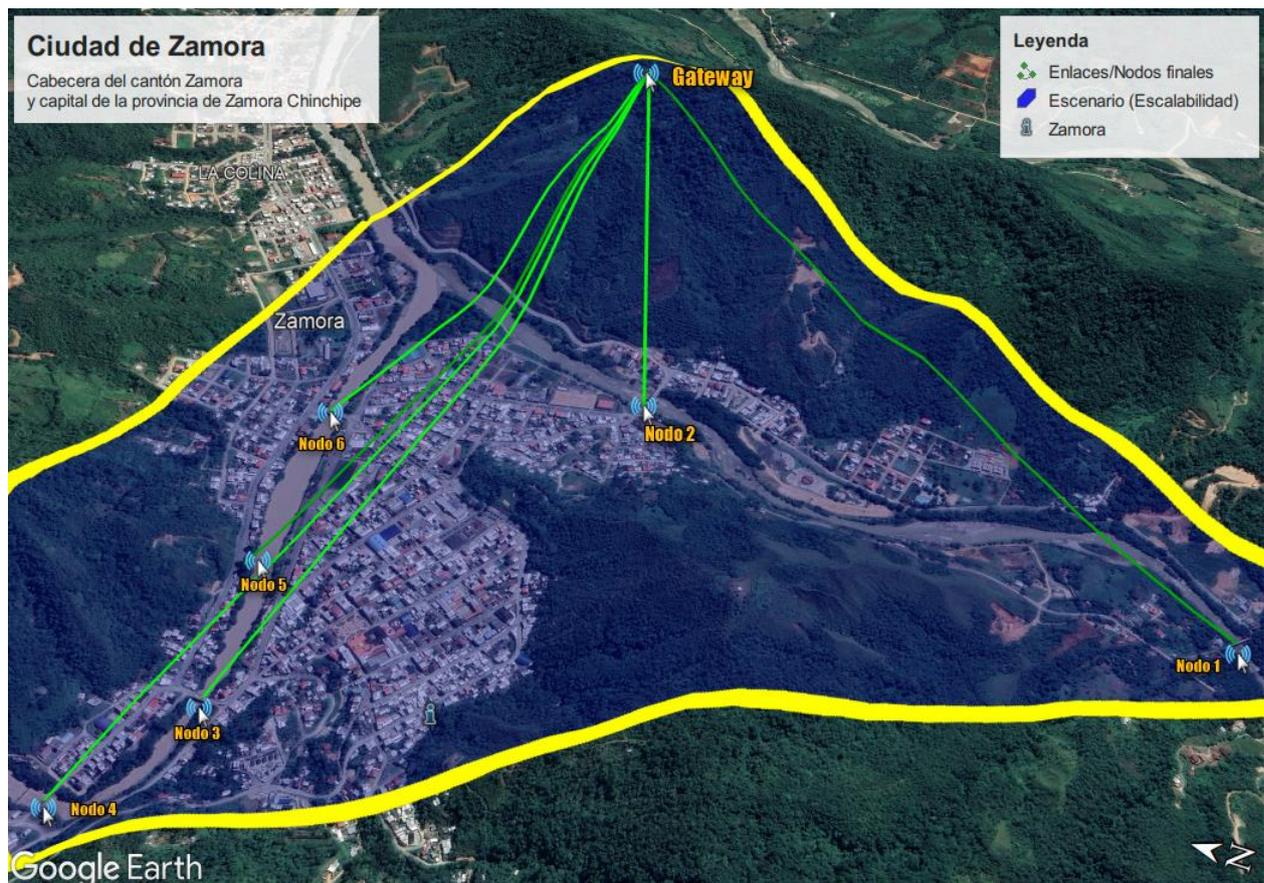


Colocación del anemómetro vista 1



Colocación del anemómetro vista 2

Anexo 18. Escalabilidad de la red WSN para el monitoreo estructural de puentes



Anexo 19. Certificado de traducción del Abstract

Loja, 25 de enero de 2024

Lic. Karina Yajaira Martínez Luzuriaga

LICENCIADA EN CIENCIAS DE LA EDUCACIÓN MENCIÓN INGLÉS

CERTIFICO:

Yo, Karina Yajaira Martínez Luzuriaga con cédula de identidad Nro. 1104902679, Licenciada en Ciencias de la Educación Mención Inglés por la Universidad Técnica Particular de Loja, con número de registro 1031-2022-2574017 en la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación, señalo que el presente documento es fiel traducción del idioma español al idioma inglés del resumen del Trabajo de Integración Curricular denominado "Diseño de una Red de Sensores Inalámbricos (WSN) de Bajo Consumo Energético para el Monitoreo Estructural de Puentes" elaborado por el Sr. Jhanes Alexander Pineda Chamba, con cédula de identidad Nro. 1900885888, estudiante egresado de la carrera de Ingeniería en Telecomunicaciones de la Universidad Nacional de Loja.



Lic. Karina Yajaira Martínez Luzuriaga

C.I. 1104902679

REGISTRO SENESCYT N°: 1031-2022-2574017