



Implementación un prototipo de una red lora con sensores para el control medioambiental en el laboratorio de sistemas de comunicación de la carrera de Tecnología Superior en Redes y Telecomunicaciones de la Universidad de las Fuerzas Armadas ESPE Latacunga.

Cajas Vilatuña, Cristóbal Eduardo y Segura Valdiviezo, Marco Antonio

Departamento de Eléctrica y Electrónica

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Trabajo de integración curricular previo a la obtención del título de Tecnólogo Superior en Redes y Telecomunicaciones

Ing. Moreta Changoluiza, Janneth Elizabeth

7 de febrero de 2023

Latacunga

Reporte de verificación de contenido

7/2/23, 11:54

Revisión tesis

Informe de originalidad

NOMBRE DEL CURSO

Revisión Tesis Cajas_Segura

NOMBRE DEL ALUMNO

MARCO ANTONIO SEGURA VALDIVIEZO



Ing. Janneth Moreta, Mgtr.

NOMBRE DEL ARCHIVO

MARCO ANTONIO SEGURA VALDIVIEZO - Documento sin título

Tutora trabajo de integración curricular

SE HA CREADO EL INFORME

7 feb 2023

Resumen

Fragmentos marcados	21	3 %
Fragmentos citados o entrecorridos	5	0,7 %

Coincidencias de la Web

electronicspot.com.co	4	0,6 %
tecnofoto.com	4	0,6 %
uelectronics.com	3	0,5 %
wordpress.com	3	0,3 %
epoch.edu.ec	2	0,3 %
amazon.es	1	0,2 %
agelectronica.lat	1	0,2 %
starware.com.ar	1	0,2 %
geekfactory.mx	1	0,2 %
hitechchain.se	1	0,1 %
scielo.org	1	0,1 %
aliexpress.com	1	0,1 %
2cigroup.com	1	0,1 %
ugr.es	1	0,1 %
cotecno.cl	1	0,1 %

1 de 26 fragmentos

Fragmento del alumno MARCADO

...desarrollo IoT diseñada y producida por Heltec Automation (TM). **Un producto altamente integrado basado en ESP32 + SX127x, tiene funciones Wifi, BLE, LoRa, sistema de administración de batería Li-Po**

Mejor coincidencia en la Web

WiFi LoRa-32 es una placa de desarrollo IoT clásica diseñada y producida por Heltec Automation(TM), es **un producto altamente integrado basado en ESP32 + SX127x, tiene funciones Wi-Fi, BLE, LoRa...**

tarjeta de desarrollo esp-32 wifi + lora +bluetooth. <https://agelectronica.lat/pdfs/textos/W/WIFI-LORA-32.PDF>

Ing. Moreta Changoluiza, Janneth Elizabeth

C.C: 050307897-4



Departamento de Eléctrica y Electrónica

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Certificación

Certifico que el trabajo de integración curricular, **“Implementación un prototipo de una red lora con sensores para el control medioambiental en el laboratorio de sistemas de comunicación de la Carrera de Tecnología Superior en Redes y Telecomunicaciones de la Universidad De Las Fuerzas Armadas ESPE Latacunga”** fue realizado por los señores **Cajas Vilatuña, Cristóbal Eduardo, y Segura Valdiviezo, Marco Antonio**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que lo sustenten públicamente.

Latacunga, 07 de febrero 2023

Ing. Moreta Changoluiza, Janneth Elizabeth

C.C: 050307897-4



Departamento de Eléctrica y Electrónica

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Responsabilidad de Auditoría

Nosotros **Cajas Vilatuña, Cristóbal Eduardo**, con cédula de ciudadanía n° 1728803808 y **Segura Valdiviezo, Marco Antonio**, con cédula de ciudadanía n° 2200105258 declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **Implementación un prototipo de una red lora con sensores para el control medioambiental en el laboratorio de sistemas de comunicación de la Carrera de Tecnología Superior en Redes y Telecomunicaciones De La Universidad De Las Fuerzas Armadas ESPE Latacunga**, es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 07 de febrero 2023

Cajas Vilatuña, Cristóbal Eduardo

C.C: 172880380-8

Segura Valdiviezo, Marco Antonio

C.C: 220010525-8



Departamento de Eléctrica y Electrónica

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Autorización de publicación

Nosotros **Cajas Vilatuña, Cristóbal Eduardo** con cédula de ciudadanía n°1728803808 y **Segura Valdiviezo, Marco Antonio**, con cédula de ciudadanía n°2200105258, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Implementación un prototipo de una red lora con sensores para el control medioambiental en el laboratorio de Sistemas de Comunicación de la Carrera de Tecnología Superior en Redes y Telecomunicaciones de la Universidad de las Fuerzas Armadas ESPE Latacunga**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 07 de febrero 2023

Cajas Vilatuña, Cristóbal Eduardo

C.C: 172880380-8

Segura Valdiviezo, Marco Antonio

C.C: 220010525-8

Dedicatoria

A Dios por otorgarme una familia maravillosa, los cuales me han apoyado, impulsado y educado, gracias a Dios por darme la fuerza y dones necesarios para poder superarme académicamente y poder completar triunfantemente mis estudios.

A mi madre por acompañarme e indicarme el camino correcto hacia la superación personal, académica y profesional, por brindarme educación, principios, y cualidades para alcanzar la meta de ser un profesional.

A mi tutora de trabajo de integración curricular, por los conocimientos impartidos, por la guía y tolerancia para el desarrollo del prototipo implementado, preparándome al camino del profesionalismo.

Cristóbal Eduardo Cajas Vilatuña

A la vida por permitirme vivir, aprender y disfrutar su generosidad cada día, por darme la fortaleza necesaria durante esta etapa de mi vida académica, logrando culminar con éxito.

A mis padres por estar presente en cada paso, decisión y camino que me he dispuesto emprender, inculcándome valores y educación para ser una mejor persona al igual que un gran profesional.

A mí tutora de trabajo de integración curricular, por brindarme el apoyo, conocimiento y en especial paciencia, conduciéndome hacía un camino de ética y profesionalismo.

Marco Antonio Seguro Valdiviezo

Agradecimiento

Agradezco a Dios por su gracia y bendición, y esencialmente a mi madre, por su amor, cariño, tolerancia, lucha durante mi etapa académica universitaria y como en mi vida profesional, gracias a ti he alcanzado llegar a cumplir muchos sueños y metas.

A la Universidad de las Fuerzas Armadas ESPE, por otorgarme la posibilidad de estudiar, por brindarme conocimientos y valores, agradezco de igual forma a todos los maestros dejándome como recuerdo la pasión a investigar. Gracias a su gran conocimiento al día de hoy contengo los conocimientos y técnicas para poder desenvolverme de manera profesional.

Finalmente agradezco muy encantado a mi tutora de trabajo de integración curricular por su ayuda y esfuerzo, ya que por su conocimiento, experiencia, eficacia y tolerancia me ayudado a cumplir la meta de obtener mi título universitario

Cristóbal Eduardo Cajas Vilatuña

En especial agradezco a mis padres por brindarme su apoyo incondicional, mismo el cual me ha vuelto el ser fuerte que soy capaz que enfrentar y resolver cada problema o dificultad que me presenta la vida.

Quiero agradecer a los docentes que me han acompañado a lo largo de mi desarrollo como profesional, quien han estado presente aportando sus valores y conocimientos que fueron de suma importancia para llevar a cabo este trabajo de integración curricular.

Marco Antonio Seguro Valdiviezo

ÍNDICE DE CONTENIDO

Carátula.....	1
Reporte de verificación de contenido.....	2
Certificación	3
Responsabilidad de Auditoría.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
Agradecimiento.....	8
Índice de contenido.....	10
Índice de figuras	14
Índice de tablas.....	17
Resumen.....	18
Abstract.....	19
Capítulo I: Introducción	20
Antecedentes.....	20
Planteamiento del problema.....	21
Justificación	22
Objetivos.....	22
<i>Objetivo general</i>	22

Objetivos específicos	22
Alcance	23
Capítulo II: Marco Teórico	24
Internet de las cosas (IoT)	24
Arquitectura del Internet de las cosas	24
Capa de aplicación.	25
Capa de soporte de servicios y aplicaciones	25
Capa de red IoT	25
Capa de dispositivo	25
Plataformas de Internet de las cosas	27
Arduino	28
Hletec Wifi LoRa 23 (v2)	28
LPWAN	31
Bandas sin licencia	32
Tecnología LoRa	32
Tecnología Sigfox	33
Protocolo LoRaWAN	34
Modulación LoRaWAN	35
Arquitectura LoRaWAN	36
Estructura de la trama física	37
Dispositivos que conforman la red LoRaWAN	39
Frecuencias empleadas para redes LoRaWAN	39
Seguridad	41
Clases de dispositivos LoRaWAN	41

Red de Sensores Inalámbricos (WSN)	42
<i>Características de una de red de sensores inalámbricos</i>	<i>42</i>
<i>Topologías para redes de sensores inalámbricas</i>	<i>44</i>
Nodos o motes.	45
<i>Topología Estrella.....</i>	<i>46</i>
<i>Sistema de red para monitoreo de variables medioambientales</i>	<i>47</i>
Sensores.....	48
<i>Clasificación de Sensores</i>	<i>49</i>
<i>Estructura de red para Sensores Inalámbricos</i>	<i>49</i>
Sensores para el monitoreo ambiental	50
<i>Sensor de Luminosidad.....</i>	<i>50</i>
<i>Sensor de medición de temperatura, humedad relativa y presión barométrica</i>	<i>51</i>
Capacidad del sensor BME280.	52
Propiedades de hardware del sensor BME280.	53
<i>Sensor de calidad del aire (CO₂).....</i>	<i>55</i>
Características	55
Capítulo III: Diseño e implementación de la red de sensores IoT	57
Diseño y construcción de la red LoRa.....	57
Diagrama de funcionamiento de la red.....	57
Emisor LoRa	58
<i>Asignación de pines de la placa</i>	<i>58</i>
<i>Configuración de la pantalla OLED.....</i>	<i>60</i>
<i>Configuración del sensor BME280.....</i>	<i>62</i>

<i>Configuración del sensor MG811</i>	64
<i>Configuraciones de lecturas</i>	65
<i>Módulo fotorresistencia LDR</i>	67
Receptor LoRa.....	68
<i>Organizar los archivos</i>	69
<i>Desarrollo HTML</i>	70
<i>Código de Arduino del receptor LoRa</i>	72
Análisis de resultados.....	80
Interpretación de variables.....	80
<i>Variables del sensor bme280</i>	81
<i>Variables del MG-811</i>	81
<i>Variables de la fotorresistencia</i>	82
Distancia Máxima	83
Capítulo IV: Conclusiones y recomendaciones	84
Conclusiones.....	84
Recomendaciones.....	86
Bibliografía	87
Anexos	91

ÍNDICE DE FIGURAS

Figura 1	<i>Arquitectura del Internet de las cosas (IoT)</i>	27
Figura 2	<i>Heltec WiFi LoRa 32</i>	31
Figura 3	<i>Tecnología LPWAN</i>	31
Figura 4	<i>Tecnología LoRa</i>	32
Figura 5	<i>Tecnología Sigfox</i>	33
Figura 6	<i>Estructura protocolo LoRaWAN</i>	35
Figura 7	<i>Modulación LoRaWAN</i>	36
Figura 8	<i>Arquitectura de red LoRaWAN</i>	37
Figura 9	<i>Trama física LoRaWAN</i>	38
Figura 10	<i>Estructura de red de sensores inalámbricos</i>	43
Figura 11	<i>Estructura del sensor inalámbrico</i>	44
Figura 12	<i>Heltec Wifi LoRa 32 V2</i>	45
Figura 13	<i>Topologías de la red de sensores inalámbrica</i>	46
Figura 14	<i>Topologías estrella</i>	47
Figura 15	<i>Sensor de luz LDR</i>	51
Figura 16	<i>Sensor BME280</i>	53
Figura 17	<i>Interfaz I2C</i>	54
Figura 18	<i>Pin Out BME280</i>	55
Figura 18	<i>Módulo sensor MG-811</i>	56
Figura 20	<i>Diagrama de funcionamiento de la red LoRa</i>	57
Figura 21	<i>Diagrama de distribución de Wifi LoRa 32</i>	59
Figura 22	<i>Líneas de código librerías para LoRa</i>	60
Figura 23	<i>Función inicialización LoRa</i>	60
Figura 24	<i>Configuración LoRa</i>	61

Figura 25	<i>Librerías para la pantalla OLED</i>	61
Figura 26	<i>Declaración de pines y pixeles pantalla OLED</i>	62
Figura 27	<i>Circuito LoRa emisor y BME280</i>	62
Figura 28	<i>Declaración de librerías BME280</i>	63
Figura 29	<i>Declaración de pines sensor MBE280</i>	63
Figura 30	<i>Función inicialización BME280</i>	63
Figura 31	<i>Circuito LoRa emisor y MG-811</i>	64
Figura 32	<i>Librerías para el sensor MG-811</i>	65
Figura 33	<i>Declaración de pines sensor MG-811</i>	65
Figura 34	<i>Declaración de variables de lectura</i>	65
Figura 35	<i>Crear bme</i>	65
Figura 36	<i>Inicialización monitor serie</i>	66
Figura 37	<i>Función obtener lecturas</i>	66
Figura 38	<i>Función enviar lecturas</i>	67
Figura 39	<i>Circuito Emisor LoRa-Fotorresistencia</i>	68
Figura 40	<i>Servidor web</i>	69
Figura 41	<i>Diagrama de archivos</i>	70
Figura 42	<i>Marcadores de posición</i>	70
Figura 43	<i>Agregar estilos</i>	71
Figura 44	<i>Código Java Scrip</i>	72
Figura 45	<i>Librerías receptor LoRa</i>	73
Figura 46	<i>Declaración de pines receptor LoRa</i>	73
Figura 47	<i>Declaración de frecuencia</i>	74
Figura 48	<i>Declaración de pines y pixeles para el receptor LoRa</i>	74
Figura 49	<i>Declaración de red local</i>	74

Figura 50	<i>Declaración de NTP Client</i>	75
Figura 51	<i>Variables para obtener y guardar datos</i>	75
Figura 52	<i>Proceso marcador de posición</i>	76
Figura 53	<i>Inicialización del monitor serie y SPIFFS</i>	76
Figura 54	<i>Configuración de rutas</i>	77
Figura 55	<i>Inicialización de pantalla OLED receptor LoRa</i>	78
Figura 56	<i>Función leer y obtener paquete LoRa</i>	78
Figura 57	<i>Declaración del protocolo NTP client</i>	79
Figura 58	<i>Inicialización del servidor web</i>	79
Figura 59	<i>Disposición de la red de sensores entre nodo-coordinador</i>	80
Figura 60	<i>Variables sensor MBE280 en el laboratorio de Sistemas de Comunicación</i>	81
Figura 61	<i>Variables sensor MBE280 en el Departamento</i>	81
Figura 62	<i>Variables sensor MBE280 en exteriores</i>	81
Figura 63	<i>Variable sensor MG-811 en el laboratorio de Sistemas de Comunicación</i>	82
Figura 64	<i>Variable sensor MG-811 en el Departamento</i>	82
Figura 65	<i>Variable sensor MG-811 en exteriores</i>	82
Figura 66	<i>Variable sensor fotorresistencia LDR en el laboratorio</i>	82
Figura 67	<i>Variable sensor fotorresistencia LDR en el Departamento</i>	83
Figura 68	<i>Variable sensor MG-811 en el Departamento de Eléctrica y Electrónica</i>	83
Figura 69	<i>Nodo-coordinador punto A</i>	83
Figura 70	<i>Nodo-emisor punto B</i>	84

ÍNDICE DE TABLAS

Tabla 1 <i>Parámetros Heltec WiFi LoRa 32 (v2)</i>	30
Tabla 2 <i>Características LoRa y Sigfox</i>	34
Tabla 3 <i>Características de bandas de LoRaWAN</i>	40
Tabla 4 <i>Propiedades principales de las topologías de red</i>	50
Tabla 5 <i>Parámetros del sensor BME280</i>	54

Resumen

El presente proyecto estuvo enfocado en implementar un prototipo de una red LoRa con sensores para el control medioambiental en el laboratorio de Sistemas de Comunicación, de la carrera de Tecnología Superior en Redes y Telecomunicaciones de la Universidad de las Fuerzas Armadas ESPE en su sede de Latacunga. El proceso estuvo fundamentado en una revisión y análisis bibliográfico, por la que fue posible recopilar información esencial para crear el prototipo y profundizar más sobre el Internet de las cosas (IoT), y uso del estándar LoRaWAN, que es la base esencial en la que se asienta este tipo de propuestas. El desarrollo implicó utilizar la tecnología del módulo Heltec WiFi LoRa 32 V2, sensores BME280, MG811 y un LDR, cada uno con el propósito de medir determinadas variables ambientales del entorno situando el despliegue de los sensores y que son visualizadas a través de una interfaz web de manera amigable con el usuario. Los datos recopilados por los sensores fueron humedad relativa, temperatura, presión barométrica, intensidad luminosa ambiental y CO₂, también se pudo identificar la distancia máxima de comunicación de los nodos de la red implantada. Al finalizar pudo elaborarse un herramienta rentable, apropiada, eficaz y práctica para el control medioambiental.

Palabras clave: LoRaWan, LoRa, IoT, Interfaz Web.

Abstract

The following degree project on the "Implementation of a prototype of a network with sensors for environmental control in the communication systems laboratory of the Higher Technology career in networks and telecommunications of the University of the Armed Forces ESPE Latacunga" is mainly focused on collecting data on environmental variables so that they can be displayed in real time through a web server to different users. The data considered to be collected are: relative humidity, temperature, barometric pressure, ambient light intensity, indoor CO₂. The development of the mentioned technical project will make use of Heltec WiFi LoRa 32 V2 technology, sensors BME280, MG811, and an LDR will be used, for the correct sending and receiving of data, and that can be visualized in a web interface. As a result of the research and prototyping, a cost-effective, appropriate, and practical tool for environmental monitoring has been obtained.

Key words: LoRaWAN, LoRa, IoT Web interface.

Capítulo I

Introducción

Antecedentes

Las redes inalámbricas son ampliamente utilizadas como por ejemplo en radioenlaces punto a punto, telefonía móvil, Wifi, entre otros. Además, producto del acelerado crecimiento tecnológico, el mundo se inclina hacia una inevitable implementación del Internet de las cosas (IoT, acrónimo en inglés), denominado así al hecho de que, todos los dispositivos estén conectados a Internet y, por tanto, tengan la capacidad de procesamiento e intercambio de datos.

Para implementar el IoT pueden utilizarse distintas alternativas de redes inalámbricas como SigFox, WiFi, BLE, Zigbee y LoRa. Esta última tiene la característica de que permite monitorear aspectos ambientales del entorno, implementar conexiones a largas distancias (WAN, *wide area network*) con un bajo consumo eléctrico, cuenta con ventajas para la comunicación entre equipos finales y central/Gateway y, por ende, se puede formar una *low power wide area network* (LPWAN, consumo de baja potencia) (Bertoletti, 2019).

Existen ya varias investigaciones previas que han abordado la implementación de LoRa. Uno de ellas es el estudio de Narváez y Contreras (2020), quienes se enfocaron en crear un prototipo de red de sensores IoT para monitorear parámetros ambientales en interiores y exteriores, lo que permite a su vez tener mayor conocimiento sobre LPWAN, WAN, LoRa y protocolo LoRaWAN, que son ideales para utilizarlas en zonas con baja cobertura y a bajo costo.

Por otro lado, vale también mencionar a Cantuña y Satián (2019), quienes aplicaron LoRaWAN para desarrollar un sistema de monitoreo de calidad de aire con el fin de mantener siempre un control de los indicadores y tomar medidas preventivas y correctivas. Además,

resaltan que esta tecnología es económica y bastante útil para emplearse en lugares con poca conectividad.

Frente a lo expuesto, el presente trabajo está centrado en desarrollar un prototipo que sirve como ejemplo del funcionamiento de una red LoRa a ser empleado en las instalaciones de la Universidad de las Fuerzas Armadas ESPE en su sede de Latacunga, específicamente en el campus Guillermo Rodríguez Lara, en el laboratorio de Sistemas de Comunicación de la carrera de Tecnología Superior en Redes y Telecomunicaciones. El objetivo es tener un control de información medioambiental a tiempo real y a su vez constituye una contribución sobre el estudio de este tema.

Planteamiento del problema

La Universidad de las Fuerzas Armadas ESPE es una institución emblemática de educación, innovación e investigación que forma profesionales en distintas ramas. Cuenta con diversos laboratorios distribuidos a lo largo de sus instalaciones, establecidos por departamentos y en los que son llevadas a cabo distintas prácticas necesarias dentro del proceso educativo.

De manera más específica, la carrera de Tecnología Superior en Redes y Telecomunicaciones, que está enfocada en graduar a profesionales capaces de llevar a la práctica de manera eficiente los conocimientos aprendidos, cuenta con su laboratorio en el que se pretende implementar el prototipo de red LoRa con sensores para el control medioambiental. Como fue ya mencionado, esta tecnología es ideal para medir el nivel de contaminación a bajo costo y puede emplearse en lugares con poca conectividad.

A lo largo del desarrollo del proyecto se analizan los distintos tipos de redes IoT, el uso del estándar LoRaWAN, los dispositivos y sensores utilizados, así como la máxima distancia a la que se pueden transmitir los datos para determinar una correcta comunicación

entre los nodos. Por su puesto, la práctica adquirida durante la implementación es un aporte para los alumnos de la carrera y permite expandir los conocimientos sobre redes inalámbricas.

Justificación

En la Universidad de las Fuerzas Armadas ESPE, en su Campus Guillermo Rodríguez Lara de Latacunga, funciona la Carrera de Tecnología Superior en Redes y Telecomunicaciones en donde se preparan profesionales que requieren adquirir conocimientos prácticos sobre nuevas tecnologías de comunicación. Es por ello que los laboratorios deben equiparse con dispositivos avanzados y brindar de esta manera una educación de calidad a la juventud.

El presente proyecto fomenta conocimientos sobre tecnología LoRa, redes IoT y sensores de control medioambiental. El propósito es implementar un prototipo de red LoRa con sensores para el control medioambiental en el interior del laboratorio de Sistemas de Comunicación con la finalidad de contar con datos de temperatura, CO₂, humedad relativa, presión barométrica e intensidad luminosa ambiente. La información y ejecución de la propuesta sirven como una guía para la adquisición de conocimientos y la puesta en práctica de comunicaciones inalámbricas.

Objetivos

Objetivo general

Implementar un prototipo de una red LoRa con sensores para control medioambiental en interiores para el laboratorio de Sistemas de Comunicaciones de la Carrera de Tecnología Superior en Redes y Telecomunicaciones de la Universidad de las Fuerzas Armadas ESPE Latacunga.

Objetivos específicos

- Analizar los distintos tipos de redes IoT, la utilización del estándar LoRaWAN y los dispositivos y sensores a utilizar en la red.

- Determinar un sistema de adquisición de datos para obtención de parámetros como temperatura, CO₂, humedad relativa, presión barométrica, intensidad luminosa ambiental, en diferentes locaciones.
- Implementar la red prototipo LoRa con sensores para control medioambiental en interiores.
- Analizar la máxima distancia en la que se puede implementar la red para determinar una correcta recepción de datos.

Alcance

El proyecto de investigación se enfoca en elaborar un análisis lógico y técnico de los equipos a utilizar, así como determinar el nodo con los que se establece la comunicación para configurar e implementar un prototipo de red LoRa con sensores para el control medioambiental al interior del laboratorio de Sistemas de Comunicación. Los equipos, materiales, herramientas y la adquisición de conocimientos que implica la ejecución del estudio son un aporte para los estudiantes de la carrera de Tecnología Superior en Redes y Telecomunicaciones. Por su puesto, el documento busca constituirse como fuente de información a la cual puedan acudir quienes a futuro requieran expandir su comprensión sobre el tema.

Capítulo II

Marco Teórico

Internet de las cosas (IoT)

El Internet de las cosas (IoT) es una red de equipos físicos interconectados que logran relacionarse entre sí a través de redes de comunicación, sistemas empotrados, dispositivos y aplicaciones en la nube, lo que permite establecer entornos inteligentes capaces de acceder, almacenar y recuperar datos informáticos e interactuar de manera directa con las personas (Quiñonez, 2019). Esta estructura hace parte del mundo tan tecnificado de hoy en día en el que se busca que exista mayor interconectividad entre los objetivos físicos y respondan a distintos eventos sin que sea necesaria la intervención humana. Es una manera de acortar la brecha entre el entorno físico y virtual y contar con un control a tiempo real de diversos datos gracias a la convergencia de redes inalámbricas, equipos inteligentes y sistemas de monitoreo electrónico.

Así mismo, como lo expresa Quiñonez (2019), contar con información detallada de distinta índole no solo permite controlarla y analizarla, sino tomar decisiones oportunas para mejorar los parámetros que sean necesarios. Este proceso ocurre por medio de una red de sensores WSN (*wireless sensor networks*) que mantiene conexión máquina a máquina (M2M, *machine to machine*) y permite recolectar datos, procesarlos y mostrarlos de forma amigable a los usuarios.

Arquitectura del Internet de las cosas

La estructura del IoT está conformada por arquitectura multicapas que dan paso a consolidar entornos inteligentes en ciudades, campos, transporte, edificaciones, áreas de la industria, hogar, salud, por nombrar unos ejemplos. Ello requiere implementar recursos tecnológicos como WSN, nanotecnología, identificación de radio frecuencia (RFID, radio frequency identification) y la nube en la que los equipos se interconectan y trabajan a través de

sus cuatro capas: de aplicación, de soporte de servicios y aplicaciones, red IoT y capa de dispositivo. (Rocha et al., 2021)

Capa de aplicación. Esta capa abarca aplicativos propios para IoT que permiten controlar el sistema gracias a la conexión que se establece con la nube y a los dispositivos informáticos implementados para analizar, recibir y transmitir información. (Garbarino, 2012)

Capa de soporte de servicios y aplicaciones. Esta capa trabaja con dos modelos de capacidad:

- Capacidades de soporte genéricas: tareas efectuadas por las aplicaciones IoT para el procesamiento o almacenamiento de información. Tienen la capacidad de asociarse a otras tareas específicas.
- Capacidades de soporte específicas: actividades que cumplen un trabajo específico en diferentes aplicaciones y forman grupos de diferentes funciones en las aplicaciones IoT.

(Garbarino, 2012)

Capa de red IoT. Esta capa está sustentada en dos aspectos:

- Red: ejerce el control de la conexión de red así como el acceso y gestión de transporte, autenticación, autorización y compatibilidad.
- Transporte: se encarga de proporcionar la conexión para enviar la información de control y gestión.

(Garbarino, 2012)

Capa de dispositivo. Aquí pueden visualizarse dos tipos de funciones: de dispositivo y pasarela.

a) Funciones de dispositivos

- Comunicación de red directa: los equipos obtienen, cargan y envían información directamente hacia los demás dispositivos incorporados en la red.

- Comunicación de red indirecta: los equipos obtienen, cargan y envían información de manera indirecta y mediante el empleo de pasarelas.
- Redes ad-hoc: los dispositivos están en la posibilidad de crear redes ad-hoc cuando sea necesario mejorar la capacidad de red y velocidad de despliegue.
- Modalidad de reposo y activo: los equipos deben contar con mecanismos que automáticamente trabajen con modos de reposo y activo para ahorrar energía.

b) Funciones de pasarela

- Interfaces múltiples: capacidad de que los dispositivos puedan conectarse por diferentes tecnologías alámbricas e inalámbricas como redes de control de zona, ZigBee, Bluetooth, WiFi, SigFox, LoRa, etc. Además, se establece comunicación por medio de diferentes tecnologías, redes móviles de tercera o cuarta generación (3G o 4G) y nueva generación (5G).
- Protocolo: hay dos razones de empleo de las funciones de pasarela. La primera, cuando la comunicación en la capa de dispositivo usa diferentes protocolos. La segunda es cuando la comunicación trabaja conjuntamente entre la capa de dispositivo y red mediante protocolos diferentes en capa (Chivita, 2020).

Para tener una mejor claridad, a continuación, la Figura 1 expone visualmente la arquitectura del IoT.

Figura 1

Arquitectura del Internet de las cosas (IoT)



Nota. Arquitectura multicapas que interconecta dispositivos para realizar diferentes tareas.

Plataformas de Internet de las cosas

Una plataforma IoT es un espacio en el que trabajan en conjunto *hardware*, *software*, distintas conexiones y al margen de una arquitectura que brinda una solución eficiente de monitoreo y gestión de los dispositivos, recopilación y análisis de datos y conectividad con la nube. En otras palabras, es el conjunto de servicios IoT para el desarrollo, despliegue, sostenimiento, evaluación de los datos y que además puede conectarse con otras plataformas (Joyanes, 2021).

La creación de plataformas IoT está en su momento de auge y puede ser mediante software de código abierto o cerrado. Los elementos que la componen son los siguientes:

- Conexión con diferentes protocolos y formatos de datos con su estandarización correspondiente.
- Gestión sobre los equipos.
- Bases de datos.
- Procesamiento de información y toma de acciones.
- Análítica de datos.

- Visualización.
- Interface externa.
- Otras herramientas.

Por su puesto, como lo expresa Joyanes (2021), toda plataforma IoT tiene que trabajar con un alto nivel de seguridad de la información, algo esencial para que las tareas de administración, monitoreo, procesamiento, análisis y almacenamiento sean realizadas adecuadamente y sin riesgo alguno. En ese sentido, el autor expone que la seguridad debe visualizarse en cuatro elementos esenciales:

- Hardware. Dispositivos y sensores que permitan la recopilación de información entre todos los nodos de la red.
- Conexión. Infraestructura complementaria para transmitir y recibir los datos hacia la nube.
- Software. Esencial para analizar los datos recopilados por los sensores.
- Interfaz gráfica. Entorno que permite a la persona relacionarse con el sistema: una aplicación web, servidor web, tablero de control, etc.

Arduino

Arduino es una plataforma electrónica de código abierto (software y hardware libre) y de fácil uso. Sus módulos pueden descifrar entradas como por ejemplo mediante un sensor de luz o de calibración que transforma la información hacia una acción en particular: prender o apagar un LED, activar el trabajo de sensores, entre otros. El proceso consiste en que el módulo transmita diversas tareas al microcontrolador gracias al cableado y a que está sustentado en el entorno de programación Arduino IDE (Arduino, 2018).

Heltec Wifi LoRa 23 (v2)

Como lo expone HiTech Chain (s.f.) en su plataforma web, WiFi LoRa 32 es una placa de desarrollo IoT diseñada y producida por Heltec Automation (TM). Un producto altamente

integrado basado en ESP32 + SX127x, tiene funciones Wifi, BLE, LoRa, sistema de administración de batería Li-Po e incluye OLED de 0,96.

Características

- Microprocesador ESP32 (MCU de 32 bits de doble núcleo + núcleo ULP) con chip de nodo LoRa SX1276/SX1278.
- Interfaz micro USB con regulador de voltaje, protección electrostática ESD, salvaguarda cortocircuitos, blindaje RF.
- Interfaz de batería SH1.25-2 y sistema de gestión de batería de litio integrado: carga y descarga, protección contra sobrecarga, detección de energía de la batería, conmutación automática de energía de batería/USB.
- Wifi integrado, LoRa, tres conexiones de red Bluetooth, antena de metal de 2,4 GHz, interfaz IPEX (U. FL) reservada para uso de LoRa.
- Pantalla OLED de 0,96 in que muestra la información de depuración.
- Chip integrado CP2102 USB a puerto serie conveniente para la descarga de programas e impresión de información. Trabaja con entorno de desarrollo de Arduino.
- Desarrollada con librerías de ESP32 + LoRaWAN. El protocolo LoRaWAN es un estándar que puede comunicarse con cualquier puerta de enlace LoRa que sea capaz de ejecutarlo.
- Buen diseño de circuito de RF y diseño básico de baja potencia (corriente de reposo $\leq 800\mu\text{A}$). Es conveniente para los proveedores de aplicaciones de IoT.

Especificaciones técnicas

A continuación, la Tabla 1 detalla las características.

Tabla 1*Parámetros Heltec WiFi LoRa 32 (v2)*

Tipo	Parámetros
MCU	ESP32 (240mhz Tensilica LX6 dual-core + 1 ULP, 600 DMIPS, 520KB SRAM, Wi-Fi, modo dual Bluetooth)
LoRa Chip	SX1276 (868 and 915 versión); SX1278 (433 and 470 versión)
Bandas LoRa	EU_433, CN_470_510, EU_863_870, US_902_928 Trabaja con cuatro bandas opcionales.
Potencia de salida máxima de LoRa	18dB ± 2dB
Protocolos de comunicación	3 UART; SPI x 2; I2Cx2; I2Sx1
Pines	29 GPIO generales
Pantalla	Pantalla OLED de 0,96 pulgadas 128*64
Flash	8MB(64M-bits) SPI FLASH
Interface	Micro USB x 1; LoRa antena interface (IPEX) x 1
Tamaño	50.2 x 25.5 x 9.74 mm
Puente USB a UART	CP2102
Batería del circuito	3.7V Litio (SH1.25 x 2)
Diseño de bajo consumo	Sueño profundo 800uA

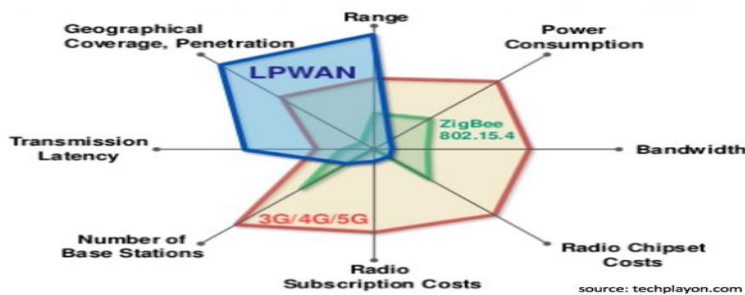
La tabla muestra los distintos parámetros de hardware de la placa Heltec WiFi LoRa 32 (v2) que es la base para elaborar el prototipo de la red LoRa. Así mismo, la Figura 2 lo expone visualmente.

Figura 2*Heltec WiFi LoRa 32*

Nota. La siguiente figura muestra los dispositivos por los cuales está equipado el módulo.

LPWAN

Como lo expone Joyanes (2021), las redes LPWAN (consumo de baja potencia) y WAN (comunicación de larga distancia) fueron desarrolladas para llevar a cabo requerimientos vinculados con las bandas ISM (industrial, científica y médica por sus siglas en inglés). Las primeras, sobre todo, han ganado terreno en la industria comercial, de salud e investigación debido a sus características de bajo consumo, largo alcance, bajo coste, la capacidad para adaptarse a aplicaciones IoT, lo que les torna en un recurso ideal para implementarse en interiores y exteriores y diseñar sensores inalámbricos personalizados (ver Figura 3).

Figura 3*Tecnología LPWAN*

Nota. Comunicación inalámbrica que transmite información desde sus nodos hacia una estación base

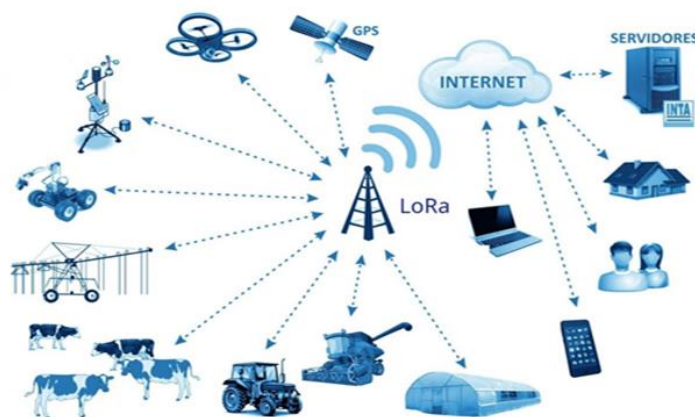
Bandas sin licencia

Como lo expresa Krupka et al. (2016), existen tecnologías esenciales que utilizan bandas sin licencia para la ejecución de LPWAN, llamadas así por el Instituto Europeo de Normas de Telecomunicaciones (ETSI) y por la Unión Internacional de Telecomunicaciones (UIT); son usadas para radiofrecuencia electromagnética no comercial en el área industrial, científica y médica. En los siguientes apartados se exponen cada una de ellas.

Tecnología LoRa. Es una tecnología desarrollada por LoRa Alliance y patentada por Semtech (fabricadora de chips de radio). Forero y Camberos (2020) expresan que fue concebida desde un inicio para implementarse en comunicaciones de datos a una baja velocidad y con gran potencia para largas distancias; está asociada con las redes de comunicación IoT y dispositivos M2M. Es una red inalámbrica basada en modulación CSS (*chip spread spectrum*), una modulación por pulsos que disminuye o aumenta a medida que la información es transmitida. Además, la transmisión de señales de baja potencia es concedida a través del bloqueo por desplazamiento de frecuencia (FSK). A continuación, la Figura 4 expone un ejemplo de esta tecnología.

Figura 4

Tecnología LoRa



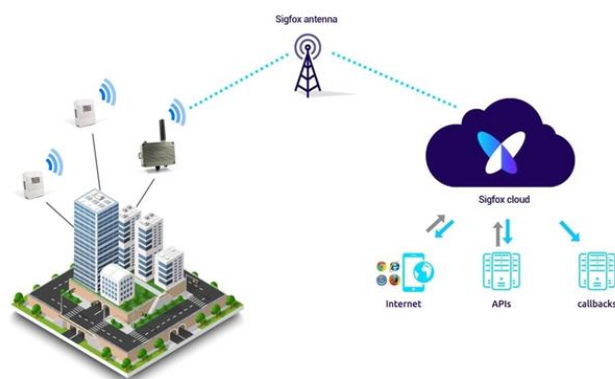
Nota. Tecnología para desplegar conexiones de largo alcance. Ideal para redes IoT porque incorpora sensores para su intercomunicación.

Tecnología Sigfox. Fue desarrollada por dos ingenieros franceses en 2009 en torno a sus estudios de señales inalámbricas e IoT. Sigfox utiliza UNB (*ultra narrow band*), banda diseñada para trabajar a bajas velocidades, transferir datos de 10 a 1000 bps y emplea un método de transmisión por radio estándar denominado codificación binaria de cambio de fase (BPSK por sus siglas en inglés). Así mismo, ocupa fragmentos de espectro estrechos al cambiar la fase de la señal portadora para codificar los datos y contrarrestar el ruido, lo que da paso a entablar una comunicación de largo alcance sin mayor inconveniente. Finalmente, cabe mencionar que esta tecnología utiliza bandas de frecuencia sin licencia y estandarizadas de la siguiente manera: en Europa opera a 868 MHz, en Estados Unidos trabaja a 902 MHz y en Asia opera a 433 MHz (Aprendiendo Arduino, 2020).

Con la finalidad de comprender de manera sencilla este esquema tecnológico, se presenta a continuación la Figura 5.

Figura 5

Tecnología Sigfox



Nota. Una de las redes IoT ideal para la conectividad de telefonía móvil. su comunicación es a baja velocidad y tiene un consumo de energía bastante bajo.

Con el propósito de resumir de manera de lo expuesto, la Tabla 2 detalla las características de ambas tecnologías.

Tabla 2*Características LoRa y Sigfox*

Detalle	LoRa	Sigfox
Bandas de frecuencia	433 - 868 MHz (Europa) 915 MHz (N América) 433 MHz (Asia)	868 MHz (Europa) 902 - 915 MHz (N América) 433 MHz (Asia)
Rango máximo	3-8 Km (Zona Urbana) 30-50 Km (Zona Rural)	30-50 km (Zona Rural) 3-10 km (Zona Urbana)
Consumo de energía	Tx: <50 mA Rx: 10-40 mA	Tx: <50 mA Rx: 10-40 mA
Tasa de datos	0.25 - 50 kbit/s	100 bit/s (Europa) 600 bit/s (N America)
Topología	ESTRELLA	ESTRELLA
Modulación	CSS (FSK)	UNB (BPSK)
Potencia de transmitida	10 – 18 dBm	0 – 14 dBm

Protocolo LoRaWAN

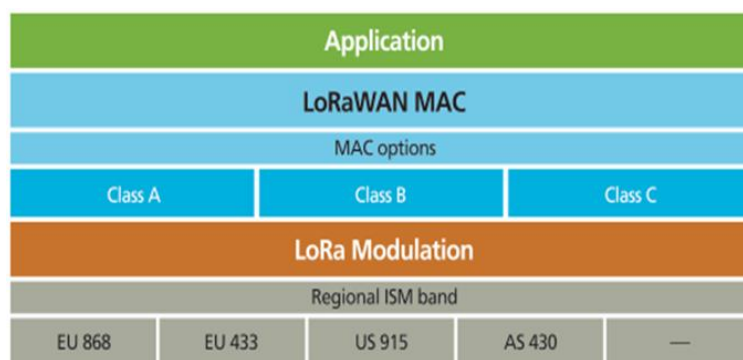
LoRaWAN es un protocolo de capa de control de acceso a medios (MAC) que fue creado para establecer una conexión inalámbrica de internet y los dispositivos suministrados de energía por batería (IoT hace uso del cifrado de extremo a extremo). Opera como protocolo de gestión de red para la tecnología LPWAN; permite que distintos terminales se comuniquen a través de la puerta de enlace por medio de la modulación LoRa, por lo que es el encargado del enrutamiento entre puertas de enlace y nodos finales (Reimondo, 2019).

Este protocolo gestiona la banda de frecuencia en la que trabajan los dispositivos para ensanchar la comunicación, velocidad de datos y potencia los del nodo final. Los datos enviados por los dispositivos instalados en el nodo final son recibidos por varias puertas de enlace que envían los paquetes de información a un servidor central en la red. El servidor filtra los datos que se repiten, verifica la seguridad y la información es remitida a los servidores de las aplicaciones LoRaWAN. Vale mencionar que se usa criptografía AES para la seguridad y

combina CMAC para protección de integridad y CTR para encriptación (Reimondo, 2019). A continuación, la Figura 6 detalla la estructura del protocolo.

Figura 6

Estructura protocolo LoRaWAN



Nota. protocolo de capa de red abierto y estandarizado que define cómo los dispositivos se comunican entre sí y con una red de gateway.

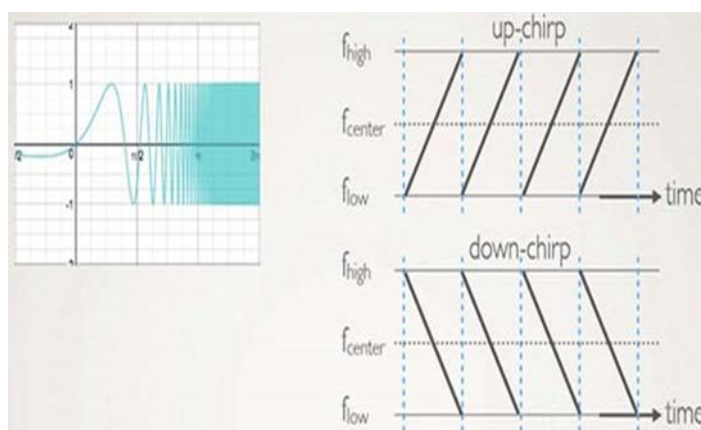
Modulación LoRaWAN

La modulación LoRaWAN utiliza una variante de la modulación DSSS, llamada *chirp spread spectrum* (CSS), como esquema de modulación de la señal. Es un método de ensanchamiento que utiliza pulsos de chirp modulados en frecuencia lineal para codificar los paquetes de información. Entre sus ventajas destaca la conectividad a bajo costo, poco consumo energético (25mA de transmisión y 10mA de recepción), gran capacidad para evadir el ruido y solidez para evitar el efecto Doppler debido a que los datos son modulados en distintos canales y velocidades (Ordóñez, 2017).

Con respecto al ancho de banda, se procesa las señales de RF que normalmente se establece en 125 kHz, pero puede extenderse a 250 kHz e incluso a 500 kHz para eventos particulares. Un aspecto interesante es que tiene la capacidad de recibir datos de alta sensibilidad (-168 dB) y baja transmisión hasta 255 bytes (Ordóñez, 2017). La Figura 7 detalla visualmente lo expuesto.

Figura 7

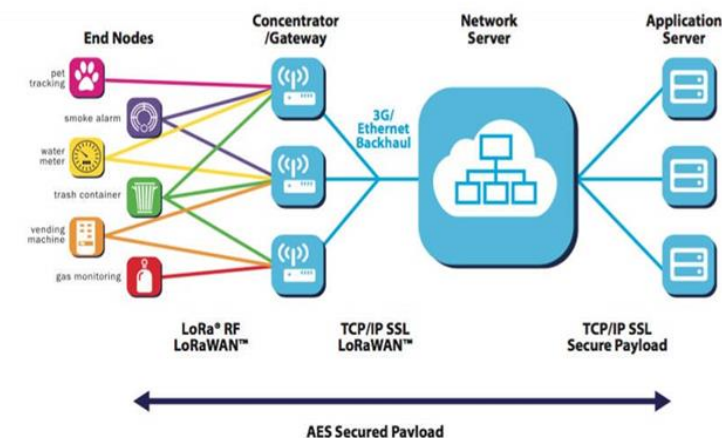
Modulación LoRaWAN



Nota. La modulación LoRa se basa en la técnica de modulación CSS que permite una mayor eficiencia espectral y mejor resistencia a los ruidos.

Arquitectura LoRaWAN

Delgado et al. (2021) exponen que usualmente son utilizados dos tipos de topologías: red en malla y estrella. La diferencia entre ambas es que la primera tiene la posibilidad de autoconfiguración y organización, mientras que en el segundo caso los nodos se conectan a un nodo central que les abre comunicación hacia un nivel más alto dentro de la red. La arquitectura estrella suele ser más empleada (requiera preservar la duración de vida útil de las baterías) debido a que establece una comunicación entre nodos y puertas de enlace, estructura una modulación de largo alcance, con bajo consumo de potencia, lo que posibilita el uso de nodos sencillos sin afectar la cobertura de red. La Figura 8 expone un esquema de esta arquitectura.

Figura 8*Arquitectura de red LoRaWAN*

Nota. En resumen, la arquitectura LoRaWAN proporciona una arquitectura de red abierta y estandarizada para conectar dispositivos IoT y M2M a la red.

Estructura de la trama física

Como lo expone Alonso (s.f.), la estructura de la trama de la capa física de LoRa está constituida por cuatro secciones, preámbulo, Header, PHY CRC y carga útil, aunque a momentos suele implementarse otra sección denominada CRC. Este preámbulo establece una sincronización entre el transmisor y receptor, que alcanza una longitud de 10 hasta 65 530 símbolos; cabe indicar que el segmento fijo tiene cuatro símbolos, pero el resto es programado con longitud mínima de seis símbolos y longitud extrema de 65 532. La función del preámbulo arranca con pulsos de estridencia de subida (upchirp) -cuya cantidad depende del tipo de programación asignada-, continúa con pulsos de sincronización y termina con pulsos tipo chillido de bajada (downchirp).

Para una comprensión más profunda se detalla a continuación una explicación sobre el preámbulo, FHDR + PHDR-CRC, PHY PAYLOAD y CRC con base en Alonso (s.f.):

- **Preámbulo.** La trama fija el esquema de modulación de los datos que puede variar para adecuarse al uso o exigencia del nodo central y así optimiza el consumo eléctrico del

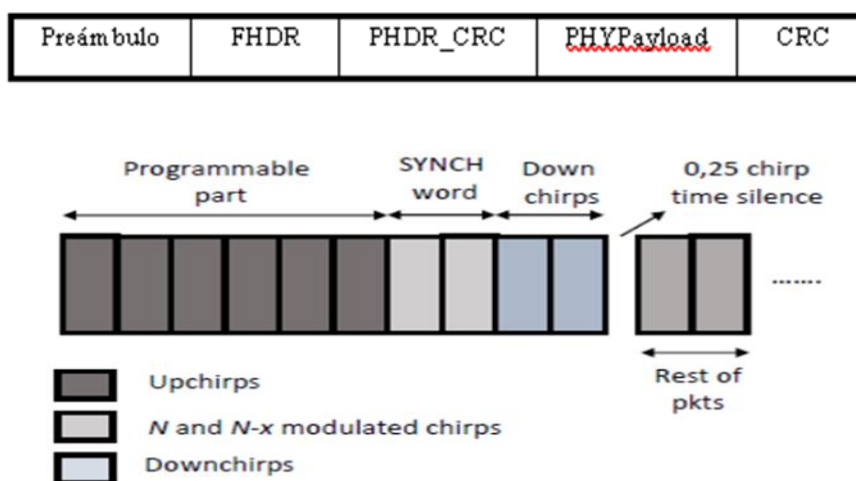
sistema. El preámbulo termina con un byte de sincronización que distingue entre redes LoRa que transmiten en la misma banda de frecuencia; el dispositivo reconoce solo transmisiones con el mismo byte de sincronización, lo que permite el uso simultáneo de varias redes LoRa dentro de la misma banda de frecuencia.

- **FHDR + PHDR-CRC.** Al finalizar el preámbulo, la parte de la trama avanza con la cabecera y código de redundancia y evita el envío de tramas erróneas que trabajan con 20 bits.
- **PHYPAYLOAD.** Envío de los paquetes de datos que deben ser transmitidos en la trama de la capa de enlace en máximo 255 bytes, debido a que la parte del payload es establecido en un solo byte en la cabecera.
- **CRC.** El código de redundancia cíclica asegura que el proceso y envío de la trama sean correctos para la lectura de los componentes de la red LoRa.

La Figura 9 presenta la trama física de esta estructura.

Figura 9

Trama física LoRaWAN



Nota. La trama física de LoRaWAN es la estructura básica de un paquete de datos transmitido a través de la tecnología LoRa. Consta de varios campos que incluyen información sobre el dispositivo que transporta los datos, el destinatario y el contenido.

Dispositivos que conforman la red LoRaWAN

Como lo indica Córdoba (2017), LoRa es una red constituida por los siguientes elementos:

- **Servidor de red.** Autentica los nodos, comprueba la integridad de los mensajes y los enruta a los servidores de aplicación.
- **Servidor de aplicación.** Incorpora mecanismos de confidencialidad de mensajes y gestiona las solicitudes.
- **Dispositivo o motas.** Dispositivos de comunicación que trabajan a baja potencia, con poco consumo energético, para evaluar la información recopilada del lugar.
- **Red.** Ruta de comunicación que permite enviar y receptor mensajes recibidos por los nodos desplegados hacia una aplicación para su análisis.
- **Gateway.** Se instala un gateway para entablar intercomunicación entre dispositivos y nodos y transmitir a Internet los datos. De esta forma se logra visualizar la información desde cualquier lugar.
- **Aplicación.** Dispositivo capaz de ejecutar un software que puede interpretar datos y leer información recibida por la puerta de enlace.
- **Mensaje de enlace ascendente.** Emite los paquetes de información desde un equipo específico hacia la aplicación.

Frecuencias empleadas para redes LoRaWAN

LoRaWAN utiliza un espectro de radiofrecuencia disponible gratuitamente, es decir que cualquiera puede usarlo sin recargo alguno, y es muy similar a las de 2,4 GHz y 5 GHz. La implementación de frecuencias de bajo espectro aumenta la relevancia del radio, pero hay que tener en cuenta que el uso de frecuencias libres depende del país e incluso del continente (Córdoba, 2017).

Como lo explica Quispe (2019), esta red utiliza nueve canales de frecuencia para la transmisión de datos y se distribuyen en siete zonas; en su caso se utilizó la frecuencia 915 correspondiente a la región de América del Norte (Tabla 4) que consta de 64 canales de acceso disponibles, de 0 a 63, de 125 kHz. Desde 902,3 y hasta 914.9 MHz tiene 8 canales de enlace ascendente agregados (500 kHz cada uno) (canales 64-71); en cambio, a partir de 903 MHz y en incrementos de 1,6 MHz a 914,2 MHz se cuenta con 8 enlaces descendentes (500 kHz cada uno); finalmente, para comunicación de puerto a dispositivo hay 8 canales, de 0 a 7, desde 600 kHz hasta 927,5 MHz.

Tabla 3

Características de bandas de LoRaWAN

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

Seguridad

Como lo indica Reimondo (2019), LoRaWAN proporciona el cifrado y el sello de los paquetes de información mediante claves simétricas conocidas por los nodos finales, la red y el servidor de aplicaciones que se entregan de dos maneras y según el método de activación utilizado: activación inalámbrica (OTAA) y activación personalizada (ABP). Estos mecanismos protegen y eliminan los posibles ataques a los dispositivos finales del nodo; además, LoRaWAN facilita diversas capas de cifrado AES-128 para la protección de los datos transmitidos:

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

Clases de dispositivos LoRaWAN

Si bien la capa LoRa física está reservada, las capas LoRaWAN restantes están abiertas y descritas. La topología es una estrella que consta de puertas de enlace y dispositivos finales que se comunican de forma inalámbrica con puertos basados en modulación LoRa y puertas de enlace que envían y reciben datos desde la computadora a través de un canal de banda ancha eficiente y confiable.

Los dispositivos LoRa pueden ser de tres clases: A, B y C. Los de clase A son terminales bidireccionales que proporcionan comunicación bidireccional con estrictos requisitos de potencia y rendimiento, lo que significa que reciben paquetes de datos siempre que no se hayan enviado paquetes antes. La razón de ello es que cada vez que el dispositivo envía datos se abren dos puertos de recepción: el primero, un segundo después de que se transmiten los datos; el segundo, un segundo después de que el primero es capaz de recibir el paquete nuevamente. en los terminales de clase A, la transmisión de los datos es realizada por tiempos

determinados y también puede hacerse por eventos, por ejemplo, si la temperatura supera los 16° o por debajo de 19°.

Por otro lado, los de clase B son terminales bidireccionales con sockets de recepción capaces de recibir datos sin enviar paquetes. Estos equipos levantan puertos de entrada adicionales en momentos determinados, lo que le permite al servidor saber cuándo la terminal escucha.

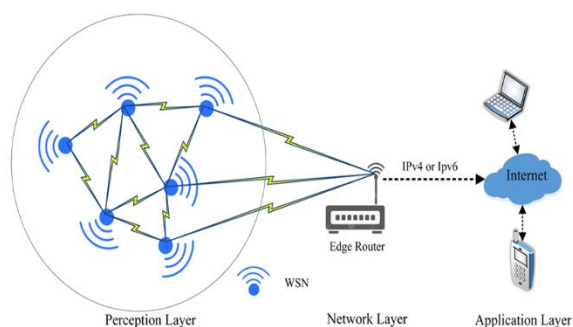
Finalmente, los de clase C son terminales bidireccionales con intervalo máximo de recepción. Aquí, el dispositivo está en modo de escucha para recibir paquetes casi continuamente hasta que se transmiten, y una característica relevante es que ofrece una mejor latencia de conexión que los dispositivos de clase A y B pero con mayor costo de consumo de energía.

Red de Sensores Inalámbricos (WSN)

La red de sensores inalámbricos (WSN) es un sistema integrado por varios nodos de sensores independientes (energizados por baterías), y es parte inherente del IoT. Su trabajo consiste en recolectar información medioambiental con una baja potencia de procesamiento y con la facultad de comunicarse con otro equipo de manera inalámbrica, proceso que es realizado a través de una sucesión de sensores que son parte del nodo y están aptos para medir la temperatura, humedad, luminosidad y, en general, características y/o eventos de los objetos ubicados alrededor del nodo. Esta información es enviada por el sensor hacia una estación base o receptor (Cobo et al., 2020).

Características de una de red de sensores inalámbricos

Las redes de sensores inalámbricas están determinadas en torno a tres áreas de tecnología e investigación: rastreo, comunicación y computación que incluye hardware, software y programación (Pérez et al., 2014). Los nodos de sensores están instalados a campo abierto como se visualiza en la Figura 10.

Figura 10*Estructura de red de sensores inalámbricos*

Nota. Cuentan con la capacidad de recopilar los datos de lectura de cada sensor o nodo y después direccionar la información hasta el nodo central mediante una arquitectura ad hoc.

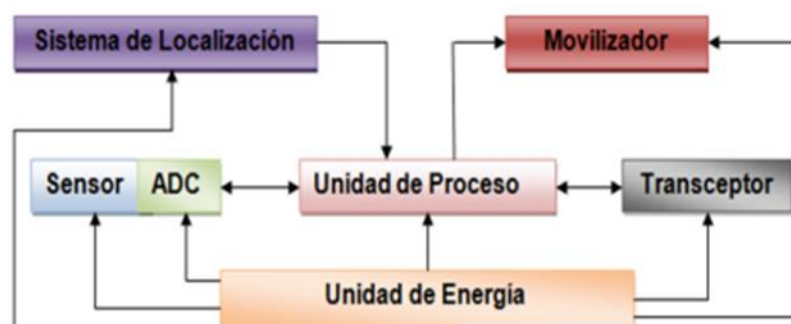
Cada sensor instalado en la red inalámbrica tiene la facultad tanto para recopilar de datos como de direccionarlos y enviárselos al nodo central. El nodo que recopila la información es capaz de comunicarse con el nodo que gestiona las tareas por medio de internet, satélite o de forma directa. Como lo menciona Pérez et al. (2014), la estructura de una red de sensores inalámbricas se conforma de los siguientes factores:

- **Fiabilidad a errores.** Ciertos sensores pueden ocasionalmente presentar errores o fallos debido a que la batería se agota, tiene daños físicos o interferencias por las condiciones medioambientales. Los errores o fallos de los nodos sensores no compromete la disponibilidad y funcionalidad de la red.
- **Escalabilidad.** La estructura debe proporcionar la capacidad de diseñar la red y ampliarla con centenares de sensores.
- **Coste de producción.** El costo de adquisición y despliegue del primer nodo sensor es primordial para evaluar el factor económico: si el costo de despliegue es más alto que las redes tradicionales perderían su importancia debido a que la red de sensores podría estar estructurada con gran número de nodos.

- **Delimitación de equipos.** Los sensores incorporan cuatro elementos básicos: sensor de conversión ADC (Analog to Digital Converter), unidad de proceso, transceptor de radio y unidad de energía. Cada uno varía según si la aplicación usada es de localización o movimiento (Figura 11).
- **Uso energético.** Los dispositivos instalados en la red de sensores generalmente se suministran con fuentes de energía limitadas. En algunos nodos, la recarga es casi inalcanzable; es decir, dependen del tiempo de vida útil de la batería.

Figura 11

Estructura del sensor inalámbrico



Nota. Los sensores están instalados en el campo requerido. Cuentan con la capacidad de recopilar los datos de lectura de cada sensor o nodo y después direccionar la información hasta el nodo central mediante una arquitectura *ad hoc*.

Topologías para redes de sensores inalámbricas

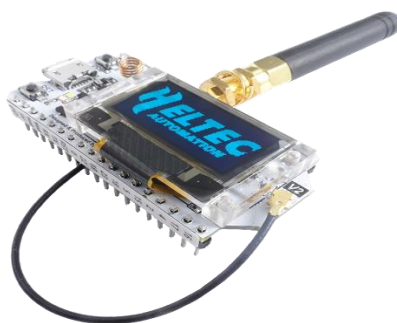
Existen varias arquitecturas utilizadas para el despliegue de una aplicación WSN: topología estrella, malla, punto a punto, o híbrida. Todas pueden trabajar en conjunto; su función es enlazar la configuración de los elementos (hardware), enviar datos y cada una cuenta con determinadas particularidades. Guijo, (2021)

Para comprender las topologías descritas es necesario conocer los elementos de las redes de sensores inalámbricas que son expuestos a continuación:

Nodos o motes. Sensores pequeños cuyo trabajo es recopilar datos del lugar en el que se encuentra instalada la red (ver Figura 12). (Perez,2014)

Figura 12

Heltec Wifi LoRa 32 V2



Nota. Placa de desarrollo ESP32 con 8 MB Flash que utiliza directamente el chip EPS32, integra un SemTech SX1276 o SX1278 para comunicación LoRaWAN y una pantalla OLED SSD1306 de 0,96 in y 128x64 conectada a través de I2C.

Estación base. Nodo encargado de formar el canal para la comunicación y establecer el PAN ID (identificador de red) de forma global para la red. Una vez instaurados los parámetros, la estación base crea la red, da acceso a conectarse y sincronizar los dispositivos finales. Una vez disponible la red, la estación base enruta la información y se convierte en origen y destino de los datos recopilados. (Perez,2014)

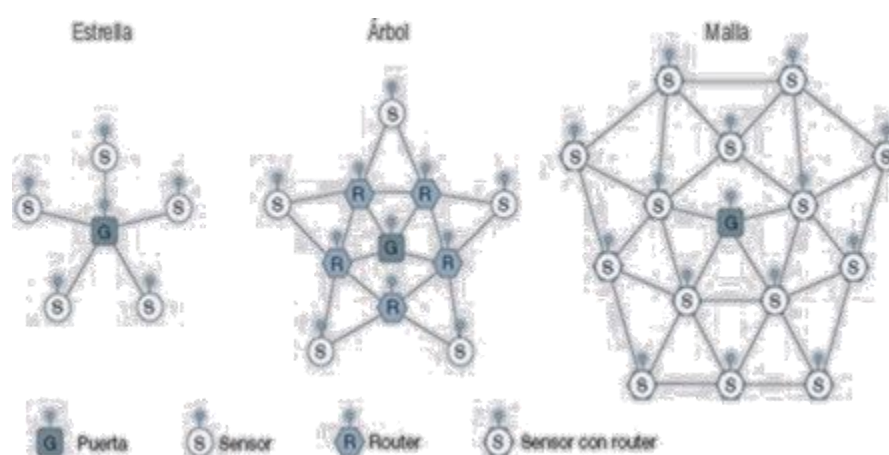
Routers. Interconecta los dispositivos distantes en la topología de red de sensores (dispone un nivel de aplicación para la ejecución de código del usuario). El router es un nodo de procesamiento y recopilador de información que establece la mejor ruta para enviar los datos. (Perez,2014)

Puertas de enlace. Recolecta los datos obtenidos por los sensores y crea un punto de unión con la red LAN o Internet. Todos los nodos de la red están comunicados directamente hacia la puerta de enlace entre 30 a 100 metros, y la mayoría de ellos son similares para que la puerta puede receptor información, así como monitorizar la red y los sensores instalados.

Además, la topología en malla se entiende como un método multisalto en el que los nodos se convierten en routers y pueden enviar y recibir información entre ellos al igual que la puerta de enlace; es extremadamente flexible ante fallos porque cada nodo puede tener distintas rutas para comunicarse con la puerta de enlace (Pérez, 2014). A continuación, la Figura 13 expone visualmente un ejemplo.

Figura 13

Topologías de la red de sensores inalámbrica



Nota. Topologías de red de sensores inalámbricas más utilizadas.

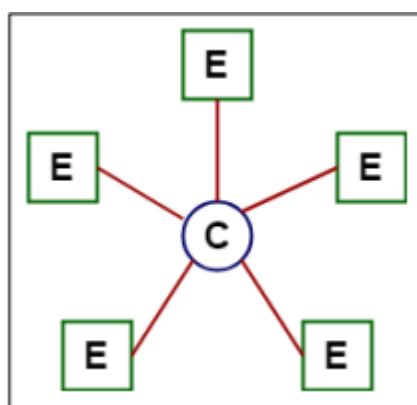
Topología Estrella

Como lo expresan Lizbeth y Fernanda (2019), una topología estrella es necesaria para sincronizar los puertos o transmisores WSN porque está ubicada en el centro de la red a una distancia aproximada entre 30 a 100 metros de los nodos. Aquí, la información a transmitir solo toma un paso y únicamente los nodos sensores están conectados de forma directa con el *gateway*; hay que tomar en consideración que los nodos finales no se pueden comunicar entre ellos. Dicho de otra manera, la información no es visible por los nodos y tendrán que publicarla mediante la puerta de enlace que, en definitiva, es utilizada para transferir datos al exterior y permite monitorear la red.

Las autoras mencionan que un inconveniente de esta topología es que la distancia de transmisión entre los dispositivos de red es limitada, pero su consumo de energía es mínimo comparado con el resto de topologías. Si llegara a darse el caso hipotético de que alguno presente congestión, no existe una ruta alternativa y se perdería la información. A continuación, la Figura 14 expone un esquema de esta topología.

Figura 14

Topologías estrella



Nota. La letra C representa al nodo coordinador -gateway- y la letra E los nodos finales.

Sistema de red para monitoreo de variables medioambientales

De acuerdo con Chapter (2019), los contaminantes del aire pueden clasificarse en dos: primarios, cuando se emiten directamente en la atmósfera ya sea por actividades humanas o por origen natural; secundarios, si se forman en la atmósfera como resultado de reacciones químicas (hidrólisis, oxidación o reacción fotoquímica). Entonces, los contaminantes primarios son los gases emitidos por vehículos y automotores (monóxido de carbono y dióxido de nitrógeno) que dependen del uso de combustibles fósiles como la gasolina y que contribuyen significativamente al deterioro de la atmósfera; también existen fuentes naturales de gases contaminantes como los incendios y erupciones volcánicas.

Los contaminantes secundarios son aquellos que tienen cambios químicos una vez que llegan a la atmósfera, por ejemplo, el ozono emitido por los vapores orgánicos y los óxidos de

nitrógeno provenientes de los gases del escape de los automóviles o de las gasolineras. El óxido de nitrógeno reacciona con los vapores orgánicos gracias a la luz solar, lo que da paso a que surja así el ozono que es el principal componente de la niebla fotoquímica.

Frente a lo mencionado, para el control ambiental es necesario construir una red de monitoreo con redes inalámbricas de sensores capaces de identificar contaminantes ambientales y controlar eficazmente temperatura, CO₂, humedad relativa, presión biométrica, intensidad luminosa ambiental, etc. Sin duda, es crucial medir la calidad del aire y evitar la concentración de componentes tóxicos en el entorno que afecten a la salud y bienestar de la población (Narváez y Contreras, 2020).

Sensores

También conocido como nodo, es un dispositivo diseñado para recibir información externa sobre una variable y convertirla en una cantidad generalmente eléctrica (impulso eléctrico) que no es posible definir ni en número ni en actividad. Con frecuencia, estos dispositivos consisten en componentes pasivos (anemómetro, PTC, NTC, LDR, etc., todos de capacitancia variable) que utilizan otros activos y convierte así variables físicas (luz, magnetismo, presión, etc.) mediante tres pasos:

- El sensor registra el fenómeno físico a medir y muestra en su salida una señal eléctrica en función del valor de la cantidad física que es regulada por el control del sistema de acondicionamiento de señal. Posee una salida de tensión.
- El sensor contiene un circuito que convierte y/o amplifica el voltaje de salida del sensor que se transmite al convertidor A/D y está conectado a una computadora.
- El convertidor A/D convierte la señal de voltaje de CC en una señal discreta (Cedeño y Quiroz, 2015).

Clasificación de Sensores

Un sensor se define generalmente como un dispositivo que interpreta información del exterior mediante impulsos eléctricos. Luego de su análisis y procesamiento envía una respuesta eléctrica que puede ser manipulada y cuantificada; el elemento que recibe la señal es el sensor, mientras que el transductor es el elemento que convierte una cantidad en otra. (Stefany, 2020)

Los sensores tienen tres parámetros básicos: rango (valores entre valores medibles), resolución (variación mínima que se puede detectar) y sensibilidad (cuánto varía la salida con variaciones en el mensurando). La manera de clasificarlos está dada en función de varios criterios:

- Por la señal que envían: analógicos y digitales (o discretos).
- Según cómo afectan al proceso: sensores pasivos o activos.
- Dependiendo de los parámetros cambiantes: mecánicos, eléctricos, electromagnéticos, ópticos, etc.

Estructura de red para Sensores Inalámbricos

Existen diferentes tipos de arquitecturas que pueden emplearse en la implementación de una aplicación WSN como por ejemplo estrella, malla, híbrida, entre tantas otras con sus propios desafíos, ventajas y desventajas.

La arquitectura -topología- hace referencia a la configuración de los dispositivos (hardware) y cómo los datos son transmitidos a través de esa configuración. Cabe resaltar que cada una es conveniente dependiendo de la particularidad y necesidades de la red y puede ser inapropiada en determinados casos (Narváez y Contreras, 2020).

A continuación, la Tabla 4 detalla las propiedades principales de las topologías.

Tabla 4

Propiedades principales de las topologías de red

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

Nota. Tomado de Narváez y Contreras, 2020.

Sensores para el monitoreo ambiental

La naturaleza y sus diferentes cambios no pueden ser controlados, y por eso es necesario implementar sistemas de medición confiables para monitorear el entorno. En ese sentido, los sensores de un sistema de monitoreo (como el WSN) cuentan con una extensa gama de variables a medir, pues la contaminación ambiental no solo se refiere a los desechos o componentes tóxicos en el agua, tierra o aire sino también a ruidos producidos en los audios. El fin es registrar, supervisar y organizar datos recopilados que son transmitidos de forma autónoma mediante canales inalámbricos y que pueden ser visualizados en aplicaciones y/o programas (Vidal, 2014).

Sensor de Luminosidad

Este módulo incorpora un fotorresistor, también llamado LDR (*light dependent resistor*), cuya resistencia cambia con la cantidad de luz que incide sobre él y está en serie con un resistor de 10K para crear un divisor de voltaje. La salida del módulo es analógica y tiene un voltaje bajo cuando no hay luz que incida en el sensor, pero aumenta a medida que la intensidad de la luz se incrementa. Entonces, bajo una iluminación intensa, la salida se aproximará a V_{cc} , y cuando la luz esté completamente bloqueada, la salida se acercará a tierra.

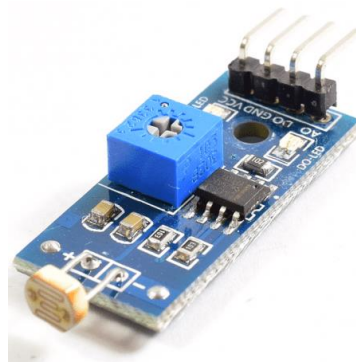
Características

- Voltaje de alimentación: de 3,3 V a 5 V
- Tipo: digitales
- Capaz de detectar el brillo ambiental y la intensidad de la luz
- Sensibilidad ajustable (a través del ajuste del potenciómetro digital azul)
- Voltaje de funcionamiento 3,3 V-5 V
- Con orificio de perno fijo para una fácil instalación
- Indicador de encendido (rojo) e indicador de salida del interruptor digital (verde)
- Dimensiones: 32 x 14 mm
- Conexión de 3 cables: VCC, GND, DO,
- Salida analógica y digital(comparador)

A continuación, la Figura 15 expone un sensor de luz LDR.

Figura 15

Sensor de luz LDR



Sensor de medición de temperatura, humedad relativa y presión barométrica

El BME280 de Bosch es un sensor de precisión utilizado en distintas aplicaciones: desde el monitoreo del clima, controles de juegos y hasta la medición de la altitud. Es fácil de

usar, está precalibrado y no requiere componentes adicionales por lo que puede comenzar a medir en poco tiempo la humedad relativa, temperatura, presión barométrica y altitud.

Capacidad del sensor BME280.

Temperatura de medición. BME280 puede medir temperaturas que oscilan entre -40 °C y 85 °C: entre 0 a 65 °C, la precisión es de $\pm 1,0$ °C; fuera de ese rango, la precisión cae a ± 1.5 °C. Se debe tener en cuenta que esta medición de temperatura se usa internamente para calibrar los sensores de presión y humedad. Además, debido a que el sensor se autocalienta, la temperatura medida suele ser un poco más alta que la real; en estos casos tiene que compararse ambas y aplicarse una compensación (Interface BME280, 2019).

Medición de la humedad. El sensor es capaz de medir la humedad relativa en un rango de 0 a 100% con una precisión de ± 3 %. Según la hoja de datos, el sensor puede medir hasta el 100% de la humedad en un rango de temperatura de 0 a 60 °C; sin embargo, la humedad máxima medible disminuye a temperaturas extremadamente altas y bajas (Interface BME280, 2019).

Presión de medición. BME280 mide la presión entre 300 hPa y 1100 hPa con una precisión absoluta de ± 1 hPa. En el rango de temperatura de 0 a 100 °C se obtiene una precisión total, lo que resulta en una precisión de medición de altitud de aproximadamente ± 1 metro. Fuera de ese rango, la precisión cae a 1,7 hPa (Interface BME280, 2019).

Cálculo de altitud/elevación. El sensor mide la presión con tal exactitud (ruido de baja altitud de 0,25 m) que también se puede utilizar como altímetro con una precisión de ± 1 metro. Además, calcula la altitud absoluta y relativa: la primera hace referencia a la altura sobre el nivel del mar (MSL), mientras que la segunda es la altura sobre el nivel del suelo (AGL). Entonces, por ejemplo, si conoce la altitud de un objeto que está sobre una mesa y lo mueve al suelo, el BME280 muestra una disminución de 2 pies en la altura; no obstante, si se busca

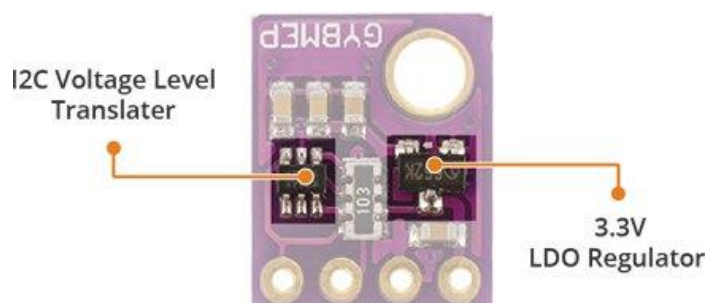
medir la altitud absoluta, las cosas se complican un poco porque el BME280 necesita tener el dato de la presión actual a nivel del mar (Interface BME280, 2019).

Propiedades de hardware del sensor BME280.

Energía. El módulo incluye un regulador LM6206 de 3,3 V integrado y un traductor de nivel de voltaje I2C, por lo que puede usarse con un microcontrolador lógico de 3,3 V o 5 V como por ejemplo Arduino. El BME280 tiene un bajo consumo energético: menos de 1 mA durante las mediciones y solo 5 μ A cuando está inactivo; es por ello que puede usarse en dispositivos que funcionan con baterías como es el caso de teléfonos, módulos GPS y relojes. La Figura 16 expone un esquema de este sensor.

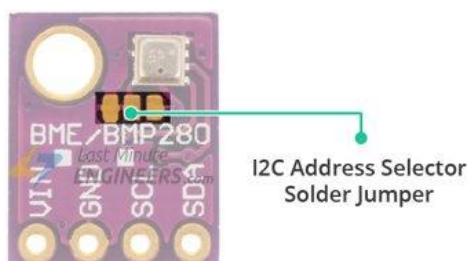
Figura 16

Sensor BME280



Nota. Sensor precalibrado que puede comenzar a medir la humedad relativa, temperatura y presión barométrica.

Interfaz I2C. El módulo BME280 se comunica a través de I2C y admite dos direcciones I2C, 0x76 y 0x77, lo que permite utilizar hasta dos sensores en el mismo bus (ver Figura 17).

Figura 17*Interfaz I2C*

Nota. La dirección I2C predeterminada del módulo es 0x76 HEX y puede cambiarse fácilmente a 0x77 HEX mediante el puente de soldadura proporcionado.

Especificaciones técnicas. A continuación, la Tabla 5 detalla los parámetros del sensor.

Tabla 5*Parámetros del sensor BME280*

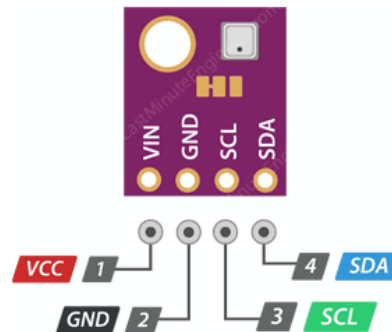
Tipo	Parámetros
Voltaje de entrada	3,3 V-5 V
Consumo de energía	1mA (típ.) y 5 μ A (inactivo)
Temperatura	-40 °C a 85 °C (\pm 1,0 °C de precisión)
Humedad	0 a 100 % de HR (\pm 3 % de precisión)
Presión	300Pa a 1100 hPa (precisión de \pm 1 hPa)
Altitud	0 a 30 000 pies (precisión de \pm 1 m)

Pines Sensor BME280. Las conexiones de pines son sencillas, pues simplemente debe conectarse el pin VCC a la salida de 5V de Arduino y el pin GND a tierra, y los pines que se usan para la comunicación I2C en las placas Arduino, SDA (línea de datos) y SCL (línea de reloj) ...

A continuación, la Figura 18 muestra un ejemplo de este pin.

Figura 18

Pin Out BME280



Nota. VIN suministra energía al módulo; conecte cualquier voltaje entre 3.3V y 5V a este pin. SCL es un pin de reloj en serie para la interfaz I2C. SDA es un pin de datos en serie para la interfaz I2C.

Sensor de calidad del aire (CO₂)

Este sensor mide los niveles de dióxido de carbono gaseoso al monitorear la cantidad de radiación infrarroja (IR) absorbida por las moléculas de CO₂, y se lo utiliza para monitorear la calidad del aire interior, vertederos, control de procesos y horticultura en ambiente controlado. El MG-811 es un sensor de óxido de metal que debe calentar el tanque al nivel especificado, genera vapor de calor y separar el CO₂ del aire (es menos sensible al alcohol y al CO); envía una señal analógica que disminuye a medida que aumenta la concentración de dióxido de carbono e incluye un pin de salida digital TTL (Robu.In, s.f.).

Características

- Amplio alcance de detección.
- Alta sensibilidad y estabilidad.
- Circuito de accionamiento sencillo.
- Salida digital: activo bajo (cuando emite una señal de bajo nivel, el indicador LED estará encendido)

- Salida analógica: dos opciones de rango: 0 a 2 V o 0 V a 4 V. El valor predeterminado es de 0 V a 2 V CC.
- Compatible con Arduino y Raspberry Pi (para usar en salida analógica con Raspberry Pi, necesitará un chip convertidor analógico a digital).
- Tamaño: 32 mm x 22 mm x 28 mm (largo x ancho x alto)
- Peso: 10g
- Tamaño: 32 mm x 22 mm x 28 mm (largo x ancho x alto)
- Peso: 10g

A continuación, la Figura 19 muestra este módulo.

Figura 19

Módulo sensor MG-811



Capítulo III

Diseño e implementación de la red de sensores IoT

Diseño y construcción de la red LoRa

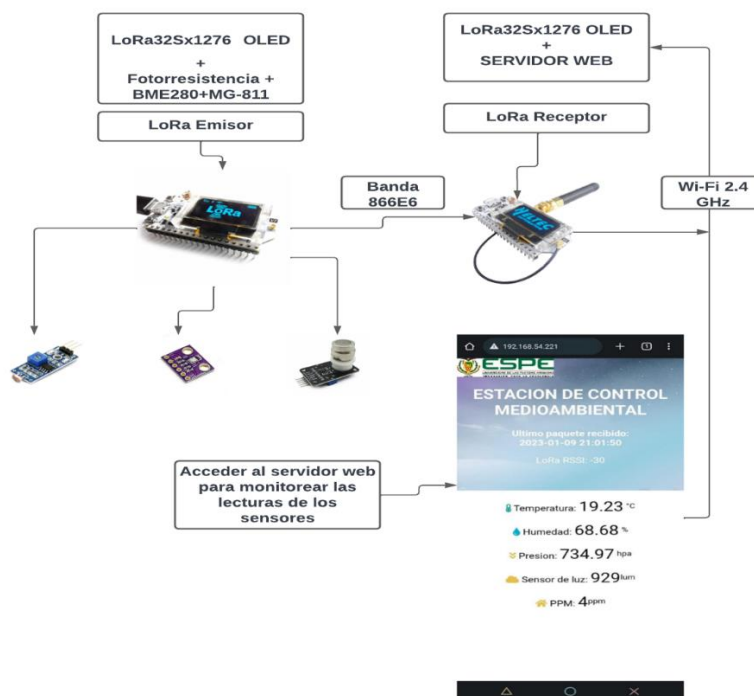
A continuación, se detalla el hardware requerido para llevar a cabo la implementación del prototipo, así como los dispositivos electrónicos con sus respectivas características. Vale mencionar que para la selección de los componentes electrónicos se tomó en cuenta que los equipos sean de uso libre y sin necesidad de una licencia. Además, son detallados a profundidad los siguientes elementos: placas Heltec WiFi LoRa 32 V2, sensor BME 280, módulo MG-811 sensor medidor de CO2 y módulo sensor de luz fotorresistencia

Diagrama de funcionamiento de la red

La Figura 20 detalla el diagrama de la red a implementar con sus respectivos componentes.

Figura 20

Diagrama de funcionamiento de la red LoRa



Como componente principal se utiliza las placas Heltec WiFi LoRa 32 V2, pues están programadas para recopilar datos medidos por los sensores de BME280 y MG-811. La placa de la Figura 20 se denomina LoRa Receptor y es la responsable de recibir las lecturas medidas por los sensores y ejecutar un servidor web asíncrono (los archivos de la página web se guardan en el sistema de archivos ESP32). Además, este receptor está programado para ejecutar el protocolo NtpClient (network time protocol) que sirve para mostrar la fecha y hora en que se recibieron las últimas lecturas.

Emisor LoRa

El emisor empleado en este prototipo es Heltec Wifi LoRa 32, que ofrece una comunicación inalámbrica de datos y modulación de radio gracias a chips transceptores. Entre las características a destacar de este tipo de tecnología es la alta inmunidad a las interferencias y el bajo consumo de energía; por lo tanto, LoRa es la mejor opción para los nodos de sensores (Riot-Os, s.f.)

Una de las características más importantes de este tipo de placa es que existen dos versiones de hardware:

- SemTech SX1278 para comunicación LoRaWAN en la banda de 433 MHz
- SemTech SX1276 para comunicación LoRaWAN en la banda de 868/915 MHz

Para este prototipo se implementa la versión SX1276 que cuenta con los siguientes componentes integrados:

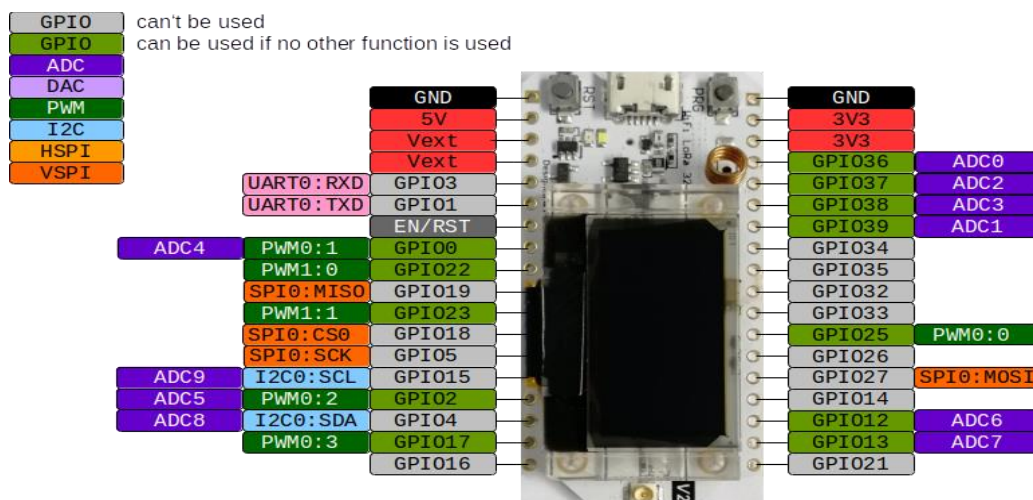
- SemTech SX1278 o SX1276 para comunicación LoRaWAN
- SSD1306 Pantalla OLED de 128x64 de 0,96 pulgadas conectada a través de I2C
- Cristal externo de 32.768 kHz para RTC

Asignación de pines de la placa

La Figura 21 muestra el pinout de la configuración por defecto definida para las placas Heltec WiFi LoRa 32 V2.

Figura 21

Diagrama de distribución de Wifi LoRa 32



Nota. Los GPIO 19, 4, 5, 14, 15, 16, 18, 32, 33, 34, 35, 21, 26 y 27 se usan para funciones de control de placa.

Los GPIO de color verde puede utilizarse para cualquier fin, mientras que los de color rojo no se aconseja usarlos a menos de que se conozca muy bien el proceso a realizar. Finalmente, los demás colores de los pines se pueden usar, pero suelen generar un efecto no deseado al momento del atranque del proyecto. Los pines empleados en la placa del proyecto son los siguientes:

- Para los módulos de emisor LoRa se usa los pines SCK 5, MISO 19, MOSI 27, SS 18, RST14, DIO0 26.
- OLED_SDA 4, OLED_SCL 15, OLED_RST 16 son designados para la pantalla OLED y sus pixeles.
- El sensor BME280 emplea los pines SDA 21 y SCL 13.
- Los pines usados para el sensor MG811 son 37, 0.99 y 100.

Configuración de la pantalla OLED

Antes de configurar la pantalla OLED es necesario añadir las librerías para la comunicación de la placa a través del protocolo Lora Wan. Para ello se utilizan las siguientes líneas de código expuestas en la Figura 22.

Figura 22

Líneas de código librerías para LoRa

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

Para llevar a cabo la conexión de Lora entre las placas es necesario la declaración de los pines SPI Lora. Cabe recalcar que SPI es un protocolo de conexión que trabaja en modo full dúplex para recibir y transmitir información, por lo que se utiliza el código de la Figura 23.

Figura 23

Función inicialización LoRa

```
//FUNCIÓN INICIALIZACIÓN LORA
void startLoRa() {
//PINES SPI LORA
  SPI.begin(SCK, MISO, MOSI, SS);
```

Nota. Si está utilizando otra placa LoRa verifique los pines utilizados por el chip transceptor LoRa.

A continuación, se muestra la configuración Lora en la Figura 24.

Figura 24

Configuración LoRa

```
//CONFIGURACIÓN LORA
LoRa.setPins(SS, RST, DIO0);

while (!LoRa.begin(BAND) && counter < 10) {
  Serial.print(".");
  counter++;
  delay(500);
}
if (counter == 10) {
  readingID++;
  Serial.println("¡ERROR AL INICIAR LORA!");
}
```

Una vez realizadas las configuraciones es ideal programar un mensaje de advertencia en caso de que exista algún error al momento de iniciar la función Lora.

Por otro lado, con respecto a la pantalla OLED es de 0,96 in con 128x64 píxeles; utiliza el controlador SSD1306 y se conecta a través de I2C_DEV(0). En este caso se usa las librerías expuestas en la Figura 25.

Figura 25

Librerías para la pantalla OLED

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

Para la declaración de pines de la pantalla OLED son empleados los pines 4,15,16 como se presenta en la Figura 26.

Figura 26

Declaración de pines y pixeles pantalla OLED

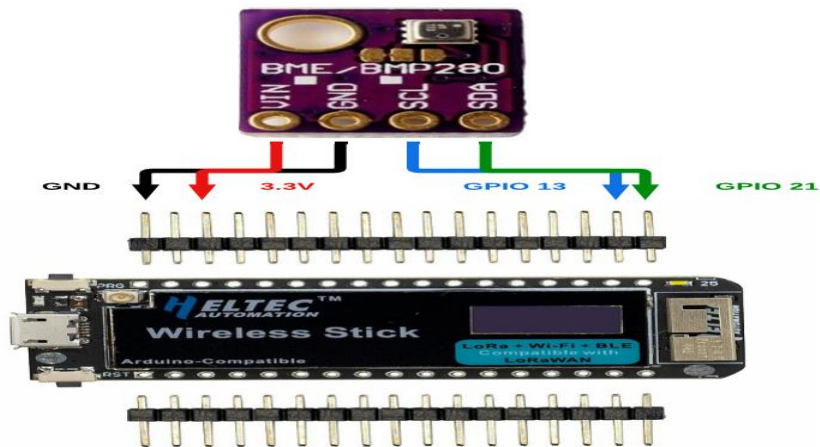
```
#define OLED_SDA 4
#define OLED_SCL 15
#define OLED_RST 16
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
```

Configuración del sensor BME280

El BME280 se caracteriza y distingue de otros modelos porque integra 3 sensores en un mismo encapsulado: de humedad relativa, temperatura y presión atmosférica. Es un sensor extremadamente pequeño, diseñado para cumplir con un consumo de energía relativamente bajo y capaz de ofrecer mediciones estables a largo plazo (puede ser empleado en aplicaciones móviles). Mide la presión barométrica con ± 1 hPa (hectopascal) de precisión; la humedad, con una precisión de ± 3 ; la temperatura, con una precisión de $\pm 1.0^{\circ}\text{C}$ (All Data Sheet, s.f.) (ver Figura 27).

Figura 27

Circuito LoRa emisor y BME280



Se designa los pines con lo que se conecta el sensor BME280 a la placa LoRa Emisor. Para programar el sensor BME280, mediante Arduino IDE, tienen que declararse las librerías necesarias tal como lo detalla la Figura 28.

Figura 28

Declaración de librerías BME280

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

Simultáneamente tienen que declararse los pines a los que está conectado el sensor BME280 con la placa Heltec WiFi LoRa 32 V2 Emisor (Figura 29).

Figura 29

Declaración de pines sensor MBE280

```
#define SDA 21
#define SCL 13
```

Para el circuito de Emisor LoRa y el BME280 se utiliza el protocolo de comunicación I2C, que es de comunicación serial y define la trama de datos y conexiones físicas para transferir bits entre dos dispositivos digitales. Este protocolo se caracteriza por hacer uso de las líneas SCL y SDA para monitorear otros dispositivos: la primera es línea de reloj en serie y la segunda es el puerto de aceptación de línea de datos.

Una vez agregadas las librerías del sensor BME280 y declarados los pines a los que está conectado a la placa Heltec WiFi LoRa 32 V2, se procede a crear una instancia I2C para el sensor BME280 mediante el código de la Figura 30.

Figura 30

Función inicialización BME280

```
void startBME(){
  I2Cone.begin(SDA, SCL, 100000);
  bool status1 = bme.begin(0x76, &I2Cone);
  if (!status1) {
    Serial.println("NO SE PUDO ENCONTRAR EL SENSOR BME280, VERIFIQUE EL CABLEADO");
    while (1);
  }
}
```

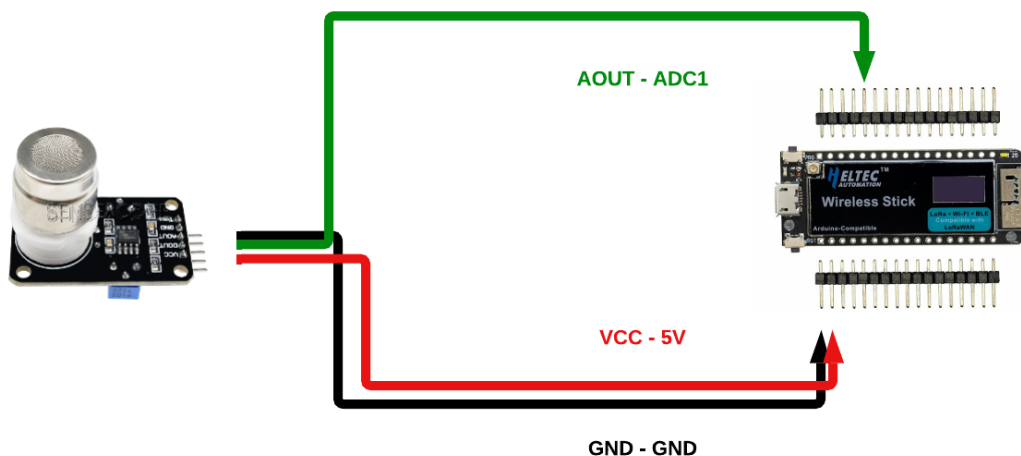
Configuración del sensor MG811

El MG-811 es un sensor analógico, uno de los más sensibles al CO₂, y puede medir en cualquier microcontrolador o tarjeta de desarrollo que cuente con ADC (convertidor de analógico - digital). Utiliza un sistema de 5v, cuenta con un potenciómetro para establecer el punto de corte o alarma en la lectura del CO₂ y un dato interesante es que su voltaje de salida es inversamente proporcional a la concentración de CO₂; es decir, disminuye a medida que aumenta la cantidad de dióxido de carbono en el ambiente. Por esa razón, este tipo de módulo es ideal para llevar un control en los procesos de fermentación y de la calidad del aire en espacios cerrados (Winsen, s.f.).

Para efectos del prototipo, la conexión a la placa es muy básica: basta con conectarlo en los GPIO indicados y leer la señal analógica con los convertidores ADC (Figura 31).

Figura 31

Circuito LoRa emisor y MG-811



Una vez conectado el MG-811, es necesario desarrollar un programa capaz de leer la salida del sensor. Entonces, primero se incluyen en Arduino IDE las librerías necesarias para el sensor MG-811 y luego deben declararse los pines a los que está conectado el sensor a la placa LoRa Emisor (Figura 32 y 33).

Figura 32*Librerías para el sensor MG-811*

```
//LIBRERÍAS PARA EL SENSOR MG811
#include "CO2Sensor.h"
```

Figura 33*Declaración de pines sensor MG-811*

```
//DECLARACIÓN DE PINES SENSOR MG811
CO2Sensor co2Sensor(37, 0.99, 100);
```

Configuraciones de lecturas

A continuación, son detalladas las configuraciones de lecturas. Aquí es necesario crear variables para contener el mensaje de LoRa, temperatura, humedad, presión, CO₂, intensidad lumínica e ID de lectura (Figura 34).

Figura 34*Declaración de variables de lectura*

```
//VARIABLE MENSAJE
String LoRaMessage = "";

//VARIABLES LECTURA SENSOR
float temperature = 0;
float humidity = 0;
float pressure = 0;
int adc = 0;
int readadc = 0;
int ppm = 0;
```

Tras declarar las variables de lectura se debe crear un bme (mostrar objeto) para la pantalla OLED tal como lo indica la Figura 35.

Figura 35*Crear bme*

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RST);
```

Luego hay que llamar a las funciones creadas previamente para inicializar la pantalla OLED, BME280, MG-811 y el módulo transmisor LoRa. Para ello se emplean las líneas de código de la Figura 36.

Figura 36

Inicialización monitor serie

```
//INICIALIZACIÓN MONITOR SERIE
Serial.begin(115200);
startAdc();
startOLED();
startBME();
startLoRA();
}

void loop() {
  getReadings();
  sendReadings();
  delay(1000);
}
```

Nota. El circuito se repite cada 1000 milisegundos (10 segundos). Las lecturas de los sensores se actualizaras cada 10 segundos.

Para obtener las lecturas de los sensores solo hay que llamar a las variables antes declaras dentro de bme (Figura 37).

Figura 37

Función obtener lecturas

```
//FUNCIÓN OBTENER LECTURAS
void getReadings() {
  temperature = bme.readTemperature();
  humidity = bme.readHumidity();
  pressure = bme.readPressure() / 100.0F;
  adc = analogRead(36);
  adc = map(adc, 1023, 0, 0,1023);
  ppm = co2Sensor.read();
}
```

Una vez obtenidas las lecturas de los sensores es esencial programar la placa LoRa Emisor que se encarga de enviar las lecturas de los sensores conectados (Figura 38).

Figura 38

Función enviar lecturas

```
//FUNCIÓN ENVIAR LECTURAS
void sendReadings() {
  LoRaMessage = String(readingID) + "/" + String(temperature) + "&"
  + String(humidity) +
  "#" + String(pressure) + "$" + String(adc) + "+" + String(ppm);
  LoRa.beginPacket();
  LoRa.print(LoRaMessage);
  LoRa.endPacket();

  display.clearDisplay();
  display.setCursor(0,0);
  display.setTextSize(1);
}
```

Nota. Tome en consideración que cada lectura se separa por un carácter especial y que el receptor puede identificar fácilmente cada valor.

Módulo fotorresistencia LDR

Este tipo de sensor, analógico y digital, es capaz de recopilar datos sobre la intensidad lumínica emitida sobre la superficie del panel sea por la luz de día, la de una lámpara, foco e incluso algún LED. El proceso de obtención de lecturas es simple, pues la resistencia del sensor varía en función de la cantidad de luz a la que es expuesto, lo que genera una variación de voltaje en su salida (Cebek, s.f.).

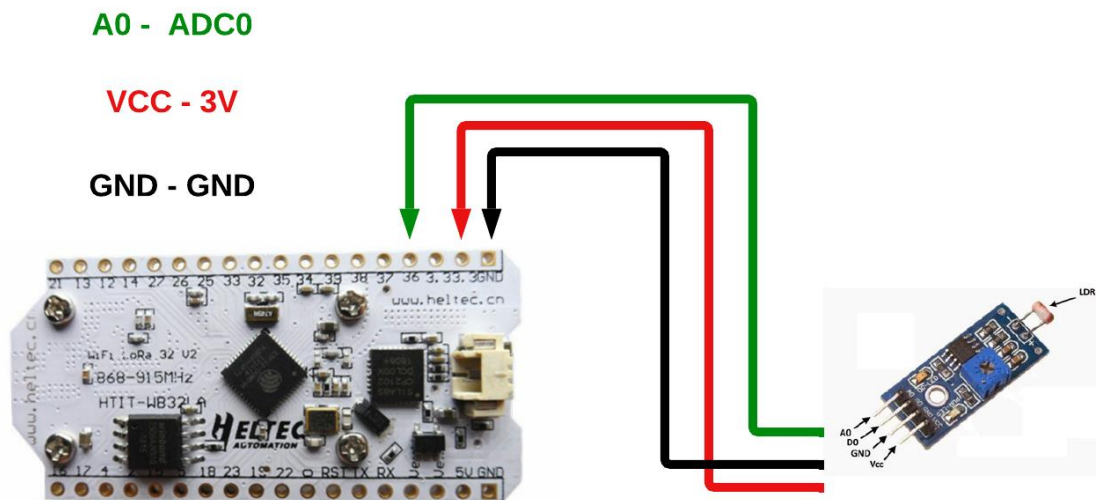
Especificaciones del módulo sensor fotorresistencia LDR.

- Voltaje de alimentación: 3.3 – 5V
- Salida: Digital y Analógica
- Sensibilidad ajustable con potenciómetro
- Número de pines: 3 (Vcc, GND & Signal)

El circuito entre placa y la fotorresistencia es muy simple, solo es necesario conectarlo al voltaje indicado en las especificaciones y al GPIO capaz de leer la señal analógica con los convertidores ADC (Figura 39).

Figura 39

Circuito Emisor LoRa-Fotorresistencia



Receptor LoRa

La placa designada para cumplir la función de receptor recibe los paquetes entrantes y muestra las lecturas de los sensores en un servidor web, en donde además se detalla la última vez en que fueron recibidas las lecturas y el indicador de intensidad de la señal recibida (RSSI). La Figura 40 expone el servidor web a construir.

Figura 40

Servidor web



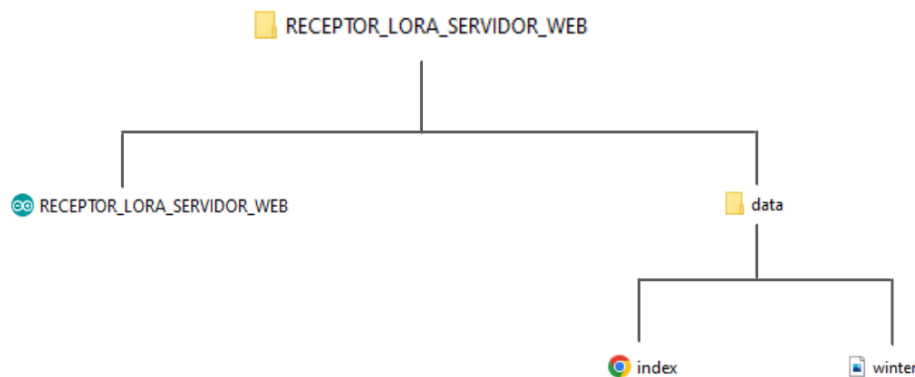
El servidor web cuenta con una imagen de fondo. Existen varias formas de mostrar imágenes en un servidor web ESP32 y en formato PNG o JPG; pero sea cual fuere, se almacenan en el sistema de archivos SPIFFS. Para insertar la imagen en un servidor web es necesario utilizar la biblioteca ESPAsyncWebServer en un servidor HTTP simple.

Organizar los archivos

Para construir el servidor web es necesario tres archivos diferentes: el código de Arduino IDE, el archivo HTML y la imagen. Todos deben guardarse en la carpeta llamada datos que está almacenada en la carpeta del código de Arduino como se muestra en la Figura 41.

Figura 41

Diagrama de archivos



Desarrollo HTML

Se crea el archivo HTML en el que deben incluirse los estilos CSS, y de igual forma se agregan unas líneas de JavaScript cuyo fin es actualizar las lecturas de los sensores automáticamente. Es de suma importancia tener en consideración los marcadores de posición que van colocados entre signos % como lo muestra la Figura 42.

Figura 42

Marcadores de posición

```

<p><strong>Ultimo paquete recibido:<br/><span id="timestamp">%TIMESTAMP%</span></strong></p>
<p>LoRa RSSI: <span id="rssi">%RSSI%</span></p>
</header>
<main>
<p>
<i class="fas fa-thermometer-half" style="color:#059e8a;"></i> Temperatura: <span id="temperature" class="readings">%TEMPERATURE%</span>
<sup>&deg;C</sup>
</p>
<p>
<i class="fas fa-tint" style="color:#00add6;"></i> Humedad: <span id="humidity" class="readings">%HUMIDITY%</span>
<sup>&#37;</sup>
</p>
<p>
<i class="fas fa-angle-double-down" style="color:#e8c14d;"></i> Presion: <span id="pressure" class="readings">%PRESSURE%</span>
<sup>hpa</sup>
</p>
<p>
<i class="fas fa-cloud" style="color:#e8c14d;"></i> Sensor de luz: <span id="light" class="readings">%LIGHT%</span><sup>lum</sup>
</p>
<p>
<i class="fas fa-home" style="color:#e8c14d;"></i> PPM: <span id="ppm" class="readings">%PPM%</span><sup>ppm</sup>
</p>
</main>

```

Los marcadores de posición son posteriormente reemplazados por los valores reales de los sensores. Por otra parte, los estilos se agregan entre los `<style>` y `</style>` como se expone en la Figura 43.

Figura 43

Agregar estilos

```
<style>
  body {
    margin: 0;
    font-family: Arial, Helvetica, sans-serif;
    text-align: center;
  }
  header {
    margin: 0;
    padding-top: 5vh;
    padding-bottom: 5vh;
    overflow: hidden;
    background-image: url(winter);
    background-size: cover;
    color: white;
  }
  h2 {
    font-size: 2.0rem;
  }
  p { font-size: 1.2rem; }
  .units { font-size: 1.2rem; }
  .readings { font-size: 2.0rem; }
</style>
```

Si es necesario cambiar la imagen, solo debe modificarse la siguiente línea de código `background-image: url(winter)`. Por otro lado, el JavaScript encargado de actualizar las lecturas de los sensores va entre `<script>` y `</script>` como es detallado en la Figura 44.

Figura 44

Código Java Scrip

```

<script>
setInterval(updateValues, 10000, "temperature");
setInterval(updateValues, 10000, "humidity");
setInterval(updateValues, 10000, "pressure");
setInterval(updateValues, 10000, "rssi");
setInterval(updateValues, 10000, "timestamp");
setInterval(updateValues, 10000, "light");
setInterval(updateValues, 10000, "ppm");

function updateValues(value) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById(value).innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/" + value, true);
    xhttp.send();
}
</script>

```

Código de Arduino del receptor LoRa

Para desarrollar el código del receptor LoRa hay que incluir las bibliotecas necesarias y seguir los siguientes pasos:

- Construir el servidor web asíncrono;
- Acceder al sistema de archivos ESP32 (SPIFFS);
- Comunicarse con el chip LoRa;
- Controlar la pantalla OLED;
- Obtener fecha y hora de un servidor NTP.

Lo primero a realizar al momento de desarrollar el código de Arduino IDE para el receptor LoRa es incluir las librerías necesarias (ver Figura 45).

Figura 45

Librerías receptor LoRa

```

//LIBRERIAS PARA WIFI
#include <WiFi.h>
#include "ESPAsyncWebServer.h"

#include <SPIFFS.h>

//LIBRERIAS PARA LORA
#include <SPI.h>
#include <LoRa.h>

//LIBRERÍAS PARA LA PANTALLA OLED
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//LIBRERÍAS PARA OBTENER LA HORA DEL SERVIDOR NTP
#include <NTPClient.h>
#include <WiFiUDP.h>

```

Una vez agregadas las librerías hay que conectar a los dos módulos LoRa con la declaración de los pines que utiliza el chip LoRa (Figura 46).

Figura 46

Declaración de pines receptor LoRa

```

//DECLARACIÓN DE PINES MÓDULO RECEPTOR LORA
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DIO0 26

```

Nota. si desea conectar otra placa a la red debe verificar los pines que utilizan los chips LoRa.

A continuación, es establecida la frecuencia por la que se comunican los módulos LoRa (Figura 47).

Figura 47*Declaración de frecuencia*

```
//BANDA
#define BAND 866E6
//#define BAND 914
```

Posterior a ello hay que configurar los pines y declarar los pixeles para la pantalla OLED. Es la misma configuración y declaraciones de pixeles usadas para la pantalla OLED del emisor LoRa (Figura 48).

Figura 48*Declaración de pines y pixeles para el receptor LoRa*

```
//DECLARACIÓN DE PINES Y PÍXELES PANTALLA OLED
#define OLED_SDA 4
#define OLED_SCL 15
#define OLED_RST 16
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
```

A continuación, hay que ingresar la SSID de la red y la contraseña que permitirá al ESP32 conectarse a la red local. Esto garantiza la conexión entre el Gateway y el servidor web (Figura 49).

Figura 49*Declaración de red local*

```
//RED Y CONTRASEÑA WIFI
const char* ssid = "Lab_Com";
const char* password = "1234567890987654321";
```

Simultáneamente se define el protocolo NTP Client que es necesario para obtener la fecha y hora. De igual forma deben declararse las variables para guardar estos datos (Figura 50).

Figura 50

Declaración de NTP Client

```
//DEFINICIÓN CLIENTE NTP PARA OBTENER EL TIEMPO
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

//VARIABLES FECHA Y HORA
String formattedDate;
String day;
String hour;
String timestamp;
```

También es esencial declarar las variables responsables de obtener y almacenar las lecturas de los sensores recibidas a través de la radio LoRa (Figura 51).

Figura 51

Variables para obtener y guardar datos

```
//INICIALIZACIÓN DE VARIABLES PARA OBTENER Y GUARDAR DATOS
int rssi;
String loRaMessage;
String temperature;
String humidity;
String pressure;
String readingID;
String light;
String ppm;
```

Mediante la librería AsyncWebServer es creado un objeto, llamado server, en el puerto 80 mediante el comando processor (). Se detalla la configuración necesaria para que cumpla con la función de atribuir valores a los marcadores de posición que anteriormente fueron creados en el archivo HTML. Es decir, acepta como argumento el marcador de posición y se encarga de devolver un string que lo reemplaza.

En pocas palabras, si encuentra el marcador de posición llamado TEMPERATURE devuelve la variable de cadena; en este caso sería temperatura, como se muestra en las siguientes líneas de código de la Figura 52.

Figura 52

Proceso marcador de posición

```

AsyncWebServer server(80);

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RST);

String processor(const String& var){
  //Serial.println(var);
  if(var == "TEMPERATURE"){
    return temperature;
  }
  else if(var == "HUMIDITY"){
    return humidity;
  }
  else if(var == "PRESSURE"){
    return pressure;
  }
  else if(var == "TIMESTAMP"){
    return timestamp;
  }
  else if (var == "RRSI"){
    return String(rssi);
  }
  else if (var == "LIGTH"){
    return String(light);
  }
  else if (var == "PPM"){
    return String(ppm);
  }
  return String();
}

```

Para habilitar la comunicación LoRa y la conexión a Wifi hay que inicializar el monitor serie e y el SPIFFS, que es un sistema de archivos para las memorias SPI Flash NOR que utiliza el ESP32 (encargado de guardar los archivos en la memoria flash ESP32). La Figura 53 expone el código.

Figura 53

Inicialización del monitor serie y SPIFFS

```

//INICIALIZACIÓN MONITOR SERIE
Serial.begin(115200);
startOLED();
startLoRA();
connectWiFi();

if(!SPIFFS.begin()){
  Serial.println("SE HA PRODUCIDO UN ERROR AL MONTAR SPIFFS");
  return;
}

```

Una vez inicializado el monitor serie es necesario programar el servidor web. Cabe recalcar que la librería ESPAsyncWebServer ofrece la configuración de las rutas en las que el servidor se encarga de recibir las solicitudes HTTP entrantes. En pocas palabras, cuando se recibe una solicitud en la URL de la ruta se envía el archivo HTML almacenado en la placa ESP32 a través del sistema SPIFFS (Figura 54).

Figura 54

Configuración de rutas

```
server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send(SPIFFS, "/index.html", String(), false, processor);
});
server.on("/temperature", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", temperature.c_str());
});
server.on("/humidity", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", humidity.c_str());
});
server.on("/pressure", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", pressure.c_str());
});
server.on("/timestamp", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", timestamp.c_str());
});
server.on("/rssi", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(rssi).c_str());
});
server.on("/light", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(light).c_str());
});
server.on("/ppm", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(ppm).c_str());
});
server.on("/winter", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send(SPIFFS, "/winter.jpg", "image/jpg");
});
```

Después de configurar las rutas se procede a iniciar la pantalla OLED, que es la responsable de asignar la dirección IP y da acceso al servidor web de la red local a la que está conectada la placa receptor LoRa (Figura 55).

Figura 55

Inicialización de pantalla OLED receptor LoRa

```
//FUNCIÓN INICIALIZACIÓN PANTALLA OLED
void startOLED(){
//RESET PANTALLA OLED POR SOFTWARE
  pinMode(OLED_RST, OUTPUT);
  digitalWrite(OLED_RST, LOW);
  delay(20);
  digitalWrite(OLED_RST, HIGH);

//INICIALIZACIÓN PANTALLA OLED
  Wire.begin(OLED_SDA, OLED_SCL);
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3c, false, false)) {
    Serial.println(F("SSD1306 FALLO DE ASIGNACION"));
    for(;;);
  }

  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(0,0);
  display.print("RECEPTOR LORA");
}
```

La función getLoRaData() es la responsable de leer y obtener los paquetes enviados por el receptor LoRa al servidor web. Su configuración está indicada en la Figura 56.

Figura 56

Función leer y obtener paquete LoRa

```
//FUNCIÓN LEER Y OBTENER EL PAQUETE LORA
void getLoRaData() {
  Serial.print("PAQUETE LORA RECIBIDO: ");

//LEER PAQUETE
  while (LoRa.available()) {
    String LoRaData = LoRa.readString();
    Serial.print(LoRaData);
  }

//OBTENER EL ID DE LECTURA
  int pos1 = LoRaData.indexOf('/');
  int pos2 = LoRaData.indexOf('&');
  int pos3 = LoRaData.indexOf('#');
  int pos4 = LoRaData.indexOf('$');
  int pos5 = LoRaData.indexOf('+');
  readingID = LoRaData.substring(0, pos1);
  temperature = LoRaData.substring(pos1 +1, pos2);
  humidity = LoRaData.substring(pos2+1, pos3);
  pressure = LoRaData.substring(pos3+1, pos4);
  light = LoRaData.substring(pos4+1, pos5);
  ppm = LoRaData.substring(pos5+1, LoRaData.length());
}
```

Por otra parte, la función `getTimeStamp()` es la encargada de obtener la hora y fecha de Internet en el momento en el que se recibe el paquete del receptor LoRa. Una vez declarada la función `getTimeStamp()` hay que declarar el formato de fecha (Figura 57).

Figura 57

Declaración del protocolo NTP client

```
//FUNCIÓN PARA OBTENER FECHA Y HORA DESDE NTPCLIENT
void getTimeStamp() {
    while(!timeClient.update()) {
        timeClient.forceUpdate();
    }

    //FORMATO FECHA
    //2018-05-28T16:00:13Z
    formattedDate = timeClient.getFormattedDate();
    Serial.println(formattedDate);

    //EXTRAER FECHA
    int splitT = formattedDate.indexOf("T");
    day = formattedDate.substring(0, splitT);
    Serial.println(day);

    //EXTRAER HORA
    hour = formattedDate.substring(splitT+1, formattedDate.length()-1);
    Serial.println(hour);
    timestamp = day;
    timestamp = day + " " + hour;
}

```

El último paso es iniciar el servidor web, el protocolo NTP client y corroborar si existen paquetes LoRa disponibles (Figura 58).

Figura 58

Inicialización del servidor web

```
//EMPEZAR SERVIDOR
server.begin();

//INICIALIZACIÓN NTPCLIENTS PARA OBTENER TIEMPO
timeClient.begin();
timeClient.setTimeOffset(-18000);
}

void loop() {
//COMPROBAR SI HAY PAQUETES LORA DISPONIBLES
int packetSize = LoRa.parsePacket();

if (packetSize) {
    getLoRaData();
    getTimeStamp();
}
}

```

Análisis de resultados

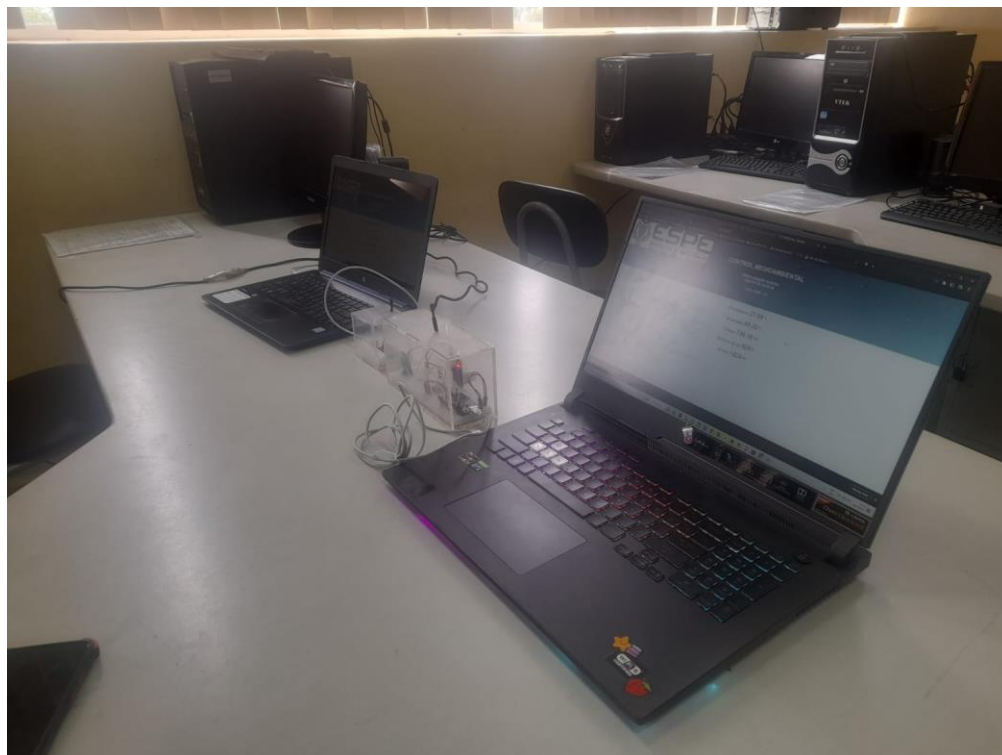
En este capítulo es expuesto el análisis de los resultados obtenidos de las mediciones de los sensores BME280 (temperatura, humedad, presión), MG.811 (Co2) y fotorresistencia (detector de luz). Tras realizar las pruebas necesarias, el prototipo cumplió con los objetivos planteados.

Interpretación de variables

Para medir los valores máximos en diferentes locaciones, horarios y condiciones se contó con un nodo sensores y un coordinador que formaron una red de sensores inalámbrica punto-punto (Figura 59).

Figura 59

Disposición de la red de sensores entre nodo-coordinador



Variables del sensor bme280

La Figura 60, 61 y 62 exponen las variables recopiladas de este sensor.

Figura 60

Variables sensor MBE280 en el laboratorio de Sistemas de Comunicación de la carrera de Tecnología Superior en Redes y Telecomunicaciones

🌡 Temperatura: 17.05 °C
💧 Humedad: 70.65 %
📉 Presion: 732.88 hpa

Figura 61

Variables sensor MBE280 en el Departamento de Eléctrica y Electrónica

🌡 Temperatura: 21.18 °C
💧 Humedad: 61.84 %
📉 Presion: 739.21 hpa

Figura 62

Variables sensor MBE280 en exteriores

🌡 Temperatura: 32.25 °C
💧 Humedad: 36.87 %
📉 Presion: 737.75 hpa

Variables del MG-811

La Figura 63, 64 y 65 exponen los datos obtenidos por este sensor.

Figura 63

Variable sensor MG-811 en el laboratorio de Sistemas de Comunicación de la carrera de Tecnología Superior en Redes y Telecomunicaciones

 PPM: 159^{ppm}


Figura 64

Variable sensor MG-811 en el Departamento de Eléctrica y Electrónica

 PPM: 224^{ppm}

Figura 65

Variable sensor MG-811 en exteriores

 PPM: 1002^{ppm}

Variables de la fotorresistencia

La variable de este sensor se expone en la Figura 65, 66 y 67.

Figura 66

Variable sensor fotorresistencia LDR en el laboratorio de Sistemas de Comunicación de la carrera de Tecnología Superior en Redes y Telecomunicaciones


 Sensor de luz: 911^{lum}

Figura 67

Variable sensor fotorresistencia LDR en el Departamento de Eléctrica y Electrónica

☁ Sensor de luz: 918^{lum}

Figura 68

Variable sensor MG-811 en el Departamento de Eléctrica y Electrónica

☁ Sensor de luz: 927^{lum}

Distancia Máxima

La Figura 69 y 70 exponen la distancia máxima.

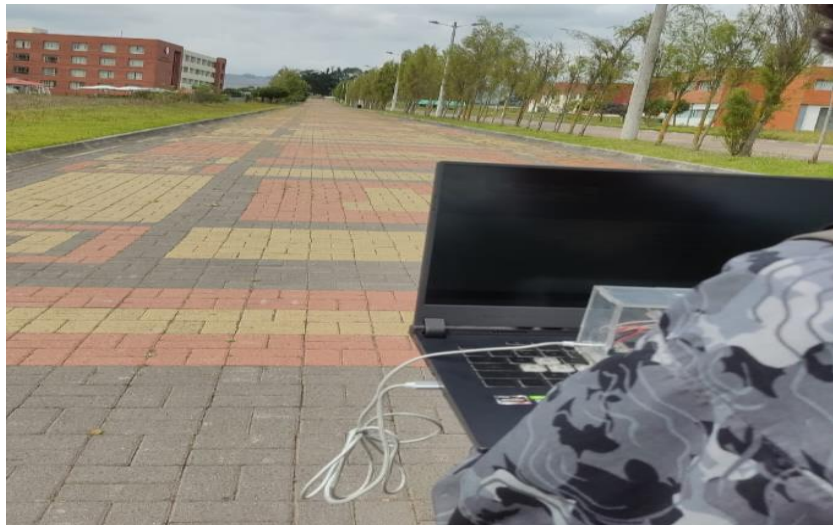
Figura 69

Nodo-coordinador punto A



Figura 70

Nodo-emisor punto B



Capítulo IV

Conclusiones y Recomendaciones

Conclusiones

- Se realizó una investigación minuciosa de los diversos tipos de redes IoT, la utilización del estándar LoRaWAN y los dispositivos y sensores a utilizar en la red, evaluando su eficiencia, alcance y seguridad, así como también, las especificaciones técnicas de cada una, cubriendo de esta manera los requerimientos para el proyecto, incluyendo su funcionamiento, precisión y costo.
- Para determinar el sistema de adquisición de datos, se ha utilizado la combinación de tecnología LoRa, los de sensores IoT y una la interfaz web; que proporcionan una solución completa y fácil de usar para el control medioambiental en interiores ya que permite monitorear las variables en tiempo real, mejorando la capacidad de tomar decisiones y gestionar los entornos de manera efectiva.

- Se ha implementado un prototipo de una red LoRa con sensores para control medioambiental en interiores, el cual cuenta con un sistema efectivo de adquisición de datos para la obtención de parámetros importantes para el control medioambiental, tales como temperatura, CO₂, humedad relativa, presión barométrica e intensidad luminosa ambiental en diferentes locaciones. Este sistema permite una monitorización precisa y en tiempo real de los ambientes, lo que es esencial para garantizar un entorno saludable y confortable.
- Mediante pruebas se determinó una correcta recepción de datos alcanzado con éxito al obtener un enlace mayor a 500 metros. Este resultado demuestra la capacidad de la tecnología para transmitir información de manera confiable y eficiente a largas distancias, lo que aumenta la precisión y confiabilidad de los datos recopilados por los sensores.
- Se llevó a cabo el desarrollo de una red de monitoreo de variables ambientales en interiores mediante sensores IoT, tecnología LoRa y una interfaz gráfica que permite visualizar a detalle cada uno de los datos recopilados.

Recomendaciones

- Se recomienda analizar cuidadosamente los requerimientos del proyecto como la eficiencia, el alcance, la seguridad y sobre todo el costo de los dispositivos y sensores para determinar si es viable la implementación del estándar LoRaWAN, y poder garantizar una red IoT eficiente y efectiva.
- Antes de la implementación, analizar a detalle cada una de las especificaciones técnicas de los sensores y dispositivos a utilizar, para asegurarse de que cumplan con las necesidades y requerimientos del proyecto.
- Una vez implementado la red, se recomienda considerar el mantenimiento y actualización periódica del sistema para garantizar su eficiencia y precisión a largo plazo.
- Es importante considerar la expansión de la red y la integración de nuevos sensores y dispositivos para aumentar la capacidad de monitoreo y mejorar la precisión de los datos, ya que si se desea agregar nuevos dispositivos a la red es de suma importancia analizar la programación realizada en Arduino IDE con el objetivo de verificar qué pines y frecuencia emplean las placas para su comunicación. Además, se sugiere evaluar la implementación de medidas de seguridad para proteger la privacidad y los datos sensibles.
- Analizar con mayor detenimiento los factores antes mencionados, como la calidad de la señal, la interferencia ambiental y la capacidad de transmisión de datos, al evaluar y planificar la implementación de la red LoRa para control medioambiental en interiores.
- Una vez encendido los nodos es recomendable esperar un tiempo considerable para lograr lecturas confiables, pues los sensores tienen un margen de error que disminuye conforme pasa el tiempo de encendido.

Bibliografía

- Garbarino, J. (2012). *Protocolos Para Redes Inalámbricas de Sensores*. Editorial Academica Espanola.
- Guijo, J. A. E. (2021). *Sistema de monitorización y auxilio para personas con paraplejia basado en redes LoraWan*. Editorial Inclusión.
- Interface BME280 Temperature, Humidity & Pressure Sensor with Arduino*. (2019, enero 4). Last Minute Engineers. <https://lastminuteengineers.com/bme280-arduino-tutorial/>
- Perez,2014. (s. f.). Recuperado 28 de diciembre de 2022, de <http://ve.scielo.org/pdf/uct/v18n70/art02.pdf>
- Rocha, Á., López-López, P. C., & Salgado-Guerrero, J. P. (2021). *Communication, Smart Technologies and Innovation for Society: Proceedings of CITIS 2021*. Springer Nature.
- Stefany, N. N. K. (2020). *Diseño y desarrollo de un prototipo de red de sensores IOT utilizando tecnología Lorawan para el monitoreo de parámetros ambientales en interiores y exteriores*. 121.
- All Data Sheet. (s.f.). *BME280 Datasheet (PDF) - Bosch Sensortec GmbH*. <https://www.alldatasheet.com/datasheet-pdf/pdf/1132060/BOSCH/BME280.html>
- Alonso, H. (s.f.). *LoRaWAN. Arquitectura de red*. https://elbosquedesilicio.es/wp-content/uploads/2020/05/LoRaWAN_3-1.pdf
- Aprendiendo Arduino. (30 de enero de 2020). *Itinerario Formación IoT/Industria 4.0*. <https://www.aprendiendoarduino.com/tag/sigfox/>
- Arduino. (5 de febrero de 2018). *What is Arduino?* Recuperado el 10 de enero de 2023 de <https://www.arduino.cc/en/Guide/Introduction>
- Bertoleti, P. N(2019). *Proyectos com ESP32 y LoRa*. Editora NCB.

- Cantuña, K. y Satián, D. (2019). *Análisis de rendimiento de la tecnología lorawan aplicado al desarrollo de un sistema de monitoreo de calidad de aire en la unidad educativa Santo Tomás Apóstol Riobamba* [Trabajo de pregrado, Escuela Superior Politécnica de Chimborazo]. <http://dspace.esPOCH.edu.ec/bitstream/123456789/11026/1/98T00246.pdf>
- Cebek. (s.f.). *Fotorresistencia LDR 4.3 mm*. <https://www.electan.com/datasheets/cebek/CE-C2795.pdf>
- Cedeño, C. y Quiroz, M. (2015). *Control de acceso por medio de cámaras y sensores en la empresa Melproyect de la ciudad Portoviejo* [Trabajo de pregrado, Escuela Superior Politécnica Agropecuaria De Manabí]. <https://repositorio.espam.edu.ec/bitstream/42000/336/1/TC82.pdf>
- Chapter,2019. (s. f.). Recuperado 28 de julio de 2022, de <http://documentacion.ideam.gov.co/openbiblio/bvirtual/001083/Course2/Lecturas/Vehiculos/chapter2.pdf>
- Chivita, Y. (mayo 2020). *Internet of Things (IoT). Diseño de una red de IoT para el hogar*. Seminario de Redes y Telecomunicaciones, Universidad Cooperativa de Colombia, Bogotá, Colombia. <https://repository.ucc.edu.co/server/api/core/bitstreams/197b735f-90fd-44b7-80b8-aceb440afbd8/content>
- Cobo, L., Hurtado, D. y Vengoechea, J. (2020). *Gestión de la calidad del servicio para redes de sensores multimedia inalámbrico*. Lulu Press.
- Córdoba, E. (2017). *Análisis y diseño de una red de sensores en un parque natural* [Trabajo de pregrado, Universidad Abierta de Cataluña]. <https://openaccess.uoc.edu/bitstream/10609/59925/3/ecordobapTFG0117mem%C3%B2ria.pdf>
- Delgado, F., Navarro, J., Chinchilla, N. y Ramos, J. (27 a 29 de octubre de 2021). *Arquitectura LoRaWAN para entornos sin cobertura*. 4. XV Jornadas de Ingeniería Telemática,

Universidad de A Coruña, España.

<https://digibug.ugr.es/bitstream/handle/10481/71143/actasJITEL21%20Arquitectura%20LoRaWAN%20para%20entornos%20sin%20cobertura.pdf?sequence=1&isAllowed=y>

Forero, H. y Camberos, D. (2020). *Desarrollo de una red punto a punto con tecnología LoRa y servidor web* [Trabajo de pregrado, Universidad Cooperativa de Colombia].

<https://repository.ucc.edu.co/items/1db5d9e2-de22-4399-a41a-8088383e4a7c>

HiTech Chain. (s. f.). *Wifi Lora 32 V2 SX1278 0.96 inch OLED Display ESP32 LoRa dev- board with antenna for iot. HiTechChain*. Recuperado el 25 de enero de 2023 de

<https://hitechchain.se/en/iot/wifi-lora-32-v2>

Joyanes, L. (2021). *Internet de las cosas: Un futuro hiperconectado: 5G, inteligencia artificial, Big Data, Cloud, Blockchain y ciberseguridad*. Marcombo.

Last Minute Engineers. (s.f.). *Interface BME280 Temperature, Humidity & Pressure Sensor with Arduino*. <https://lastminuteengineers.com/bme280-arduino-tutorial/>

Lizeth, C. E. K., & Fernanda, S. G. D. (2019). INGENIERA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES. 144.

Narváez, S. y Contreras, V. (2020). *Diseño y desarrollo de un prototipo de red de sensores IOT utilizando tecnología Lorawan para el monitoreo de parámetros ambientales en interiores y exteriores* [Trabajo de pregrado, Universidad Politécnica Salesiana].

<https://dspace.ups.edu.ec/bitstream/123456789/19439/1/UPS-GT003019.pdf>

Pérez, J., Urdaneta, E. y Custodio, A. (2014). Metodología para el diseño de una red de sensores inalámbrico. *Universidad, Ciencia y Tecnología*, 18(70), 12-22.

<http://ve.scielo.org/pdf/uct/v18n70/art02.pdf>

Quiñonez, O. (2019). *Internet de las Cosas (IoT)*. Ibukku

Quispe, 2019. (s. f.). Recuperado 2 de junio de 2022, de

<https://files.zotero.net/eyJleHBpcmVzljoxNjU0MjE2NDQ4LzJoYXNoIjoiYmI1YTdmNGE>

1NjIwNGI5Y2Q3M2U2OTYzNTdiZjFiYjUiLCJjb250ZW50VHlwZSI6ImFwcGxpY2F0aW9uXC9wZGYiLCJjaGFyc2V0IjoiliwiZmlsZW5hbWUiOiJXaWxsaWFtX1RyYWJham9fQmFjaGlzIjE6ImFwcGxpY2F0aW9wMTkucGRmIn0%3D/793582b0d0100571f238c4a8966bb78b79b5466a0d6efd689acb6fd6e948357e/William_Trabajo_Bachiller_2019.pdf

Reimondo, G. (15 de enero de 2019). *Seguridad en Redes LoRaWAN TM*. Tecnología Humanizada. <https://humanizationoftechnology.com/seguridad-en-redes-lorawan/revista/iot/01/2019/>

Riot-Os. (s.f.). *Heltec WiFi LoRa 32 V2*. boards https://doc.riot-os.org/group__boards__esp32__heltec-lora32-v2.html

Robu.In. (s. f.). *MG811 Module, Air Carbon Dioxide/CO2 Sensor*. Recuperado el 26 de enero de 2023 de <https://robu.in/product/mg811-module-air-carbon-dioxide-co2-sensor/>

Vidal, M. (2014). *Sistema de censado remoto para la medición de variables ambientales* [Trabajo de posgrado, Universidad Veracruzana]. <https://cdigital.uv.mx/bitstream/handle/123456789/41561/VidalCrisantoMariaConcepcion.pdf?sequence=1&isAllowed=y>

Winsen. (s.f.). *Solid Electrolyte CO2 gas sensor (Model: MG811). Manual version 1.2*. <https://datasheetspdf.com/mobile/1415579/Winsen/MG811/1>

Anexos