



Monitoreo de ambientes con Redes LPWAN LoRA

Tutor Técnico: Lic. Gonzalo Zabala

Profesora de Trabajo Final: Dra. Marcela Samela

Alumno: Gabriel Enrique Seri

Trabajo Final de Carrera presentado para obtener el título de
Lic. en Gestión de Tecnología Informática

(diciembre, 2024)

Resumen

Desde los inicios de las comunicaciones inalámbricas ha persistido la necesidad de abarcar distancias mayores con un consumo energético eficiente. Es por ello que con el transcurso del tiempo han surgido las redes LPWAN (Low Power Wide Area Network). Al abordar las mismas, se encuentran tecnologías específicas como LoRa (Long Range) y LoRaWAN (Long Range Wide Area Network).

Esta investigación se centra en analizar la viabilidad de implementar dicha tecnología en un entorno de silos ubicados en áreas rurales. Las tecnologías tradicionales como las redes de telefonía celular y la red WiFi no cumplen los requisitos necesarios para su despliegue en estas áreas, dado que para abarcar grandes distancias necesitan un gran número de antenas y una fuente de energía constante la cual no está disponible en las zonas rurales.

El presente trabajo describe paso a paso el armado de una solución de bajo costo y largo alcance. El objetivo es evaluar una red para dispositivos IoT, Internet of Things, y optimizar recursos en aquellas áreas que no cuentan con otro medio de comunicación.

Para llevar a cabo la investigación se implementó un prototipo o PoC (proof of concept) con kits de desarrollo ESP32 LoRa los cuales se desempeñaron como transmisor y receptor.

De esta manera se obtuvo como resultado un enlace punto a punto sobre el cual se realizaron ensayos para evaluar su alcance y confiabilidad, tomando muestras de temperatura con un sensor configurado para este fin.

Palabras Claves: *iot, lpwan, lora, lorawan, wifi*

Abstract

Since the inception of wireless communications, there has been a persistent need to cover greater distances with efficient energy consumption. This is why LPWAN (Low Power Wide Area Network) networks have emerged over time. When addressing these networks, specific technologies such as LoRa (Long Range) and LoRaWAN (Long Range Wide Area Network) are encountered.

This research focuses on analyzing the feasibility of implementing this technology in a silo environment located in rural areas. Traditional technologies such as cellular networks and WiFi do not meet the necessary requirements for deployment in these areas, as they need a large number of antennas and a constant power source to cover long distances, which is not available in rural zones.

This work describes step-by-step the assembly of a low-cost, long-range solution. The objective is to evaluate a network for IoT (Internet of Things) devices and optimize resources in areas that lack other means of communication.

To carry out the research, a prototype or PoC (proof of concept) was implemented with ESP32 LoRa development kits, which functioned as transmitter and receiver. As a result, a point-to-point link was obtained on which tests were conducted to evaluate its range and reliability, taking temperature samples with a sensor configured for this purpose.

Keywords: *iot, lpwan, lora, lorawan, wifi*

Dedicatoria

Dedico este trabajo final con profundo cariño a mi esposa, quien ha sido mi constante apoyo a lo largo de toda mi carrera. También va dirigido a mi hijo, quien con comprensión entendió los momentos en los que prioricé el estudio y, en consecuencia, dejé de compartir tiempo con él. Además, quiero rendir homenaje a mis padres, quienes siempre me alentaron a seguir adelante. Por último, pero no menos importante, dedico este trabajo también a mi hija Josefina, quien recientemente llegó a nuestras vidas.

Agradecimientos

Quiero expresar mi profundo agradecimiento al Licenciado Gonzalo Zabala, mi tutor, por brindarme valiosos consejos y aportes que no solo me guiaron en el desarrollo de la investigación, sino que también contribuyeron significativamente a la mejora de mi capacidad de redacción.

No puedo dejar de agradecer a mi esposa, Soledad, por su crítica constructiva durante la revisión de este trabajo.

Asimismo, agradezco enormemente al Licenciado Sebastián Galleano, mi amigo, quien me brindó consejos cruciales en diferentes momentos.

Por último, mi reconocimiento a mi compañero, Ezequiel McGovern, por su colaboración fundamental al proporcionar los componentes de hardware necesarios para la realización exitosa del prototipo.

Índice General

Capítulo 1	11
1.1 Introducción	11
1.2 Pregunta de Investigación	12
1.3 Hipótesis	12
1.4 Objetivo General	12
1.5 Objetivos Específicos	12
1.6 Metodología	13
1.7 Justificación	13
Capítulo 2	14
2.1 Introducción	14
2.2 LoRa y LoRaWAN	14
2.2.1 Definición de LoRa	14
2.2.2 Definición de LoRaWAN	15
2.2.3 Tipos de Dispositivos LoRaWAN	15
2.2.4 Métodos de Conexión	17
2.2.5 Indicadores RSSI y SNR	18
2.3 Arquitectura LoRaWAN	18
2.4 Tipos de flujos de datos utilizados	19
2.5 Diferencia con otras Tecnologías	19
2.6 Limitaciones de LoRaWAN	21
2.7 Aplicaciones Prácticas	23
2.8 Estudios Actuales	23
2.9 Características del Problema a Investigar	25
2.9.1 Tecnologías Actuales	25
2.10 Propuesta	27
2.10.1 Cronograma	27
Capítulo 3	28
3.1 Introducción	28
3.2 Metodología	28
3.3 Componentes del Prototipo	29
3.2.1 Kit de desarrollo	29
3.2.2 Sensor de Temperatura	30
3.2.3 Antena	31

3.2.4 Protoboard	32
3.2.5 Batería	32
3.2.6 Sensor de descarga de batería	32
3.4 Software	33
3.4.1 IDE Arduino	33
3.5 Desarrollo de la Solución	34
3.5.1 Configurando el entorno	35
3.5.2 Instalación de bibliotecas	36
3.5.3 Consideraciones previas a la programación	37
3.5.4 Protocolo SPI	37
3.5.5 Display Oled	38
3.5.6 Diagrama de interconexión y pines de programación	38
3.6 Primer montaje	39
3.6.1 Esquema del primer montaje	40
3.6.2 Programación de los nodos	41
3.7 Armado del Prototipo Final	42
3.7.1 Nodo emisor con sensor de temperatura	43
3.7.3 Servidor en la nube	44
3.8 Resumen del capítulo	45
Capítulo 4	46
4.1 Introducción	46
4.2 Pruebas de sobre el primer montaje	46
4.2.1 Prueba Nro. 1 Conectividad	46
4.2.2 Prueba Nro. 2 Recuperación de la red	47
4.2.3 Prueba Nro. 3 Alcance y calidad de señal	48
4.3 Pruebas sobre el montaje final	49
4.3.2 Obtención de lecturas del sensor	50
4.3.3 Prueba almacenado de datos	51
4.3.4 Prueba de duración de la batería	52
4.3.5 Monitoreo e informe de datos para usuario final	52
4.4 Síntesis	53
Capítulo 5	54
5.1 Introducción	54
5.2 Conclusiones	54

5.3 Síntesis	55
Capítulo 6	56
6.1 Introducción	56
6.2 Recomendaciones	56
6.2.1 Obtener buena señal con LoRa	56
6.3 Trabajos Futuros	56
6.3.1 Implementación de base de datos	56
6.3.3 Implementación de botón de información	57
6.3.4 Pruebas exhaustivas sobre la seguridad	57
Anexos	58
Programación del nodo emisor	58
Programación del nodo receptor	60
Nodo emisor con sensor de temperatura	62
Nodo receptor con envío de datos a internet	65
Acrónimos	71
Referencias	72

Índice de Tablas

Tabla 1	Descripción general de las tecnologías inalámbricas	21
Tabla 2	Propuestas similares	26
Tabla 3	Cronograma	27
Tabla 4	Pruebas de Calidad de Señal	49

Índice de Gráficos

Figura 1	Clase de Dispositivos	16
Figura 2	Arquitectura LoRaWAN	18
Figura 3	Comparación con Otros Tipos de Redes	20
Figura 4	Monitoreo de Ganado	24
Figura 5	Placa de desarrollo TTGO ESP32 LoRa	30
Figura 6	Módulo de Temperatura BME280	31
Figura 7	Antena LoRa Exterior	31
Figura 8	Protoboard	32
Figura 9	Módulo ina219	33
Figura 10	IDE Arduino	34
Figura 11	Configuración IDE Arduino	35
Figura 12	Configuración de Placa	36
Figura 13	Instalación de librerías adicionales	37
Figura 14	Esquema de Pines Placa de Desarrollo LoRa.	39
Figura 15	Esquema del Primer Montaje	40
Figura 16	Test Nodo Emisor	41
Figura 17	Test Nodo Receptor	42
Figura 18	Esquema Prototipo Final	42
Figura 19	Conexión de Sensor en Nodo de Envío	43
Figura 20	Pantalla Oled Nodo Emisor	46
Figura 21	Monitor Serial IDE Arduino Nodo Emisor	47
Figura 22	Pantalla Oled Nodo Receptor	47
Figura 23	Conexión LoRa y WiFi en Nodo Receptor	50
Figura 24	Lectura de Temperatura	50
Figura 25	Recepción de temperatura y RSS	50
Figura 26	Envío de Datos a Planilla	51
Figura 27	Recepción de Datos en Planilla	51
Figura 28	Medición de Consumo	52
Figura 29	Representación Gráfica de Resultados en Grafana	53

Capítulo 1

En este capítulo se abordan los conceptos generales sobre IoT, LoRa y LoRaWAN, se describen los objetivos, el contexto científico, el motivo y alcance del trabajo de investigación.

1.1 Introducción

Internet de las cosas, IoT por sus siglas en inglés, hace mención a una amplia variedad de dispositivos, como smartphones, sensores de humo o cámaras de seguridad, que pueden conectarse a internet y generar datos en tiempo real. Sin embargo, estos dispositivos tienen diferentes capacidades computacionales, lo que se traduce en diferencias en su ancho de banda, procesamiento, memoria y consumo de energía. Esto hace que las tecnologías actuales, como el Wi-Fi, tengan dificultades para satisfacer las demandas que requieren los dispositivos IoT. (Centenaro, Vangelista, 2016).

Los avances en las tecnologías de transmisión inalámbrica de datos responden a la necesidad de estándares de bajo consumo, amplia cobertura y bajo costo que satisfagan los requisitos de IoT. Como respuesta, surgió LoRa, un estándar de comunicación para redes LPWAN, que es el tema principal de este trabajo de investigación. Se espera que estas redes manejen una parte importante de las conexiones IoT en los próximos años, las mismas han sido estandarizadas bajo la especificación LPWAN LoRaWAN™ bajo una alianza que ha creado un programa de certificación para garantizar su funcionamiento. (Centenaro, Vangelista, 2016).

La motivación de este trabajo de investigación surge de la necesidad de contar con dispositivos de monitoreo y control en zonas rurales aisladas donde no llegan redes de datos inalámbricas. Por ello, se propone analizar la factibilidad del uso de la tecnología LPWAN LoRa para controlar la temperatura de los silos en dichas áreas. El objetivo es comprobar la eficacia de la tecnología tanto en entornos libres como así también con obstáculos, a fin de obtener un medio probado y listo para diferentes dispositivos IoT.

Aunque el enfoque principal de la investigación es brindar una solución para áreas sin cobertura de redes inalámbricas, también se considera la posibilidad de su uso en entornos urbanos con red celular de datos 4G y Wifi. En este caso, se espera facilitar el despliegue y optimizar la implementación, permitiendo cubrir un área mayor con una cantidad menor de antenas.

El trabajo está estructurado en introducción, descripción detallada de LoRa y su protocolo LoRaWAN, investigación de casos de uso, comparación con otras tecnologías existentes, diseño y construcción de un prototipo funcional, evaluación de resultados, conclusiones y propuestas para futuros trabajos.

1.2 Pregunta de Investigación

¿Cuál sería la forma adecuada para monitorear la temperatura de un silo de granos en una zona rural que carece de conectividad a redes de datos e instalaciones de energía eléctrica?

1.3 Hipótesis

Supervisar la temperatura de silos de granos es una tarea importante para garantizar la calidad y seguridad del producto almacenado. Para llevar a cabo este monitoreo se requiere de un sistema remoto de sensores. En áreas rurales a menudo no existe cobertura de redes de datos inalámbricas tradicionales. Para abordar esta necesidad se formula la siguiente hipótesis:

Una red LPWAN LoRa es efectiva para monitorear en forma remota la temperatura de silos de granos en zonas rurales dadas sus características de largo alcance y bajo consumo.

1.4 Objetivo General

Analizar la factibilidad del uso de tecnología LoRa para el monitoreo remoto de temperatura de silos en espacios rurales desprovistos de conectividad.

1.5 Objetivos Específicos

Para poder alcanzar el objetivo principal se redactaron los siguientes objetivos secundarios los cuales buscan:

- 1- Comparar el estándar LoRa con respecto a otras tecnologías inalámbricas para transmisión de datos con respecto al alcance y eficiencia energética.
- 2- Desarrollar un prototipo para verificar su factibilidad en la medición de temperatura de silos que envíe los valores tomados al receptor que a su vez estará conectado a internet y permitirá acceder a los mismos desde cualquier punto.
- 3- Elaborar recomendaciones de diseño basadas en los resultados obtenidos en la investigación.

La investigación permitirá ampliar el conocimiento acerca de las tecnologías de transmisión de datos inalámbricas, y además presentará alternativas innovadoras para solucionar problemas y necesidades frecuentes en el ámbito de las comunicaciones de dispositivos IoT.

Las mismas pueden ir desde el monitoreo de la calidad de agua en un estanque, hasta sistemas de seguridad en áreas remotas sin acceso a la red eléctrica.

1.6 Metodología

En este trabajo se utilizará una metodología de investigación cuantitativa del tipo experimental, donde se estudiará el resultado de la implementación de un prototipo utilizando la red inalámbrica LoRa.

1.7 Justificación

El motivo de este trabajo es investigar la tecnología de comunicación inalámbrica LoRa con el fin de monitorear y reportar las variaciones de temperatura de un silo de granos ubicado en un ambiente rural. Tanto el escenario como la variable a medir se escogieron pensando en la utilidad del dato obtenido. Los granos almacenados generan un nivel de temperatura que debe ser controlado para evitar variaciones abruptas. De lo contrario, se puede generar humedad dentro del silo debido a la condensación, lo que puede provocar la proliferación de hongos y moho, resultando en pérdidas parciales o totales de la cosecha. Además, las variaciones abruptas de temperatura, junto con otros factores, pueden desencadenar una combustión instantánea en los granos, lo que podría resultar en la pérdida total del silo. (Appert, 1993).

La lectura en tiempo real de la temperatura del silo es crucial para los productores rurales, ya que permitirá detectar cualquier variación en el nivel de temperatura y tomar medidas oportunas para mantener la cosecha almacenada correctamente y evitar pérdidas económicas. Por lo tanto, es fundamental contar con alertas en caso de cualquier cambio en la temperatura del silo, de esta manera el productor podrá actuar rápidamente.

Capítulo 2

2.1 Introducción

La tecnología de red inalámbrica LPWAN fue desarrollada para abarcar áreas extensas, permitiendo la conexión de dispositivos de bajo ancho de banda alimentados por baterías, específicamente diseñada para redes de máquina a máquina (M2M) e Internet de las cosas (IoT). (Centenaro, Vangelista, 2016).

A diferencia de las redes móviles tradicionales, las redes LPWAN operan a un costo significativamente menor y con mayor eficiencia energética. Además, son capaces de admitir una cantidad mayor de dispositivos conectados en un área más amplia, lo que las convierte en una alternativa eficaz para aplicaciones IoT que requieren monitoreo remoto de datos en tiempo real y que se encuentran en áreas rurales con limitada infraestructura de conectividad. (Centenaro, Vangelista, 2016).

2.2 LoRa y LoRaWAN

Aunque su nombre pueda parecer similar, es importante mencionar que LoRa y LoRaWAN son tecnologías distintas. LoRa es una técnica de modulación desarrollada por Semtech, mientras que LoRaWAN es el protocolo utilizado para conformar la red de gateway y sensores. A pesar de que ambas tecnologías están diseñadas para trabajar juntas, es esencial distinguir entre ellas para comprender correctamente su funcionamiento en un sistema IoT (Hernández, R., 2021).

2.2.1 Definición de LoRa

LoRa es un tipo de red inalámbrica que se diferencia de otras redes por su largo alcance y bajo consumo energético. Estas características la hacen ideal para proyectos de IoT que operan con una fuente externa de energía, como por ejemplo una batería. LoRa utiliza una modulación en radiofrecuencia patentada por la empresa Semtech, llamada Chirp Spread Spectrum (CSS). (Centenaro, Vangelista, 2016).

Dicha modulación se usa en comunicaciones militares y espaciales, y sus principales ventajas son la capacidad para lograr grandes distancias y mantener la estabilidad frente a diferentes tipos de interferencias. LoRa opera en el espectro de radiofrecuencia sin licencia conocido como banda de radio ICM (Industrial, Científica y Médica). Dependiendo de la zona

geográfica donde se utilice, las frecuencias para operar serán: 915 MHz para América, 868 MHz para Europa y 430 - 433 MHz para Asia. (Centenaro, Vangelista, 2016).

2.2.2 Definición de LoRaWAN

LoRaWAN es un protocolo de red que utiliza la tecnología LoRa para conectar diferentes dispositivos LoRa, y manejar sus especificaciones de conexión (canal, ancho de banda, cifrado de datos, etc.). Está compuesto por gateways y nodos, siendo la principal tarea de los gateways recibir y enviar información a los nodos. Los nodos son los dispositivos finales que cumplen una función específica según su aplicación, y envían y reciben información hacia el gateway. (Centenaro, Vangelista, 2016).

Entre sus características principales se destacan:

- Seguridad con encriptación AES 128 bits.
- Redes públicas y privadas.
- Bajo consumo y largo alcance.
- Baja transferencia de datos (hasta 242 bytes).

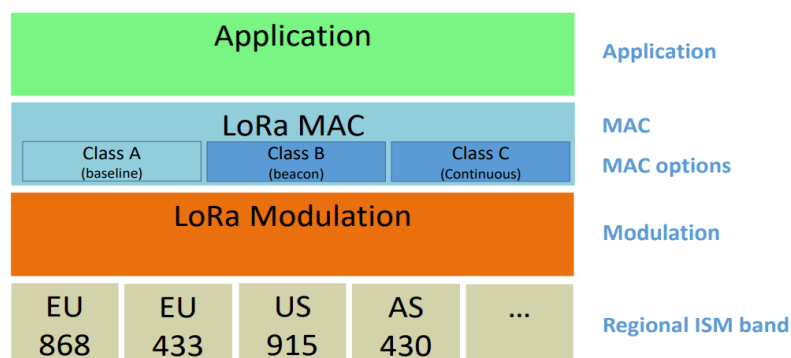
LoRaWAN es una solución diseñada para transmitir datos a baja tasa de transferencia y con exigencias limitadas en cuanto a la latencia, es decir, el tiempo que tarda un paquete en ser transmitido dentro de la red. (Centenaro, Vangelista, 2016).

Algunos de los escenarios donde se utiliza son:

- Redes de sensores en ciudades o Smart cities, industrias o zonas rurales.
- Redes IoT que no requieran streaming audio y/o video.
- Redes privadas que operen con un ancho de banda bajo.

2.2.3 Tipos de Dispositivos LoRaWAN

En la actualidad existen tres tipos de dispositivos LoRaWAN, clasificados como Clase A, Clase B y Clase C. (2) Todos los dispositivos LoRaWAN deben implementar la Clase A, mientras que la Clase B y la Clase C son extensiones de los dispositivos de la Clase A, como se muestra en la figura 1. (LoRaWAN® specification v1.0.3. 2020).

Figura 1*Clase de Dispositivos*

Nota. Imagen tomada de (LoRaWAN® specification v1.0.4. 2020).

Clase A: La comunicación de LoRaWAN es bidireccional: cuando un dispositivo transmite un paquete a través de la red, entra en modo de escucha y luego abre dos ventanas de recepción en momentos específicos. Si no se recibe ninguna respuesta en estas ventanas, la siguiente ventana será después de la siguiente transmisión del dispositivo. La Clase A es la de menor consumo energético, por lo que es ideal para dispositivos alimentados por batería. (LoRaWAN® specification v1.0.3. 2020).

Clase B: Los dispositivos de Clase B abren ventanas de recepción en momentos programados que están sincronizados con el Gateway. La conveniencia de usar una batería dependerá de la duración de estas ventanas de recepción programadas. (LoRaWAN® specification v1.0.3. 2020).

Clase C: En este tipo de clase, los dispositivos están en modo de escucha la mayor parte del tiempo y solo se activan en modo transmisor cuando es necesario, lo que permite una comunicación de baja latencia. Sin embargo, este enfoque también conlleva un mayor consumo de energía en comparación con los dispositivos de Clase A. (LoRaWAN® specification v1.0.3. 2020).

2.2.4 Métodos de Conexión

La conexión hacia un dispositivo final se puede realizar de dos maneras, estas son OTAA y ABP. (LoRaWAN® specification v1.0.3, 2020).

OTAA: El modo de activación OTAA (Over The Air Activation) es considerado el más seguro para conectar un dispositivo a la red, ya que se establecen tres parámetros esenciales para la seguridad de la comunicación:

- DevEUI: El DevEUI es un identificador único que se configura en el dispositivo durante su fabricación y que no puede ser modificado posteriormente. (LoRaWAN® specification v1.0.3, 2020).
- AppEUI: El AppEUI es un identificador único de 64 bits que se utiliza para agrupar los dispositivos según su aplicación. A diferencia del DevEUI, este parámetro puede ser configurado por el usuario final. (LoRaWAN® specification v1.0.3, 2020).
- AppKey: Es una clave configurable por el usuario del tipo AES de 128 bits. (LoRaWAN® specification v1.0.3, 2020).

ABP: La activación por personalización es el método más sencillo de conexión, ya que combina los pasos de personalización y activación en un solo proceso sin requerir un procedimiento de aceptación adicional. Los parámetros asociados a este método son:

- DevAddr: El DevAddr es una dirección única de 32 bits que se utiliza para identificar un dispositivo en la red LoRaWAN. Esta dirección es equivalente a una dirección IP en otras redes de comunicación. (LoRaWAN® specification v1.0.3, 2020).
- NwkSKey: s una clave de sesión de red específica para el dispositivo final, utilizada para cifrar y autenticar los mensajes transmitidos en la red LoRaWAN, garantizando así la integridad y la confidencialidad de los mismos. (LoRaWAN® specification v1.0.3, 2020).
- AppsKey: Es una clave de sesión de aplicación, utilizada para cifrar de extremo a extremo la comunicación entre el dispositivo final y el servidor de aplicaciones. (LoRaWAN® specification v1.0.3, 2020).

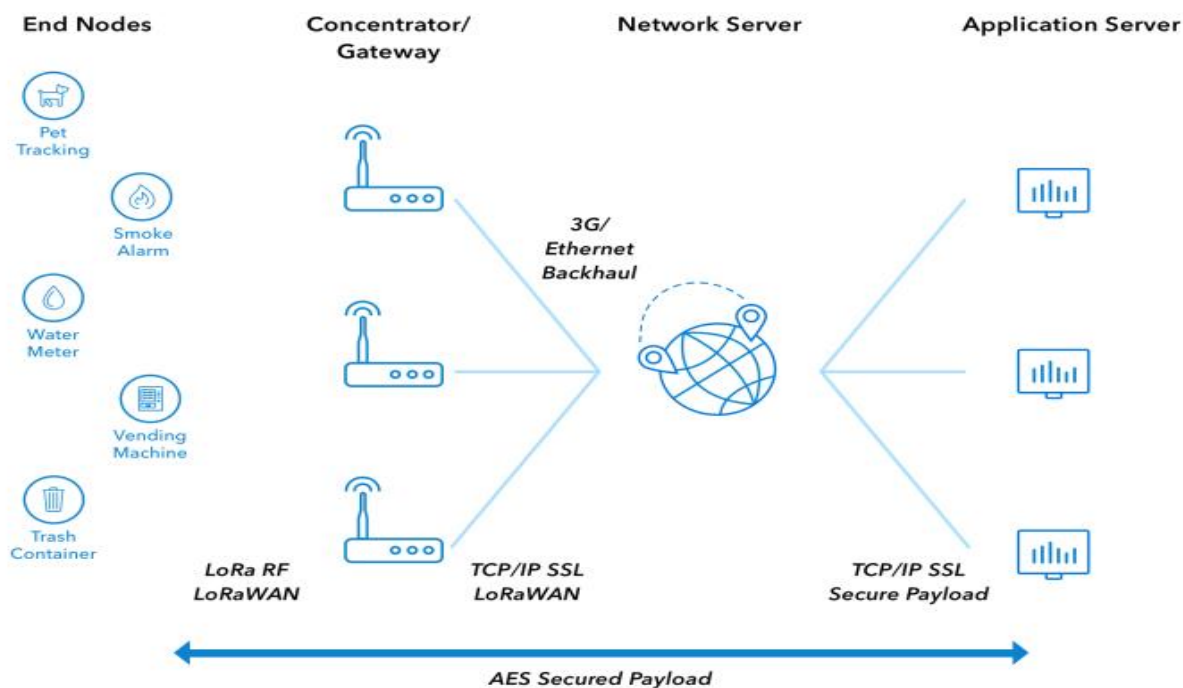
2.2.5 Indicadores RSSI y SNR

La calidad y el alcance de la señal LoRa dependen de dos valores: RSSI y SNR. RSSI mide la potencia de la señal recibida a una distancia determinada, mientras que SNR mide la relación señal-ruido en el ambiente donde se despliega la red. Ambos valores se miden en decibelios (dB). En una red LoRa, el rango de valores admitidos para RSSI va desde 0 dB hasta -125 dB. Si el valor de RSSI es mayor a -125 dB, se perderán paquetes y la comunicación no será confiable. En cuanto a SNR, el valor óptimo es 9 dB, lo que indica que no se registra ningún tipo de ruido. El rango de valores de SNR admitidos en una red LoRa puede comprender entre 9 dB y -20 dB. Un valor de SNR de -20 dB es el máximo tolerado por la red. (*LoRaWAN® specification v1.0.4*, 2020).

2.3 Arquitectura LoRaWAN

Figura 2

Arquitectura LoRaWAN



Nota. Imagen tomada de (The Things Network, 2022).

La figura 2 ilustra la composición típica de una red LoRaWAN, cuya topología se conoce como estrella. En esta topología, los dispositivos finales, llamados "nodos", se conectan directamente a los gateways, los cuales a su vez transmiten los datos a los servidores finales a través de internet. (The Things Network, 2022).

2.4 Tipos de flujos de datos utilizados

Respecto a los tipos de flujo de datos compatibles, LoRa es muy versátil y acepta una amplia variedad que incluye:

- Simplex: Es una comunicación unidireccional entre un transmisor y un receptor. (Bertoletti, P. 2019).
- Half-dúplex: En este tipo de comunicación dos o más dispositivos pueden enviar o recibir datos, pero no pueden hacerlo en forma simultánea. (Bertoletti, P. 2019).
- Full-dúplex: Todos los dispositivos de la red pueden transmitir y recibir datos en forma simultánea. (Bertoletti, P. 2019).

2.5 Diferencia con otras Tecnologías

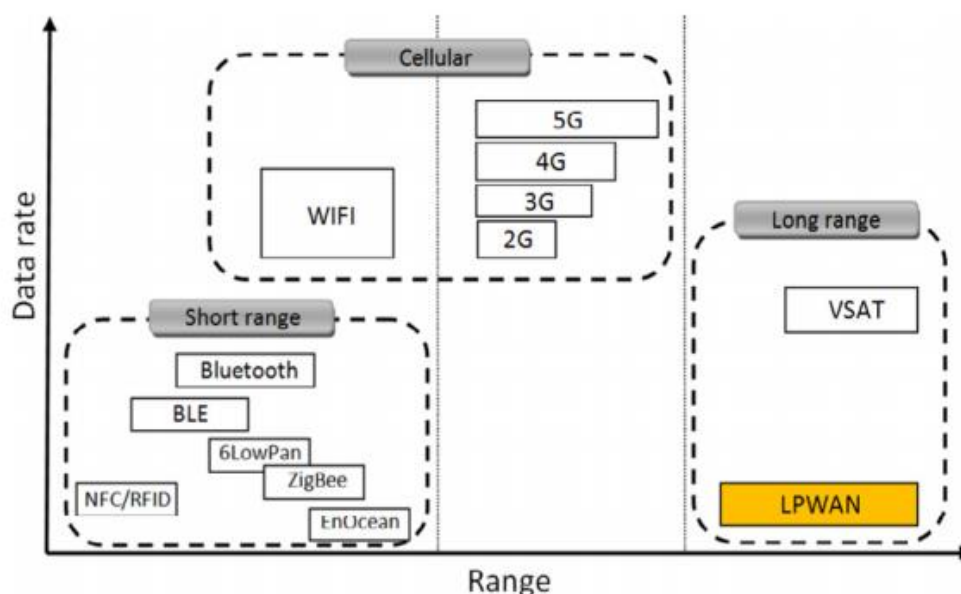
Tal y como se ha mencionado con anterioridad, resulta fundamental destacar que las redes de baja potencia y largo alcance (LPWAN) encuentran su óptimo desempeño en las aplicaciones del Internet de las Cosas (IoT) que demandan la transmisión de volúmenes reducidos de información a distancias notables. En la figura 3, se ilustra una comparativa entre la tecnología LPWAN y otras alternativas de transmisión inalámbrica, tales como las redes de alcance medio y corto, como es el caso de Wifi o 4G. (Mekki, Bajic, Chaxel, Meyer., 2018).

Este análisis comparativo presenta una importante disyuntiva: la velocidad de transmisión de datos en contraposición con la capacidad de cobertura máxima. Mediante esta representación gráfica, se delinean las propiedades fundamentales de cada tecnología, tomando como base estos dos elementos esenciales para su desempeño. (Mekki, Bajic, Chaxel, Meyer., 2018).

En esta perspectiva, se evidencia claramente cómo las redes LPWAN emergen como la opción idónea en contextos donde la cantidad de datos a transmitir es limitada y la cobertura geográfica requerida es significativa. (Mekki, Bajic, Chaxel, Meyer., 2018).

Figura 3

Comparación con Otros Tipos de Redes



Nota. Imagen tomada de (Mekki, Bajic, Chaxel, Meyer., 2018).

Como se puede apreciar en la figura 1.3, las redes LPWAN exhiben una capacidad de transmisión de datos inferior en contraste con las redes WiFi o 4G. No obstante, un aspecto llamativo es su capacidad de alcanzar distancias considerables con la utilización de un número reducido de antenas, lo cual supone una característica distintiva de estas redes de baja potencia y largo alcance. Esta particularidad no solo se traduce en eficiencia espacial, sino también en la posibilidad de abarcar áreas extensas con una infraestructura menos densa. (Mekki, Bajic, Chaxel, Meyer., 2018).

Durante el transcurso de los últimos años el panorama tecnológico ha sido testigo del florecimiento de nuevas propuestas en el ámbito de las tecnologías de redes de baja potencia y largo alcance entre las que figuran destacadas opciones como Sigfox y NB-IoT en conjunto con LoRa. Este último grupo ha demostrado una rápida consolidación en el liderazgo dentro de este ámbito. Cabe destacar que entre estas tecnologías existen diferencias técnicas de suma relevancia. (Mekki, Bajic, Chaxel, Meyer., 2018).

Es importante destacar que existen notables diferencias técnicas entre ellas, como se presenta en la siguiente tabla:

Tabla 1

Descripción general de las tecnologías inalámbricas

	Sigfox	LoRaWAN	NB-IoT
Modulación	BPSK	CSS	QPSK
Frecuencia	Banda ISM sin licencia (868 MHz en Europa, 915 MHz en Norteamérica y 433 MHz en Asia)	Banda ISM sin licencia (868 MHz en Europa, 915 MHz en Norteamérica y 433 MHz en Asia)	Bandas de frecuencia LTE con licencia
Banda Ancha	100 Hz	250 kHz y 125 kHz	200 kHz
Velocidad Máxima de datos	100 bps	50 kbps	200 kbps
Bidireccional	Limitado / Half-duplex	Sí / semidúplex limitado	Sí / semidúplex ilimitado
Mensajes por día	140	Ilimitado	Ilimitado
Longitud máxima de mensaje	12 bytes	243 bytes	1600 bytes
Rango	10 km (urbano), 40 km (rural)	2 km (urbano), 10 km (rural)	1 km (urbano), 10 km (rural)
Tolerancia a la Interferencia	Muy alto	Muy Alto	Bajo
Encriptación	No Compatible	AES 128b	LTE
Velocidad Adaptativa	No	Sí	No
Permite red privada	No	Sí	No

Nota. Tabla tomada de (Mekki, Bajic, Chaxel, Meyer., 2018).

2.6 Limitaciones de LoRaWAN

La tecnología LoRa (Long Range) ofrece ventajas significativas para la implementación en zonas rurales debido a su capacidad para transmitir datos a largas distancias con bajo consumo de energía. Sin embargo, también presenta algunas limitaciones y desafíos que deben considerarse al implementarla en entornos rurales:

Cobertura limitada: A pesar de su largo alcance, la cobertura de una sola estación base LoRa puede ser limitada en áreas rurales con topografía accidentada o vegetación densa. Se

necesitarían más estaciones base para garantizar una cobertura confiable en toda el área. (The Things Network, 2022).

Ancho de banda limitado: LoRa utiliza un ancho de banda estrecho para maximizar la eficiencia energética y la cobertura. Sin embargo, esto también limita la cantidad de datos que se pueden transmitir en un solo intervalo de tiempo, lo que puede ser una restricción para aplicaciones que requieren transferencias de datos rápidas o grandes volúmenes de información. (The Things Network, 2022).

Latencia: Debido a su diseño de bajo consumo de energía, las comunicaciones LoRa pueden tener una latencia relativamente alta en comparación con tecnologías de alta velocidad como el 4G o el 5G. Esto puede ser problemático para aplicaciones que necesitan respuestas en tiempo real. (The Things Network, 2022).

Consumo de energía en dispositivos finales: Aunque LoRa es eficiente en términos de consumo de energía, los dispositivos finales aún requieren energía para transmitir y recibir datos. La necesidad de conservar la vida útil de la batería puede limitar la frecuencia y la cantidad de transmisiones de datos. (The Things Network, 2022).

Capacidad de carga de red: A medida que el número de dispositivos conectados a la red LoRa aumenta, la capacidad de la red para manejar simultáneamente múltiples conexiones puede verse afectada, lo que podría resultar en retrasos y pérdida de datos. (The Things Network, 2022).

Seguridad: Al igual que con cualquier tecnología de comunicación inalámbrica, la seguridad de las comunicaciones LoRa es crucial. Los datos transmitidos pueden estar sujetos a riesgos de interceptación o manipulación si no se implementan medidas de seguridad adecuadas. (The Things Network, 2022).

En resumen, aunque la tecnología LoRa tiene muchas ventajas para su implementación en zonas rurales, es importante considerar y abordar estas limitaciones y desafíos para garantizar una implementación exitosa y efectiva. Cada aplicación y entorno específico requerirá un análisis cuidadoso de estas consideraciones antes de decidir si LoRa es la mejor solución.

LoRaWAN no puede ser utilizada para comunicaciones de voz, video o streaming, ya que no fue desarrollado para tal fin. (The Things Network, 2022).

2.7 Aplicaciones Prácticas

LoRa y LoRaWAN pueden ser instalados en escenarios muy variados, en la actualidad existen una gran familia de sensores IoT que son utilizados para una gran variedad de tareas entre los cuales se destacan:

- Monitoreo del medio ambiente (calidad del aire, gestión y protección del agua, prevención de incendios en bosques, estudio de la tierra).
- Edificio y/o casa inteligente.
- Cadena de suministro y logística.
- Seguimiento de vehículos.
- Aplicaciones para el agro (estudio del suelo, riego)
- Cuidado de la salud.

A finales de 2019 los nodos basados en LoRa desplegados por el mundo alcanzan la cifra de 80 millones. (Mekki, Bajic, Chaxel, Meyer., 2018).

2.8 Estudios Actuales

Despliegues en Áreas Urbanas

En diciembre de 2017, las empresas Semtech y MachineQ llevaron a cabo un estudio conjunto diseñado para simular una red densa a gran escala. Para ello, instalaron múltiples sensores que se conectaron a varias puertas de enlace para simular un entorno de 10.000 sensores. De esta manera, pudieron medir y evaluar el valor de una red densa, asegurando la tasa de éxito de paquetes enviados y recibidos, así como su resistencia a la interferencia, manteniendo un buen rendimiento y calidad de servicio. Se obtuvieron resultados sobre la capacidad de LoRaWAN para manejar grandes cantidades de datos (250.000 paquetes por día) generados por sensores ubicados estratégicamente en un entorno urbano. Finalmente, se comprobó la confiabilidad de esta tecnología en la transmisión y recepción de datos. (Gilson, R., Bajic, K., Chaxel, E., & Meyer, F., 2018).

Despliegues en Entornos Rurales

Los dispositivos LoRa son ideales para su uso en entornos rurales, ya que logran una gran cobertura en áreas con buena visibilidad y baja interferencia. Sus aplicaciones son muy diversas, como soluciones para la agricultura, como la automatización del riego, estudios del suelo, monitoreo climático, optimización del uso del agua, control de plagas, entre otros. También son muy útiles en la ganadería, como se puede ver en la figura 4, donde se utiliza para monitorear la salud del ganado y localizarlo en grandes terrenos. Al aprovechar la tecnología LoRa y el protocolo abierto LoRaWAN, los productores pueden digitalizar cada aspecto de su negocio, mejorando sus operaciones en general y su retorno de inversión (ROI) (Fulton, A., 2019).

Figura 4

Monitoreo de Ganado



Nota. Imagen tomada de (Fulton, A., 2019).

Estudios en nuestro país

En Argentina, durante el año 2017, se realizó una prueba de concepto para evaluar la cobertura de la red LoRaWAN en ambientes urbanos con obstáculos. Se analizó la fiabilidad de los datos simulados especialmente para la prueba. Los ensayos se llevaron a cabo en la ciudad de Córdoba, donde se midió la cobertura en edificios de hasta 50 metros de altura, y en la ciudad de Buenos Aires, en un ambiente densamente urbano con edificios de hasta 175 metros de altura. En ambos escenarios, se utilizó una sola antena para determinar su alcance máximo, obteniéndose un resultado de 1,6 km en ambiente urbano y 1,1 km en ambiente densamente urbano. (Grion, F., Petracca, G., & Lipuma, O., 2017).

Aplicaciones en el sector pesquero

La empresa ICT International, dedicada a proveer soluciones IoT para aplicaciones ambientales, ha implementado sensores LoRaWAN en boyas marinas para medir la salinidad y temperatura del agua. Estas boyas envían los datos recolectados a una aplicación de monitoreo en tiempo real, la cual notifica directamente a los pescadores y cultivadores de ostras marinas. Esta solución permite predecir y alertar sobre cambios bruscos en el clima que puedan afectar negativamente la pesca y el cultivo, brindando así una herramienta valiosa para la toma de decisiones. (Pégulu, M., 2022).

2.9 Características del Problema a Investigar

Actualmente, la mayoría de las zonas rurales carecen de cobertura de red celular de datos extendida, lo que impide la implementación de soluciones de monitoreo remoto. Debido a esta limitación, los productores no tienen información en tiempo real sobre el estado de sus activos y no pueden tomar decisiones oportunas en caso de inconvenientes.

Además de la conectividad, la disponibilidad de energía eléctrica es otro factor a considerar al momento de implementar una solución de monitoreo remoto en zonas rurales. Aunque la red eléctrica ha sido extendida en los últimos años, aún no llega a todas las áreas rurales, lo que limita la posibilidad de monitorear activos en lugares alejados.

2.9.1 Tecnologías Actuales

Conectividad Celular

Ante la necesidad de monitorear zonas rurales aisladas que carecen de cobertura de red y suministro eléctrico, se están evaluando las tecnologías disponibles que podrían ofrecer una solución viable. A pesar de que se ha considerado la posibilidad de extender la red celular existente, esta opción conllevaría la instalación de nuevas antenas de gran tamaño y la necesidad de una infraestructura de red eléctrica más robusta.

Conectividad WiFi

Una alternativa adicional podría ser la implementación de una red WiFi, la cual ofrecería independencia de una compañía de telefonía celular, sin embargo, esta opción implicaría la necesidad de instalar antenas cada 1 km para cubrir la distancia requerida, lo que podría resultar costoso y complejo de implementar en zonas rurales.

Conectividad Satelital

La tercera opción sería una conexión satelital, que podría solucionar el problema de cobertura, pero no resolvería la necesidad de conexión a la red eléctrica de los sensores de temperatura. Además, los costos de implementar una conexión satelital podrían ser significativos, lo que podría ser un obstáculo importante para los productores con recursos limitados. Por último, la falta de proveedores de servicios satelitales en todas las zonas también podría limitar la viabilidad de esta opción.

En Argentina, actualmente existen empresas que ofrecen servicios y equipamientos similares a los propuestos en este trabajo, como se detalla en la tabla 2.

Tabla 2

Propuestas similares

Empresa	Descripción	Orientación	Red	IoT	Link
Duolink	Tecnologías para el monitoreo del medioambiente	Grandes productores y PyMES	LoRA	Si	www.duolink.com.ar
Agrobit	Agricultura sustentable e inteligente	Grandes productores	SigFOX	Si	www.agrobit.com.ar
FiberCorp	Soluciones IoT para el campo	Grandes productores	4G y Satelital	Si	www.telecomfibercorp.com.ar

La solución propuesta en esta investigación considera un prototipo de bajo costo al alcance de pequeños, medianos y grandes productores, a pesar de que existen productos y servicios similares disponibles en la actualidad. La principal diferencia radica en la posibilidad de implementar la solución sin la necesidad de desplegar grandes infraestructuras de energía y/o comunicaciones.

2.10 Propuesta

Motivado por la necesidad previamente mencionada de monitorear la temperatura en un silo de granos ubicado en una zona rural, se propone implementar una red de datos LoRA que cumpla con los requisitos necesarios. Se ha elegido probar esta tecnología debido a su amplia cobertura con una sola antena, así como a la optimización de sus sensores para funcionar con baterías que, según el fabricante, pueden durar hasta 5 años, ya que solo consumen energía al momento de enviar datos.

La investigación incluye el desarrollo de un emisor con un sensor de temperatura LoRa y un receptor que puede estar conectado directamente a un servidor en la nube para almacenar los datos recibidos. También se podrá complementar con tecnología Wifi para enviar los datos a Internet.

2.10.1 Cronograma

Tabla 3

Cronograma

Tarea	Descripción	Tiempo Previsto
1	Revisión del Marco Teórico / Estado del Arte.	6 semanas
2	Determinar el ámbito físico y los actores involucrados.	4 semanas
3	Recolección de información y datos técnicos.	4 semanas
4	Diseño de la propuesta.	4 semanas
5	Construcción del prototipo.	8 semanas
6	Pruebas en entorno de laboratorio.	4 semanas
7	Implementación del prototipo en la zona rural.	1 semana
8	Pruebas en entorno definitivo.	4 semanas
9	Informe de Resultados.	3 semanas

Capítulo 3

3.1 Introducción

En este capítulo se describirá el proceso de desarrollo del prototipo que servirá como prueba de concepto y permitirá evaluar la viabilidad de la tecnología LoRa en entornos rurales. Se detallará la metodología utilizada, así como los diferentes componentes de hardware y software empleados, justificando su elección y exponiendo sus ventajas y desventajas.

Para la construcción del prototipo, se investigaron los diferentes sensores de temperatura LoRa disponibles en el mercado. Después de comparar las distintas opciones de hardware, se optó por el sensor BME280, el cual es compatible con el kit de desarrollo ESP32 LoRa.

3.2 Metodología

Este trabajo de investigación es una prueba de concepto que tiene como objetivo probar la tecnología de transmisión inalámbrica LoRa. Para lograrlo se llevó a cabo el desarrollo de la PoC siguiendo la metodología que se describe a continuación:

A) Definición del problema.

Se explica el problema a investigar, se plantean hipótesis y objetivos a lograr.

B) Estudio de Casos.

Se lleva a cabo una investigación en donde se verifican estudios actuales de la tecnología con el fin de conocer de forma temprana los alcances y limitaciones de LoRa.

C) Diseño de PoC.

Se construye el prototipo que se utilizará para realizar las pruebas de laboratorio.

D) Implementación.

Se construye el prototipo que se utilizará para realizar las pruebas de laboratorio.

E) Ejecución de pruebas.

Se llevan a cabo diferentes tipos de pruebas sobre el prototipo, verificando su funcionamiento y tomando notas de los resultados.

F) Informe de resultados.

Se elaboran informes finales en los que se detallan los resultados obtenidos, informando si se cumple con la hipótesis formulada y los objetivos propuestos.

3.3 Componentes del Prototipo

3.2.1 Kit de desarrollo

El ESP32 es un microcontrolador fabricado por Espressif Systems, que se destaca por su bajo consumo de energía y bajo costo, lo que lo hace ideal para la construcción de soluciones IoT. Otra característica importante es su amplia variedad de entornos de programación de código abierto, lo que significa que su código es gratuito y está disponible para su reutilización o modificación por parte de cualquier persona.

El microcontrolador ESP32 es un tipo de SoC (System on Chip) que incluye un procesador dual core Xtensa de 32 bits que alcanza una velocidad máxima de procesamiento de 2,4 GHz, una memoria RAM de 520 kB y soporte para conectividad inalámbrica como WiFi, Bluetooth y LoRa. Además, puede funcionar en ambientes con temperaturas extremas que van desde los -40 °C hasta los 125 °C (Bertoletti, P. 2019). (Bertoletti, P. 2019).

Por todas estas razones, se decidió utilizar el microcontrolador ESP32 para construir el prototipo, ya que será responsable de manejar la comunicación entre el sensor de temperatura, la antena LoRa y el servidor de datos en la nube. Una de las dificultades encontradas durante la investigación del kit de desarrollo fue encontrar el que mejor se ajuste a los requisitos, puesto que en el mercado existen muchos kits con diferentes características y componentes orientados a diferentes necesidades.

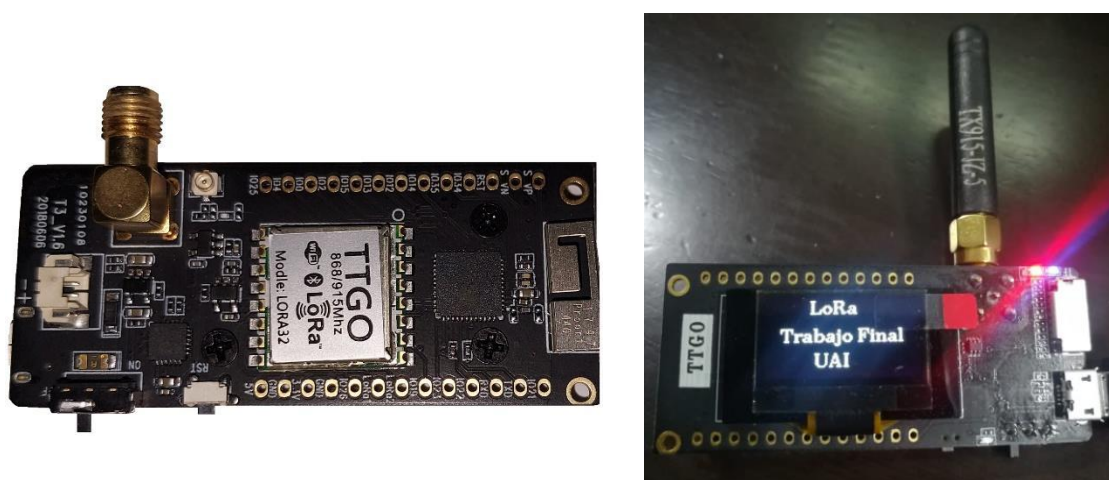
El kit seleccionado es el TTGO ESP32 LoRa, fabricado por la empresa Lily Go, que se muestra en la figura 5 y cuenta con las siguientes características:

Microcontrolador ESP32. Chip LoRa compatible con las frecuencias 868 MHz y 915 MHz. Pantalla OLED de 0.96 pulgadas. Antena SMA LoRa de 915 MHz. (Shenzhen Xin Yuan electronic technology Co., Ltd., 2016).

Este módulo en particular tiene un rango de temperatura de trabajo que va desde los -40 °C hasta los 85 °C. Su tamaño, de 64,47 mm de largo por 27 mm de ancho, lo hace ideal para armar soluciones pequeñas que no requieren grandes piezas o cajas para su ensamblado. Opera con un rango de energía entre 1,8 y 3,7 voltios y un rango de corriente entre 10 y 14 mA, por lo que se convierte en una excelente opción para funcionar con una batería recargable de litio. (Shenzhen Xin Yuan electronic technology Co., Ltd., 2016).

Figura 5

Placa de desarrollo TTGO ESP32 LoRa



3.2.2 Sensor de Temperatura

Una vez elegido el kit de desarrollo LoRa, solo queda seleccionar el sensor de temperatura que se utilizará en la construcción del prototipo. Después de buscar sensores de temperatura compatibles con el kit de desarrollo LoRa, se encontró el sensor BME280 (Figura 6). Este sensor destaca por su diminuto tamaño y su bajo consumo energético, ya que opera en un rango de 1.8 a 3.3 voltios y sensa un rango de temperaturas que van desde los -40 °C hasta 85 °C. Además de sensar temperatura, también mide con precisión la presión atmosférica y la humedad relativa. (Alldatasheet.com. 2018).

Figura 6

Módulo de Temperatura BME280



Nota. Imagen tomada de (Alldatasheet.com. 2018).

3.2.3 Antena

Dado que la antena incluida en el kit no proporciona la suficiente calidad de señal y alcance necesario para el prototipo, se decidió incorporar una antena multidireccional exterior para el nodo emisor, tal como se muestra en la figura 7. La antena exterior seleccionada tiene un largo de 600 mm, una ganancia de 8.5 dBi y trabaja dentro del rango de frecuencias de 860 y 930 MHz. Este tipo de antena es adecuada tanto para entornos urbanos como para áreas rurales. Con la incorporación de esta antena se espera mejorar significativamente la calidad de la señal y aumentar el alcance del prototipo, lo que resultará en una mejor transmisión de datos en diferentes entornos. (Seeedstudio.com. 2022).

Figura 7

Antena LoRa Exterior



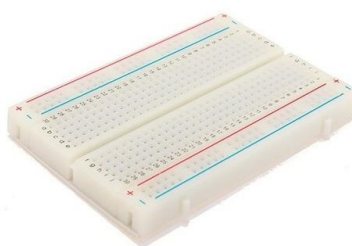
Nota. Imagen tomada de (Seeedstudio.com. 2022).

3.2.4 Protoboard

Como se puede apreciar en las figuras 5 y 6 las placas vienen preparadas para que se puedan soldar pines los cuales pueden utilizarse en otras placas o bien en protoboards. El protoboard o placa de prueba es un elemento clave en el montaje de prototipos, ya que permiten interconectar diferentes módulos y circuitos en forma rápida a través de sus orificios interconectados por pequeñas placas metálicas. Una de sus principales ventajas es que no es necesario realizar soldaduras para interconectar componentes, lo que permite montar todo tipo de circuitos en pocos minutos.

Figura 8

Protoboard



3.2.5 Batería

El nodo emisor se alimentará de una batería recargable de litio de las siguientes características:

- 5 volts de salida.
- Corriente de salida de 1000 ma.
- Dimensiones de 95 mm de largo x 25 mm de ancho y 20 mm de alto.

3.2.6 Sensor de descarga de batería

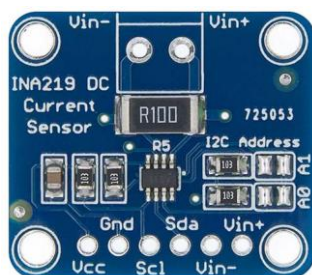
Como el nodo emisor será alimentado por una batería externa surge la necesidad de monitorear la descarga de la misma. Con el fin de conocer el consumo e informarlo en todo momento se investigó la mejor opción compatible con el kit de desarrollo. Luego de analizar varias opciones existentes en el mercado se eligió el módulo denominado “ina219” el cual se puede apreciar en la figura 9. El módulo ina219 es un sensor desarrollado por Texas Instrument encargado de monitorear el consumo de energía de cualquier circuito electrónico. Esto es posible al medir la caída de tensión a través de su resistencia “shunt”. El módulo calcula la corriente que consume el circuito midiendo la caída de tensión que se produce en la resistencia “shunt”.

A partir de ese dato es posible calcular la duración de la batería de litio conectada al nodo de origen. (INA219 Zero-Drift, 2015).

Este módulo es ideal para trabajar con el kit de desarrollo con ESP32 ya que posee una interfaz I2C la cual permite guardar fácilmente los valores censados en diferentes variables de software.

Figura 9

Módulo ina219



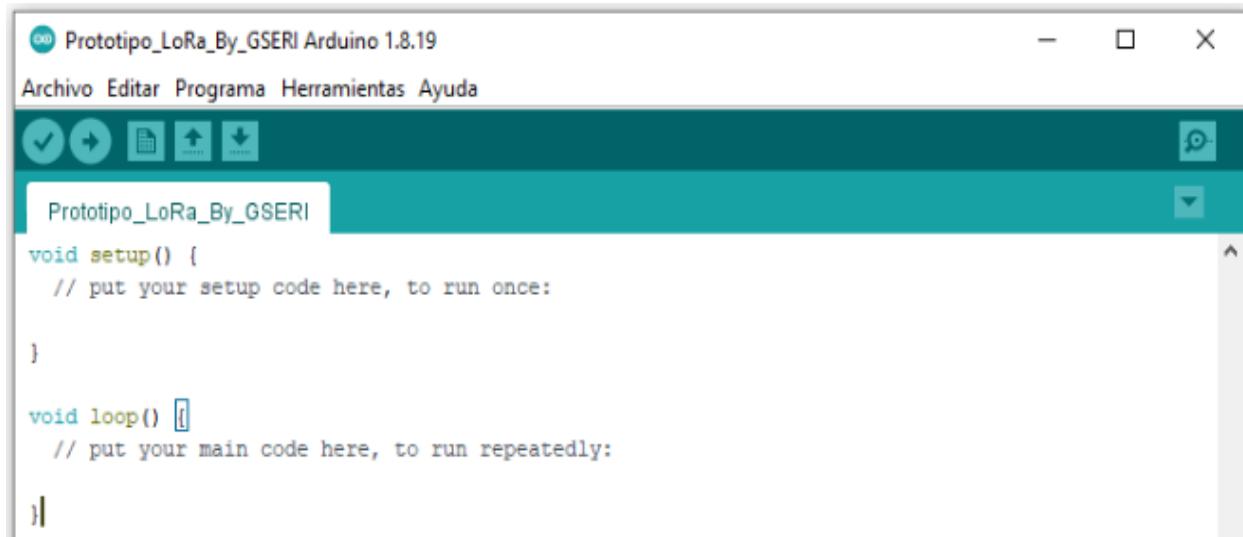
Nota. Imagen tomada de (Sensor de Corriente ina219. 2022).

3.4 Software

Al contar con el hardware para la construcción del prototipo solo resta comenzar con el análisis del software para programar e integrar la placa de desarrollo LoRa con el sensor de temperatura. Para esto se investigó que IDE (integrated development environment) es compatible con el hardware a utilizar. ESP32 es compatible con el lenguaje de programación C/C++ y Python. (Bertoletti, P. 2019).

3.4.1 IDE Arduino

Investigando los diferentes tipos de IDE compatibles con el hardware seleccionado, se encontraron dos opciones. La primera opción fue el IDE proporcionado por el fabricante de la placa de desarrollo, Espressif ESP-IDF. Como alternativa, se encontró también el IDE de Arduino, que fue diseñado para ser utilizado en establecimientos educativos y para programar placas de desarrollo de bajo costo. Además, este IDE es compatible con una gran variedad de lenguajes de programación. Por estas razones, se eligió este último para la construcción del prototipo. El mismo se puede apreciar en la figura 10.

Figura 10*IDE Arduino*

3.5 Desarrollo de la Solución

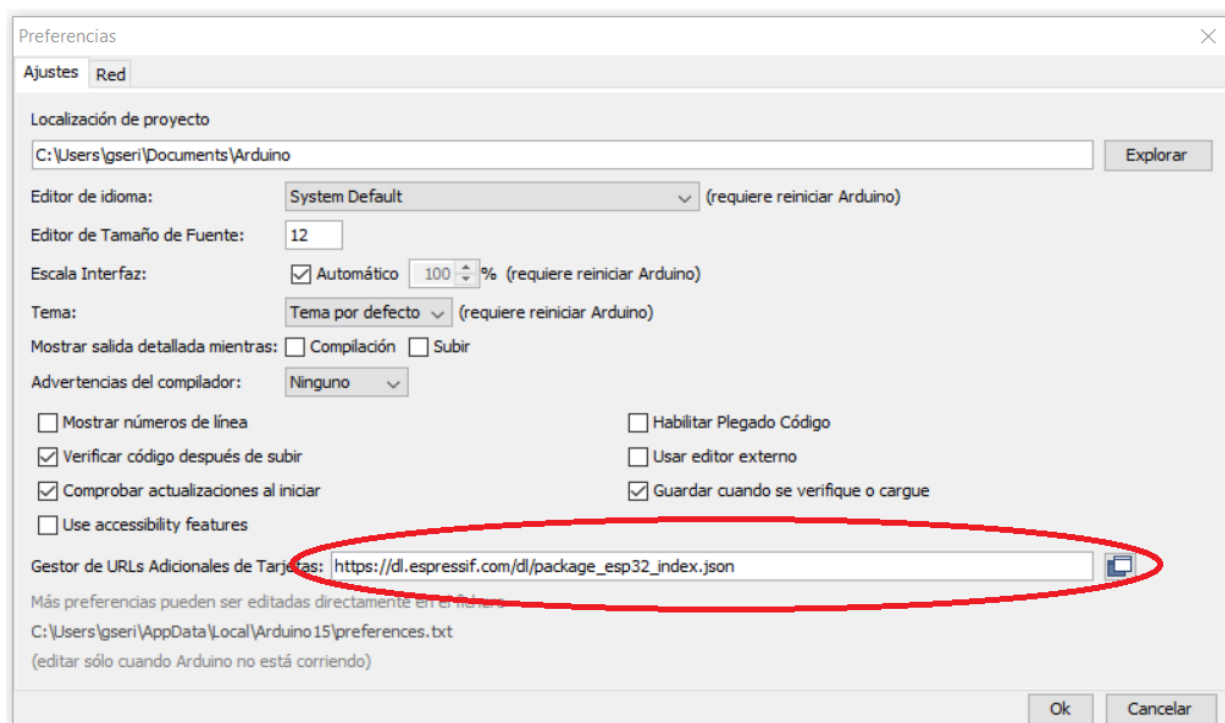
Una vez instalado el IDE Arduino en la computadora, se contará con el entorno necesario para desarrollar los nodos de la red LoRa. A continuación, se deberá preparar el entorno de programación configurando la placa de desarrollo en el IDE y descargando las librerías necesarias para el código.

Para comenzar, es necesario configurar la URL que ha sido proporcionada por Espressif, para que el IDE pueda descargar los datos correspondientes a la placa de desarrollo elegida. (Bertoletti, P. 2019).

https://dl.espressif.com/dl/package_esp32_index.json

Dicha configuración se debe realizar de la siguiente manera:

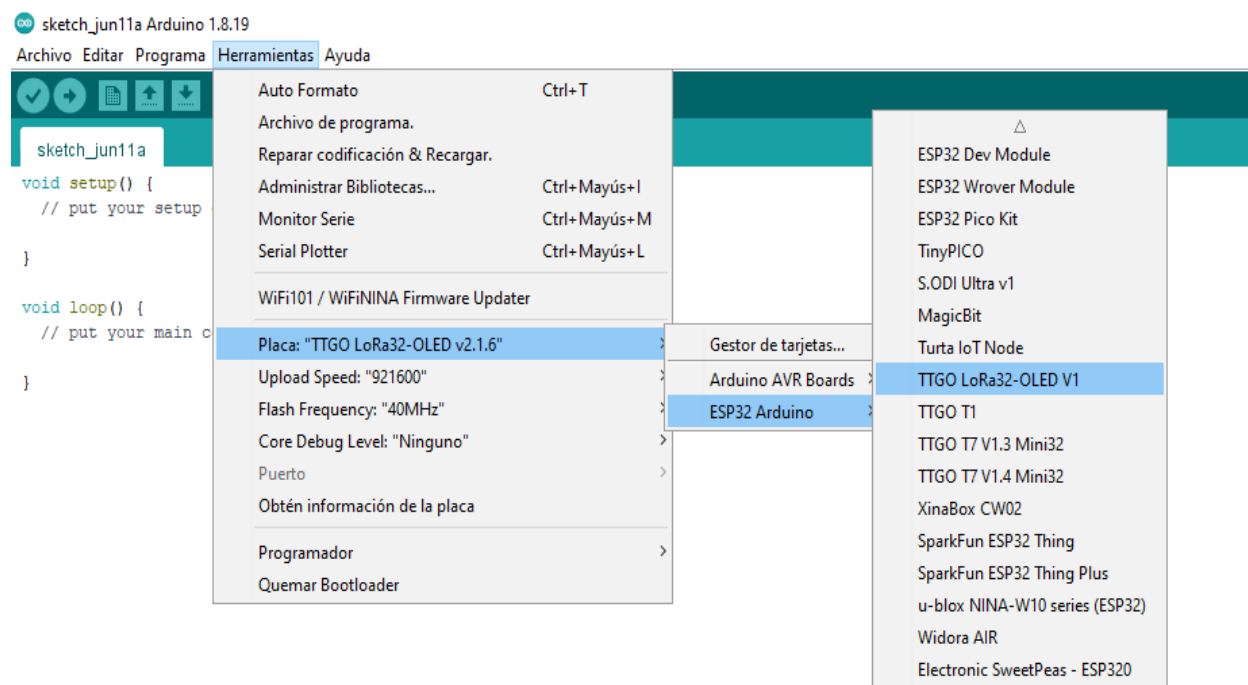
1. En el IDE Arduino seleccionar el menú “Archivo”.
2. Seleccionar la opción “Preferencias”.
3. Pegar la URL anteriormente mencionada en el cuadro de texto indicado en la figura 11.

Figura 11*Configuración IDE Arduino*

3.5.1 Configurando el entorno

Luego de haber realizado la configuración inicial del Entorno de Desarrollo Integrado (IDE) de Arduino, el paso subsiguiente consiste en precisar el modelo de placa de desarrollo que se empleará, con el fin de lograr el guardado correcto del código fuente. Para llevar a cabo esta acción, es necesario dirigirse al menú principal titulado "Herramientas" en la barra de opciones superior del IDE, seleccionar la opción "Placa" y luego optar por "ESP32 Arduino". Dentro de esta elección, es crucial escoger específicamente "TTGO LoRa32-OLED V1", tal como se detalla visualmente en la figura 12 que se presenta en el contexto. (Bertoleti, P. 2019).

Figura 12
Configuración de Placa



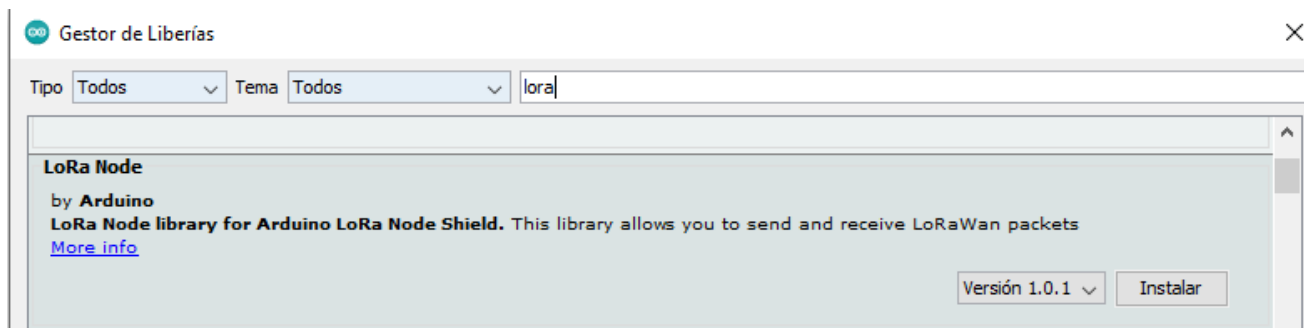
3.5.2 *Instalación de bibliotecas*

Luego de haber configurado la placa de desarrollo, el siguiente paso es instalar las librerías necesarias para utilizar el chip LoRa y el display oled. Las librerías requeridas son: Adafruit_SSD1306.h y LoRa.h

Para incluir dichas librerías se deberán realizar los siguientes pasos en el menú del IDE de Arduino: “Herramientas” - “Administrar Bibliotecas...”, se abrirá una nueva ventana en la cual se deberán buscar ambas librerías por su nombre, luego presionar el botón instalar como se muestra en la figura 13 (Bertoleti, P. 2019).

Figura 13

Instalación de librerías adicionales



3.5.3 Consideraciones previas a la programación

La librería LoRa posee una serie de constantes cuyos valores pueden variar según el hardware utilizado. Estas constantes deberán ser seteadas al comenzar la programación de la placa de desarrollo, los valores podrán ser obtenidos del manual del fabricante como se mencionan a continuación. (Bertoletti, P. 2019).

Constantes del chip LoRa

```
#define SCK 5
#define MISO 19
#define MOSI 27
#define CS 18
#define RST 23
#define DIO 26
```

Para poder operar correctamente la placa de desarrollo se deberán configurar los valores del protocolo SPI, estos son SCLK, MOSI, MISO y CS, los cuales pueden observarse en la figura 14.

3.5.4 Protocolo SPI

SPI cuya sigla significa “Serial Peripheral Interface” es un protocolo de comunicación utilizado en circuitos integrados y microcontroladores. Consta de un bus de cuatro líneas por el cual transmite información entre los diferentes dispositivos conectados. Dos de estas líneas son usadas para transferir datos, mientras que la tercera es la del reloj y la cuarta es de control.

Cada dispositivo puede actuar como transmisor y receptor al mismo tiempo, por lo tanto su comunicación es del tipo full duplex. Los dispositivos que se conectan al bus se definen como maestros y esclavos. El dispositivo maestro inicia la transferencia de información sobre el bus y genera señales de control. El dispositivo esclavo es controlado por el maestro desde una línea selectora conocida como Chip Select (CS). (López, E. s.f.).

Las líneas de comunicación del protocolo SPI se describen de la

SCLK: La señal sobre la línea de reloj es generada por el maestro para sincronizar la transferencia de datos.

MOSI (Master out Slave In): Transporta los datos del maestro hacia el esclavo.

MISO (Master In Slave Out): Transporta los datos del esclavo hacia el maestro.

CS (Chip Select): Por esta línea cada dispositivo maestro controla al dispositivo esclavo. (López, E. s.f.).

3.5.5 Display Oled

En cuanto a la configuración del display de la placa de desarrollo también se deberá configurar una serie de variables. Para que el programa pueda mostrar información por pantalla se deberán especificar las constantes como se muestran a continuación. (Bertoleti, P. 2019).

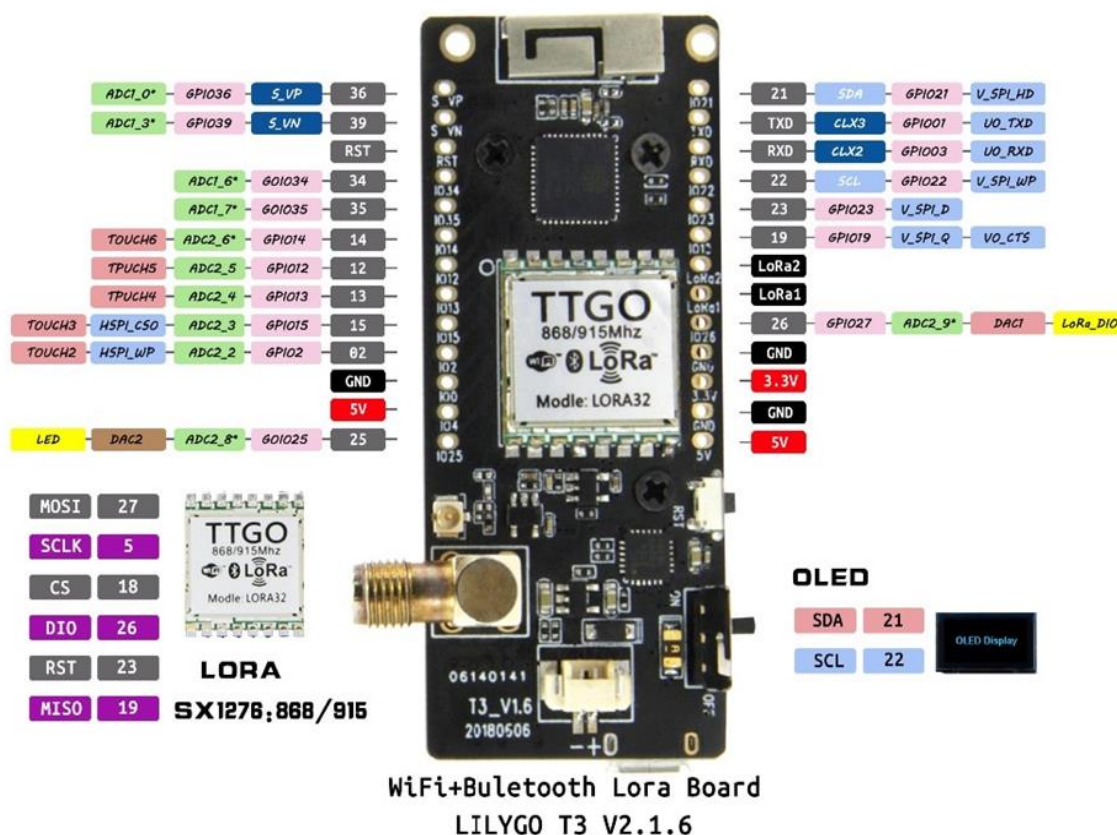
```
#define OLED_SDA 21
#define OLED_SCL 22
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
```

3.5.6 Diagrama de interconexión y pines de programación

Es necesario configurar e inicializar correctamente la placa de desarrollo y sus componentes para asegurar su correcto funcionamiento. Para ello, es recomendable consultar el manual del fabricante y obtener los valores que deben configurarse en cada constante requerida por las diferentes librerías mencionadas. En el caso de la placa de desarrollo TTGO ESP32, la figura 14 del manual muestra la distribución de pines y los valores necesarios para su configuración.

Figura 14

Esquema de Pines Placa de Desarrollo LoRa.



Nota. Imagen tomada de (Shenzhen Xin Yuan electronic technology Co., Ltd., 2018).

3.6 Primer montaje

Con el hardware y el software reunidos, se reúnen las condiciones para montar la primera versión del prototipo. En este primer montaje, se programará una placa de desarrollo como emisor y otra como receptor para realizar pruebas sobre la misma. De esta manera, se podrá establecer la primera conexión LoRa, en la que se medirá su alcance, potencia de señal y duración de la batería en el nodo emisor. Para este montaje, se necesitarán los siguientes componentes:

Hardware

- Placa de desarrollo TTGO ESP32 LoRa x 2.
- Batería para el nodo receptor.
- Antena exterior.
- Cable pigtail para antena exterior.

Software

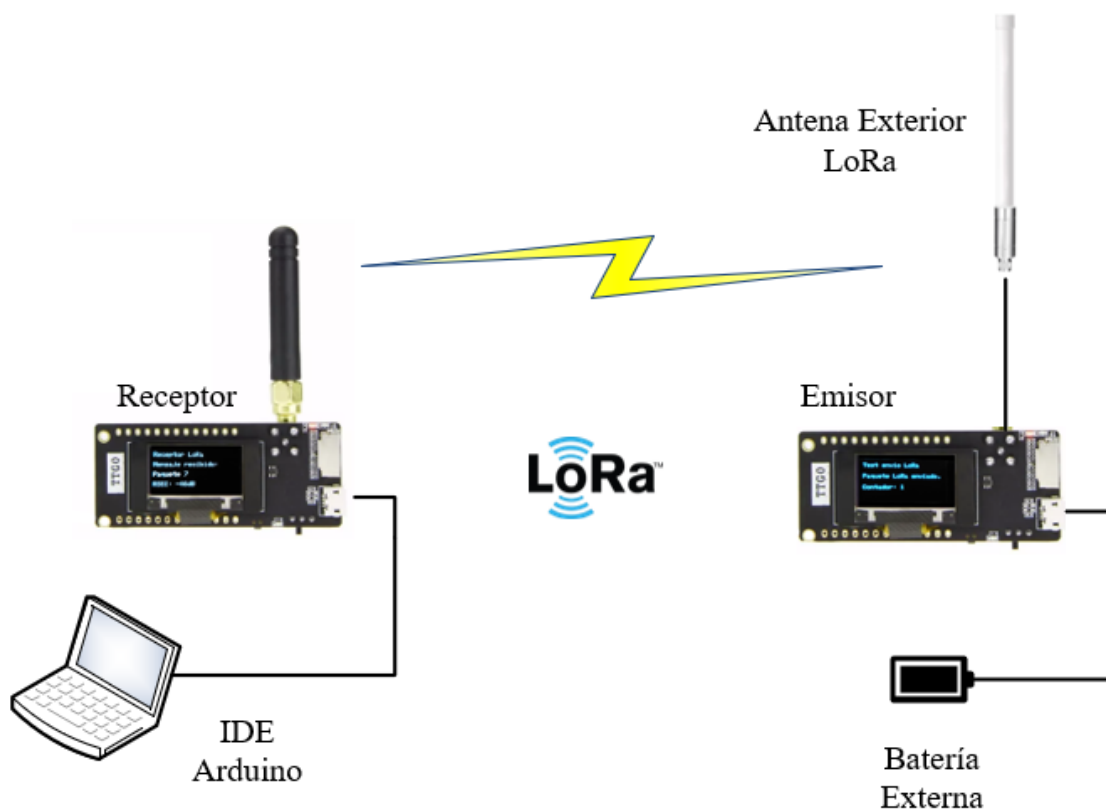
IDE Arduino.

3.6.1 Esquema del primer montaje

En la figura 15 se muestra el primer montaje del prototipo, en el cual se busca medir el alcance máximo de la señal LoRa y la duración máxima de la batería. El nodo emisor estará fijo conectado a una fuente de energía constante y a la antena LoRa exterior, mientras que el nodo receptor será el que tendrá movilidad, y por lo tanto, será alimentado con una batería externa.

Figura 15

Esquema del Primer Montaje



3.6.2 Programación de los nodos

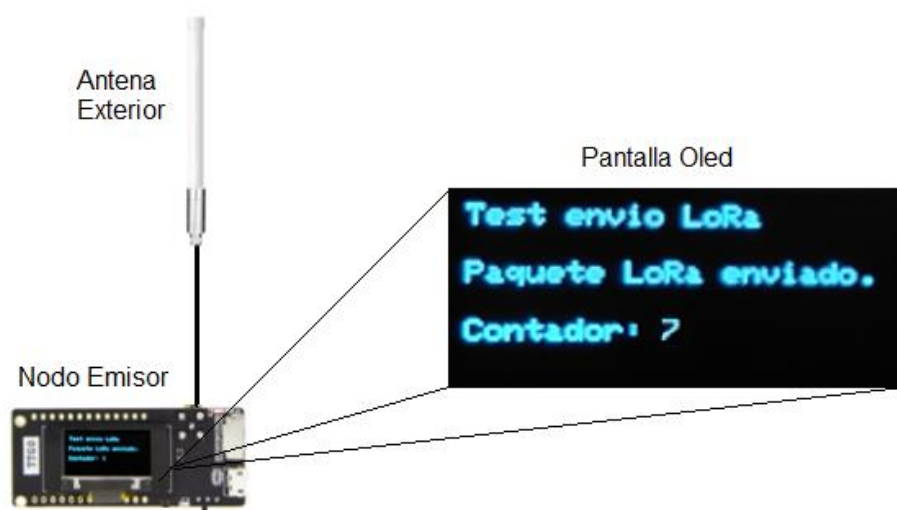
Para que ambos nodos puedan establecer una red LoRa y enviar y recibir paquetes periódicamente, será necesario programar cada placa de desarrollo con su código fuente correspondiente, el cual se puede encontrar en el anexo de este mismo documento junto con su descripción correspondiente. En el capítulo 4 se detallarán las mediciones y resultados de esta prueba, en la cual se establecieron los primeros nodos de la red.

3.6.3 Encendido del primer prototipo

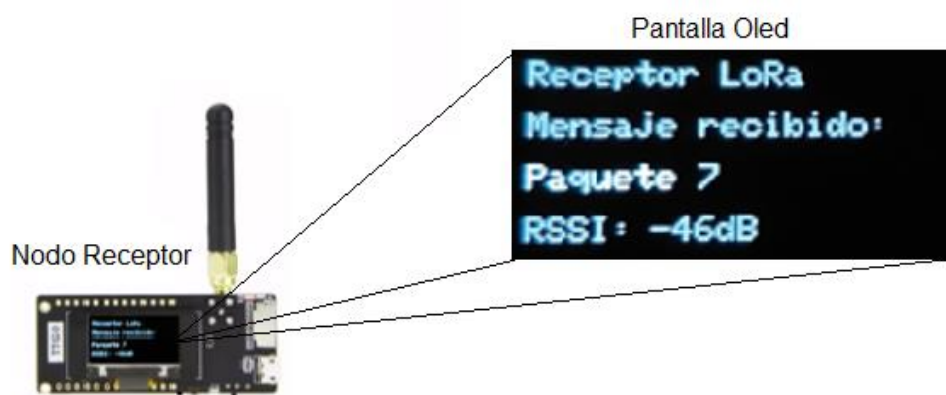
Al contar con ambos nodos programados solo resta encender el emisor y el receptor para comenzar a realizar pruebas sobre la red LoRa. Al encender el nodo emisor se visualizará que el mismo comenzó el envío de paquetes por la red, como puede observarse en la figura 16.

Figura 16

Test Nodo Emisor

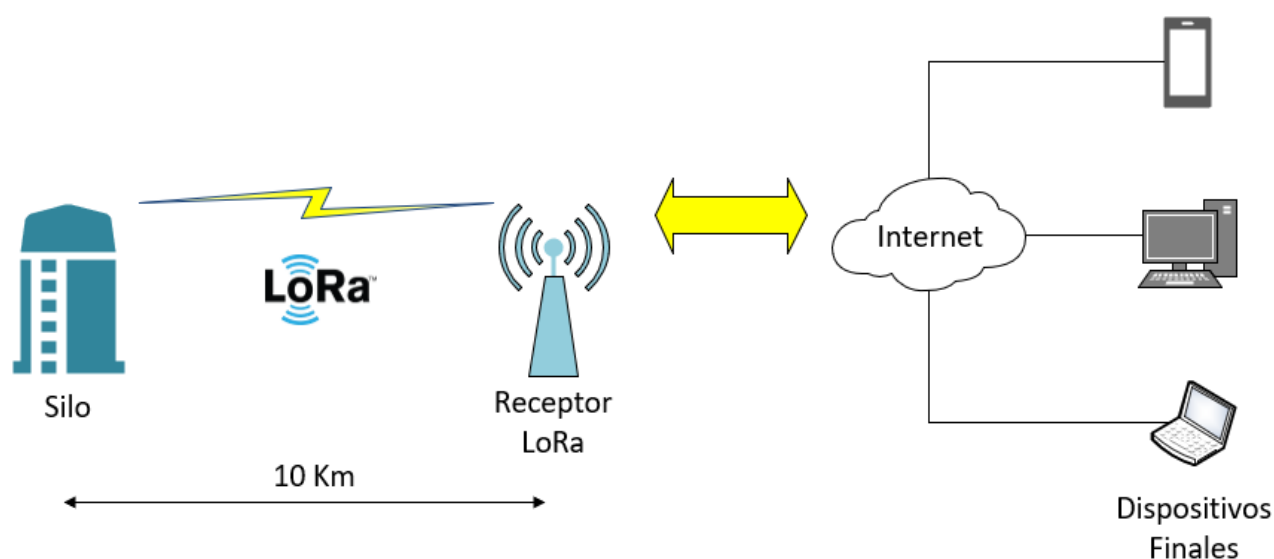


Al encender el nodo receptor, se puede observar que recibe los paquetes correctamente. Además, con la llegada de cada nuevo paquete, también se informa la potencia de la señal RSSI y el SNR del ambiente en el que se desarrolla la prueba, tal como se observa en la figura 17.

Figura 17*Test Nodo Receptor*

3.7 Armado del Prototipo Final

Después de llevar a cabo la investigación teórica de LoRa y crear el primer prototipo emisor/receptor, llega el momento de construir el prototipo final y poner a prueba el uso práctico de la tecnología. En este caso, el prototipo se utilizará para monitorear las variaciones de temperatura de un silo de granos que se encuentra alejado de la red eléctrica y fuera del alcance de la red celular.

Figura 18*Esquema Prototipo Final*

3.7.1 Nodo emisor con sensor de temperatura

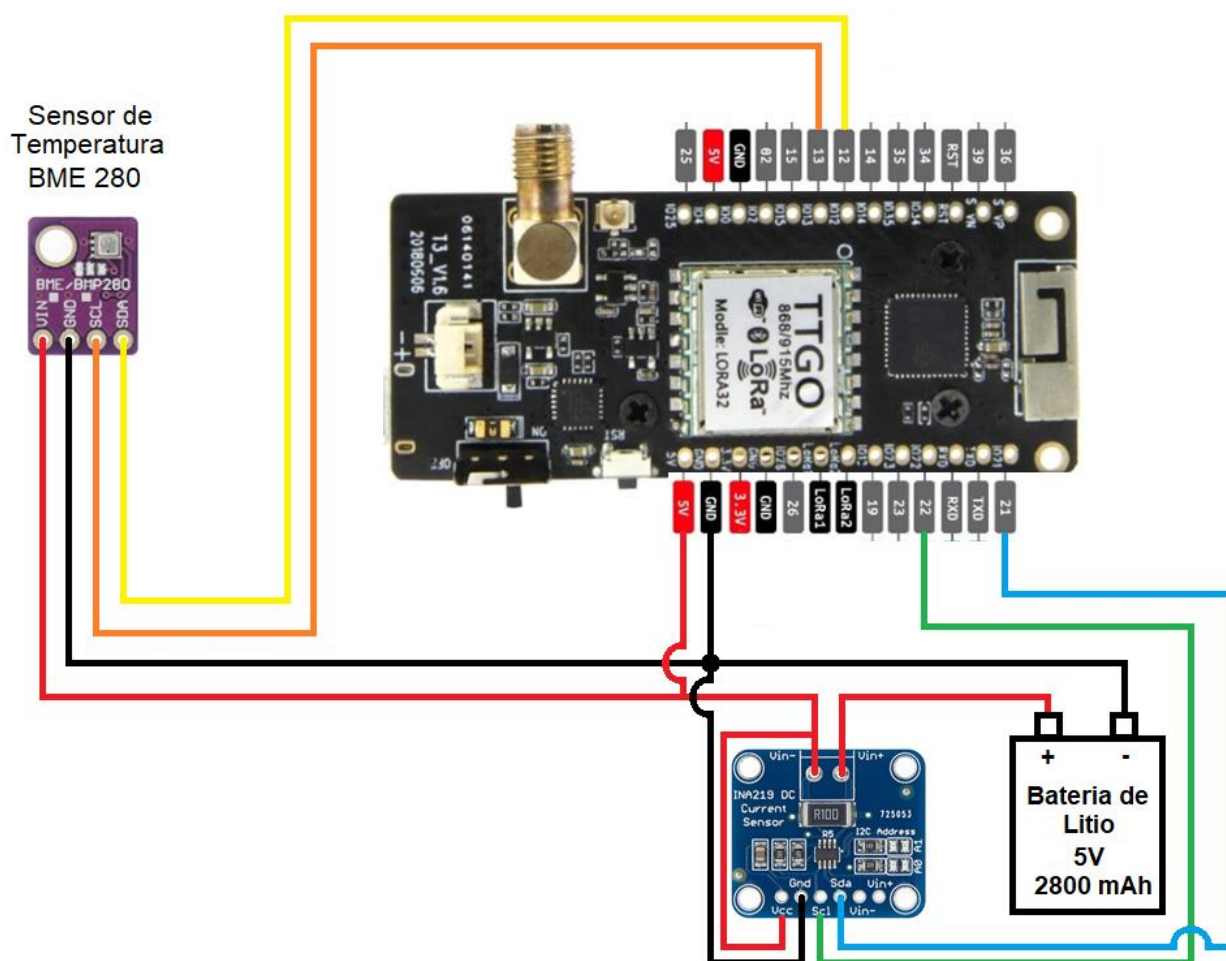
El nodo emisor se montará sobre protoboard y estará conformado por los siguientes componentes:

- Placa de desarrollo con microcontrolador ESP32 y chip LoRa integrado.
- Sensor de temperatura BME 280.
- Antena móvil para exteriores.
- Sensor de descarga de batería ina219.

Este nodo será el que se instalará lejos de toda fuente de energía eléctrica y señal de datos móvil, por lo tanto, será alimentado por la batería de litio. Para extender el tiempo de vida de la batería, el nodo será programado de forma tal que cense la temperatura cada dos horas.

Figura 19

Conexión de Sensor en Nodo de Envío



Este nodo será programado de manera similar a la primera prueba, en la que se enviaron paquetes a través de la red LoRa. Sin embargo, esta vez se modificará su código para incluir la temperatura y el estado de la batería. El código fuente estará disponible en el anexo de este trabajo. Una vez finalizada la programación del nodo emisor, se deberá instalar en el silo y colocar el sensor en su interior para obtener la temperatura del mismo.

3.7.2 Nodo receptor con envío de datos a la nube

El nodo receptor estará conectado a la fuente de energía convencional y tendrá acceso a internet a través de su dispositivo WiFi. Una vez que reciba el valor de la temperatura del silo a través de la red LoRa, lo enviará a una hoja de cálculo de Google Sheets a través de internet. Esta hoja de cálculo almacenará la siguiente información: fecha, hora, valor de temperatura y valor de señal RSSI.

Se optó por almacenar los datos en una hoja de cálculo de Google porque cumple con las necesidades de la prueba. No obstante, esta opción podría mejorarse sustituyendo la hoja de cálculo por un motor de base de datos, lo que proporcionaría una solución más sólida y adecuada para aplicaciones comerciales.

Para generar el informe para el usuario final, los datos serán recopilados por un software de monitoreo gráfico en tiempo real llamado Grafana. Este software presentará los valores actuales e históricos obtenidos por la red LoRa mediante gráficos y tablas.

La programación del nodo receptor se llevará a cabo de manera similar al nodo emisor, utilizando su correspondiente código fuente. Este último se encuentra adjunto en el anexo de este trabajo para facilitar su lectura.

3.7.3 Servidor en la nube

Para instalar Grafana y publicarlo en internet, se eligió la Nube Pública Nacional alojada en el centro de datos de ARSAT. Esta opción proporciona un período de prueba gratuito suficiente para realizar pruebas de recepción de datos y acceso desde diferentes ubicaciones y dispositivos. Grafana se instaló en un servidor virtual con sistema operativo Linux CentOS con las siguientes especificaciones:

- CPU: 4 vCPU's.
- Memoria RAM: 8GB.
- Root Disk: 8 GB.
- Data Disk: 10 GB.
- Sistema Operativo: Linux CentOS 7 x64.

3.8 Resumen del capítulo

En este capítulo se presentan los diferentes componentes de hardware y software que se analizaron cuidadosamente para la creación del prototipo. En primer lugar, se llevó a cabo una investigación exhaustiva para seleccionar los componentes adecuados. Se evaluaron varias opciones disponibles en el mercado, se analizaron las especificaciones técnicas de cada una de ellas, y se consideraron factores como la compatibilidad, la calidad y el costo.

Una vez que se obtuvieron los componentes necesarios, se procedió a realizar una prueba para analizar la fiabilidad y el alcance máximo de la red LoRaWAN. Durante esta prueba, se realizaron mediciones y se evaluaron los resultados obtenidos para determinar si la red cumplía con los requisitos.

Después de completar la primera prueba y asegurarnos de que la red LoRaWAN era confiable y tenía un alcance suficiente, se procedió a la construcción del prototipo final. Esto implicó la programación del hardware para el muestreo de la temperatura de un silo y el envío de los datos a una planilla de datos en línea. Se elaboró un software de monitoreo que permitió acceder a los datos desde cualquier lugar a través de internet. El software de monitoreo se alojó en un centro de datos en línea para garantizar la disponibilidad y la seguridad de los datos.

En resumen, se llevó a cabo un proceso cuidadoso y detallado para la creación del prototipo, que incluyó la selección de componentes adecuados, la realización de pruebas para asegurar la fiabilidad y el alcance de la red, y la construcción del prototipo final. Este proceso garantizó la calidad y la eficacia del prototipo final.

Capítulo 4

4.1 Introducción

En este capítulo se realizarán diversas pruebas sobre los prototipos construidos en el capítulo 3. Para ambos casos se realizaron pruebas de distancia a fin de tomar muestras de intensidad de la señal y obtener su alcance máximo. De esta manera se revisará el funcionamiento de todos sus componentes. Por último, se desarrollarán los resultados y se comprobará el funcionamiento del prototipo.

4.2 Pruebas de sobre el primer montaje

El primer montaje permitió establecer la red LoRaWAN y, a través de ella, enviar paquetes periódicamente. En este punto, se llevaron a cabo las siguientes pruebas.

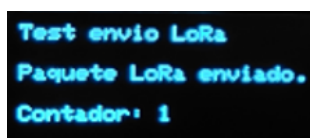
4.2.1 Prueba Nro. 1 Conectividad

Para la primera prueba, se encendieron los nodos emisor y receptor en un entorno de laboratorio para verificar el envío y recepción de paquetes, independientemente de la distancia entre ellos. Acto seguido se logró comprobar el envío y recepción de datos por la red LoRa.

En la pantalla OLED del nodo emisor se muestra que la red LoRa ha iniciado correctamente y también se visualiza el contador de paquetes enviados.

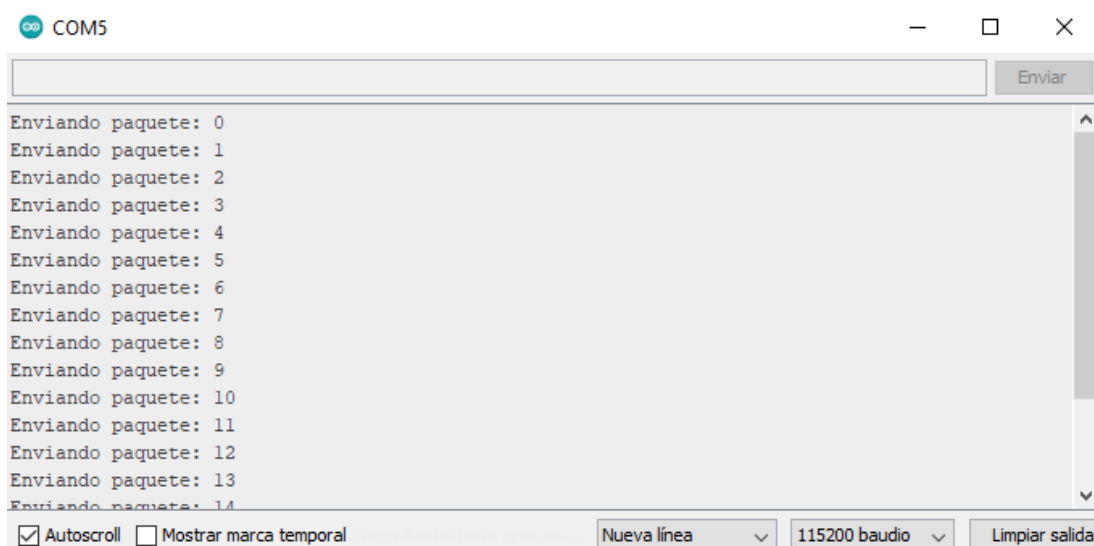
Figura 20

Pantalla Oled Nodo Emisor

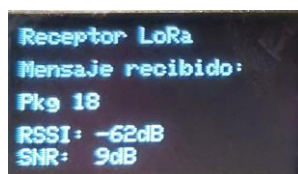


```
Test envio LoRa
Paquete LoRa enviado.
Contador: 1
```

De igual manera, en el monitor serial del IDE Arduino, se puede observar que los datos se han enviado correctamente y que se están enviando periódicamente según el tiempo programado, como se muestra en la figura 21.

Figura 21*Monitor Serial IDE Arduino Nodo Emisor*

El nodo receptor inicia el protocolo de la red LoRa y permanece en modo de escucha indefinidamente hasta que registra la llegada de un paquete. Una vez que recibe un paquete, muestra la confirmación en la pantalla OLED, como se muestra en la figura 22.

Figura 22*Pantalla Oled Nodo Receptor*

Esta prueba permitió verificar el correcto funcionamiento de los nodos según su programación, así como también el envío y recepción de paquetes a través de la red.

4.2.2 Prueba Nro. 2 Recuperación de la red

En el capítulo 2 se destacó que LoRa constituye una red confiable; ante posibles problemas en alguno de sus nodos la red se restablece automáticamente al normalizarse, prescindiendo de intervención humana.

Para esta prueba se utilizaron los nodos emisor y receptor de la misma forma que en la prueba Nro. 1. En esta ocasión, se simuló un corte de energía en cada nodo, como se describe a continuación.

Prueba A

Desconexión de nodo receptor:

En esta prueba, se interrumpió abruptamente la energía del nodo receptor mientras el emisor continuaba enviando paquetes. Esto no afectó de ninguna manera al nodo emisor. Una vez que se reconectó el nodo receptor, este se volvió a conectar automáticamente a la red y comenzó a recibir paquetes nuevamente. Cabe aclarar que los paquetes enviados por el nodo emisor durante la desconexión se perdieron, ya que no había ningún nodo receptor que pudiera recibirlos.

Prueba B

Desconexión del nodo emisor:

En esta ocasión, se procedió a desconectar abruptamente la energía del nodo emisor. Esto provocó que se detuviera el envío de paquetes y que el nodo receptor volviera a su modo de escucha sin sufrir ninguna afectación. Una vez que se restableció el nodo emisor, este reinició el envío de paquetes sin presentar problemas en la reconexión, mientras que el nodo receptor reanudó su tarea de forma normal. Este tipo de pruebas permitió comprobar la fiabilidad de la red en caso de sufrir interrupciones en algunos de sus nodos.

4.2.3 Prueba Nro. 3 Alcance y calidad de señal

En esta prueba, se llevó a cabo la medición del alcance de la red LoRa utilizando los nodos del primer prototipo para obtener su máxima cobertura. Para ello, se instaló una antena exterior multidireccional en el nodo emisor, a una altura de 60 metros para lograr un campo visual amplio. A continuación, se tomó muestras del RSSI y SNR de la señal a medida que se incrementaba la distancia entre ambos nodos, utilizando el nodo receptor alimentado por batería. Los resultados obtenidos se pueden observar en la tabla 4.

Para comprender los resultados de esta prueba, es de vital importancia recordar los valores máximos de RSSI y SNR tolerados por LoRa, los cuales fueron mencionados en el punto 2.2.5. Dichos valores son fundamentales para la evaluación de los resultados obtenidos en la prueba.

RSSI mayor o igual que -120db y SNR mayor o igual a -20db

Tabla 4*Pruebas de Calidad de Señal*

Test Nro.	Distancia (Metros)	RSSI (DB)	SNR (DB)
1	50	- 60	9
2	100	- 70	8
3	500	- 86	5
4	1000	-90	2
5	2000	-94	0
6	3000	-98	0
7	4000	-102	-1
8	5000	-106	-3
9	6000	-110	-6
10	7000	-114	-8
11	8000	-118	-11
12	9000	-120	-14
13	10000	-122	-16

Aunque se ha mencionado que el valor máximo tolerado de RSSI por la red LoRa es de 120 dB, se ha comprobado que al alcanzar la distancia máxima entre los nodos, dicho valor variaba entre 120 dB y 122 dB. Cabe mencionar que el desvío en el valor de RSSI no es significativo, ya que los paquetes seguían llegando al receptor de forma constante, sin pérdida de datos.

4.3 Pruebas sobre el montaje final

En las siguientes pruebas, se procedió a verificar el funcionamiento del prototipo final, con el fin de observar su desempeño en el escenario real. En estas pruebas, también se tuvieron en cuenta los resultados obtenidos con el primer prototipo, en cuanto a distancia y calidad de señal.

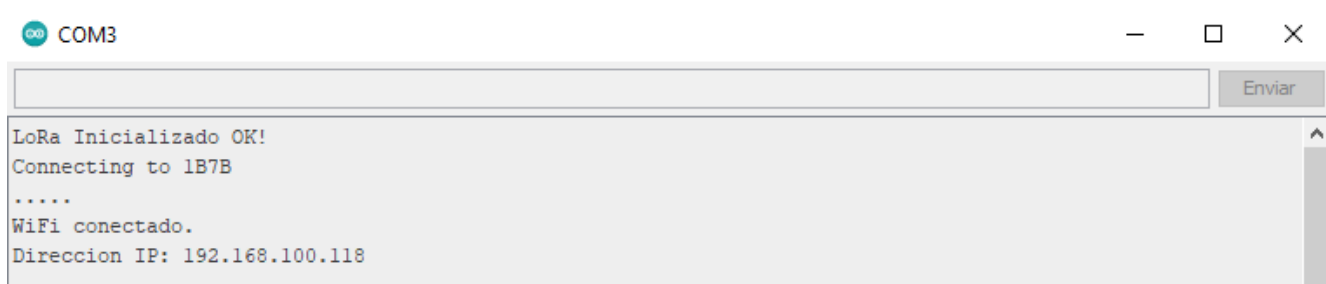
4.3.1 Prueba de conectividad

En esta prueba, se encendió el prototipo final para obtener las lecturas de temperatura del silo de granos. Una vez obtenidas por el nodo receptor, este estableció la conexión a Internet

mediante su dispositivo WiFi. Con la red LoRa y la conexión a Internet establecidas (ver figura 23), el nodo receptor quedó listo para enviar los datos obtenidos a la planilla de Google Sheets.

Figura 23

Conexión LoRa y WiFi en Nodo Receptor

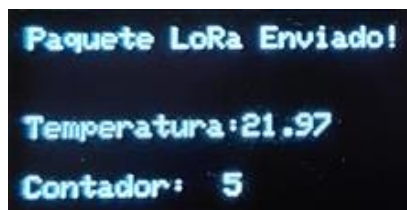


4.3.2 Obtención de lecturas del sensor

Una vez encendido el nodo emisor y establecida la red LoRa, se muestra en la pantalla OLED la temperatura capturada, junto con la confirmación del envío y el número de lecturas realizadas.

Figura 24

Lectura de Temperatura



En el monitor serial del IDE Arduino, se obtienen también los resultados de la temperatura del silo con su correspondiente lectura de RSSI.

Figura 25

Recepción de temperatura y RSS

```
2022/06/24 22:44:27 Paquete LoRa Recibido: Temp: 21.97 with RSSI -15
```

4.3.3 Prueba almacenado de datos

En esta prueba se corroboró el correcto envío de datos a la planilla de datos de Google Sheets, cómo se puede apreciar en las figuras 26 y 27.

Figura 26

Envío de Datos a Planilla

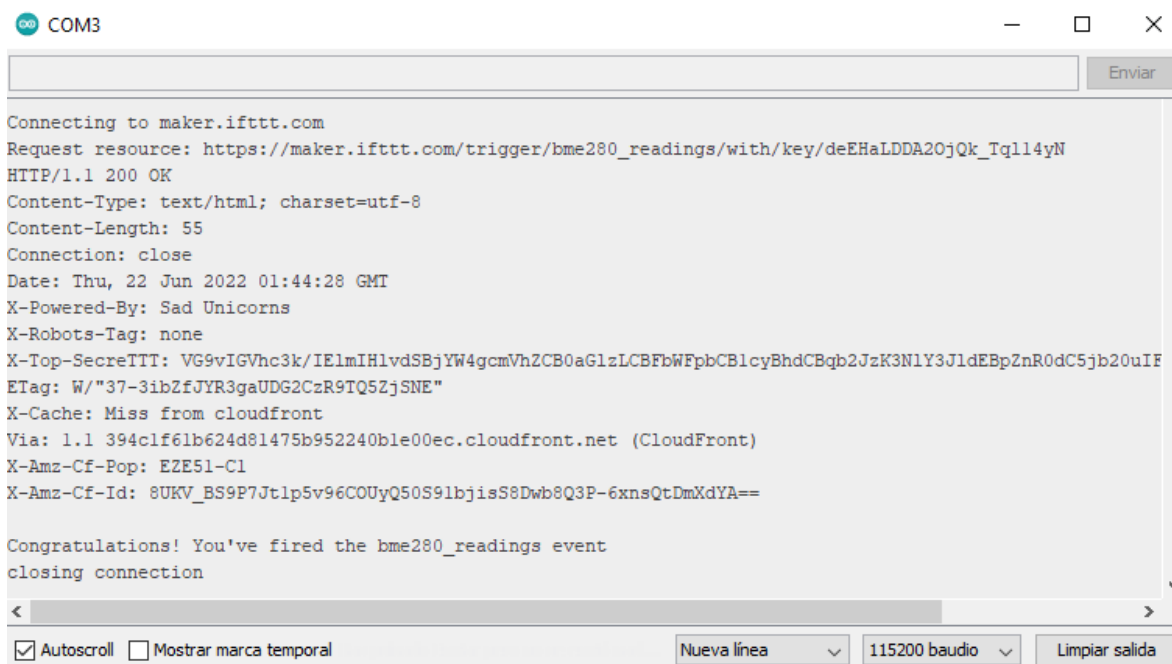


Figura 27

Recepción de Datos en Planilla

BME280_Readings ☆ 📁 ☁

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones

100% \$ % .0 .00 123 Predetermi... 10 B

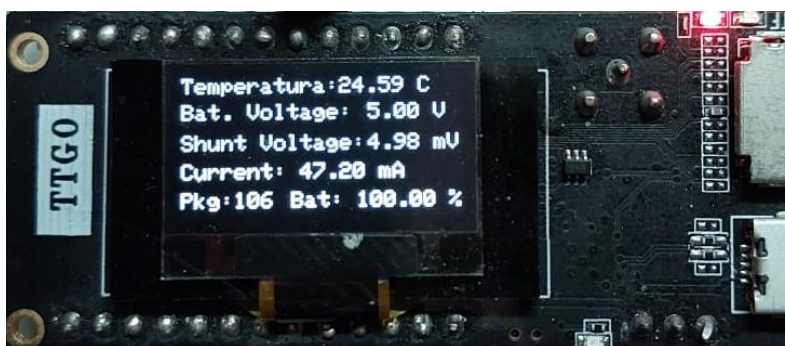
	A	B	C	D
1	Fecha	Temperatura	Rssi	Batería (V)
1046	June 24, 2022 at 07:46PM	22.74	-20	4.97
1047	June 24, 2022 at 07:47PM	22.75	-21	4.97
1048	June 24, 2022 at 07:47PM	22.74	-20	4.96
1049	June 24, 2022 at 10:44PM	21.97	-15	4.95
1050	June 24, 2022 at 10:44PM	21.93	-14	4.96
1051	June 24, 2022 at 10:44PM	21.9	-14	4.96
1052	June 24, 2022 at 10:44PM	21.9	-35	4.95
1053	June 24, 2022 at 10:45PM	21.92	-35	4.96
1054	June 24, 2022 at 10:45PM	21.88	-13	4.96

4.3.4 Prueba de duración de la batería

En esta prueba se midió la descarga de la batería del nodo emisor ubicado en el silo de granos, el cual estaba programado para obtener y enviar la lectura cada hora. La duración de la batería se calculará basándose en el consumo de todos los dispositivos que conforman el nodo emisor. En la figura 28 se muestra el voltaje actual de la batería de 5v, la caída de tensión en la resistencia shunt que es de 4,98 mV y la corriente consumida por el nodo emisor en cada lectura es de 47,2 mA.

Figura 28

Medición de Consumo



4.3.5 Monitoreo e informe de datos para usuario final

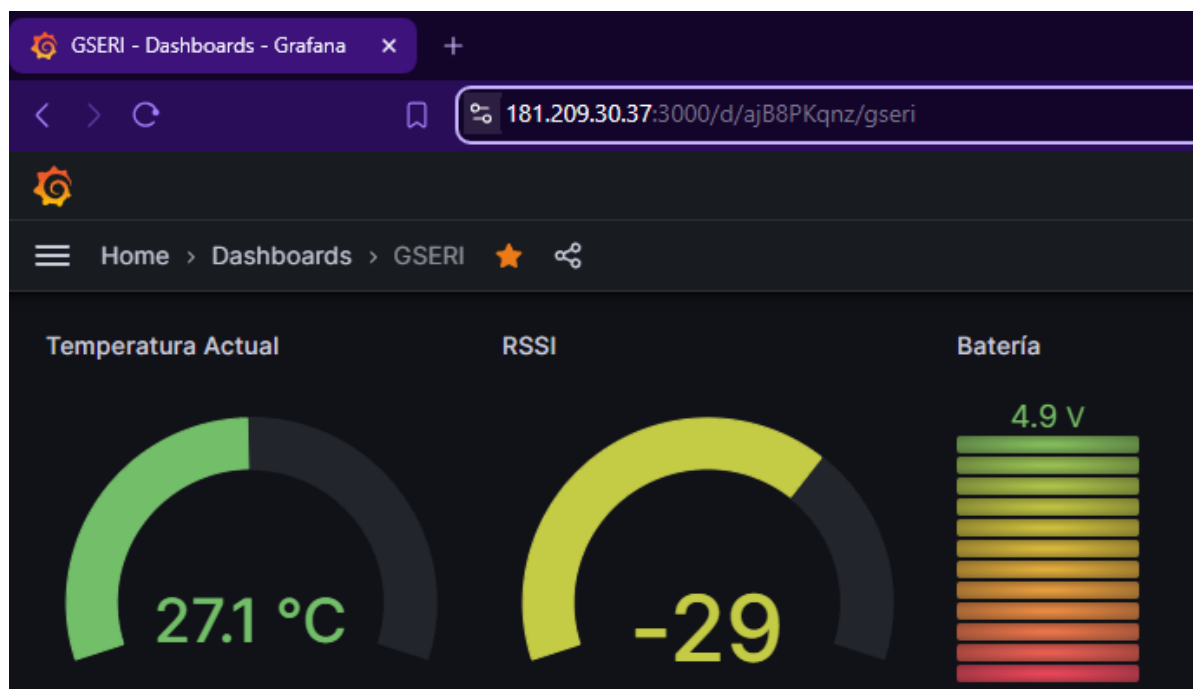
En el contexto de la evaluación definitiva, se llevó a cabo la configuración integral del servidor de Grafana con el fin de recolectar los datos almacenados en la planilla de Google Sheets y mostrarlos de forma gráfica en un portal web. De esta manera, es posible visualizar el último valor de temperatura registrado, el RSSI de la señal LoRa y el estado de la batería.

La culminación de este proceso se materializa en la Figura 29, donde se representa en forma visual fácilmente interpretable en el tablero de Grafana configurado para tal fin.

En resumen, la fase de la evaluación final no solo demuestra la integración exitosa entre el servidor Grafana y la hoja de cálculo Google Sheets, sino también la habilidad para transformar datos estáticos en un recurso dinámico y eficaz para la toma de decisiones, a través de una representación visual clara y accesible.

Figura 29

Representación Gráfica de Resultados en Grafana



4.4 Síntesis

En este capítulo se llevaron a cabo distintos tipos de pruebas con el fin de corroborar el funcionamiento de cada módulo por separado y en el escenario final. En primer lugar, se realizaron pruebas en entornos de laboratorio para verificar la conectividad. Posteriormente, se midió el alcance de la red LoRa y se simuló pruebas de conectividad en el escenario final. Luego, se realizaron pruebas detallando el funcionamiento del prototipo final paso a paso, desde la obtención del dato censado hasta su publicación en una página web en forma gráfica. Estas pruebas permitieron verificar el correcto funcionamiento del prototipo final en condiciones reales de uso. En cada una de las pruebas realizadas se obtuvieron resultados detallados y se presentaron evidencias gráficas que permiten evaluar el desempeño del prototipo y de cada uno de sus módulos. De esta manera, se logró validar la eficacia y la eficiencia del prototipo, lo que demuestra la viabilidad de su uso en aplicaciones reales.

Capítulo 5

5.1 Introducción

En este capítulo se presentan las conclusiones obtenidas a partir de las pruebas realizadas en el capítulo 4, en las que se comprobó que el prototipo de red LPWAN LoRa es capaz de recolectar y transmitir datos de forma eficiente y confiable. Además, se verificó la hipótesis planteada al inicio del trabajo. Estas conclusiones son de gran relevancia para el diseño de próximos pasos para la optimización del prototipo con todos sus componentes.

5.2 Conclusiones

El prototipo desarrollado en este trabajo de investigación permitió mostrar cómo establecer una red LoRa mediante la construcción de cada uno de sus nodos. Los resultados obtenidos en las pruebas permitieron confirmar la hipótesis inicial demostrando que la tecnología LoRaWAN es una solución efectiva para el monitoreo y control en áreas rurales, especialmente en aplicaciones agrícolas donde no existe una fuente constante de energía eléctrica. Esto es debido a que los sensores compatibles con LoRa son pequeños y en su mayoría requieren baja energía para funcionar lo cual los hace ideales para ser instalados en diferentes ubicaciones empleando baterías de larga duración.

Un punto importante a destacar es que la construcción de una red LPWAN LoRa no requiere materiales costosos ni difíciles de obtener. Por otro lado la implementación de un servidor virtual alojado en la nube demostró ser una elección acertada ya que reduce significativamente los costos de adquisición y mantenimiento al proporcionar un servicio independiente del hardware.

Durante el desarrollo del prototipo surgió la necesidad de explorar el mercado de placas de desarrollo con microcontroladores, particularmente el ESP32 que fue diseñado para proyectos de IoT, esto facilitó la integración de módulos y su programación. Sin embargo es importante destacar que estos descubrimientos no convierten al prototipo en una solución definitiva. Después de su construcción y prueba, surge un nuevo desafío: investigar los últimos avances tecnológicos que podrían mejorar aún más la eficiencia y confiabilidad de las redes LoRaWAN.

Las pruebas del prototipo no solo sirvieron para demostrar la hipótesis inicial sino que además permitió estudiar el comportamiento de la señal LoRa en diferentes ambientes. Al probar el prototipo en entornos de laboratorio se observó que la señal mostraba una variación constante en su RSSI, esto es debido al rebote de la onda en los objetos presentes en la habitación. Sin embargo en pruebas al aire libre en entornos rurales la señal se mantuvo estable y alcanzó su umbral máximo a una distancia de 10 km entre los nodos.

Durante el trabajo se investigaron diversas tecnologías de redes inalámbricas, según se ha podido comprobar no existe una tecnología mejor que otra, sino que cada una se adapta a necesidades específicas según los requisitos de cada proyecto. La tecnología LoRaWAN tiene un gran potencial en aplicaciones de monitoreo y control en diversas áreas, como la agricultura, la industria, el transporte, entre otras. Esto la convierte en una tecnología muy prometedora para el futuro.

Un punto importante a destacar es su seguridad. Durante las pruebas del prototipo se implementó el cifrado de extremo a extremo integrado con la red LoRa, lo que garantiza la transmisión segura de los datos impidiendo que sean interceptados por posibles atacantes. Aunque no se llevaron a cabo ensayos específicos para poner a prueba este tipo de encriptación, este descubrimiento plantea una nueva interrogante en cuanto a la seguridad. Se necesitarán medidas adicionales para mitigar interferencias y posibles ataques cibernéticos

5.3 Síntesis

El estudio realizado ha confirmado de manera contundente que la implementación de la tecnología LoRaWAN ofrece una solución efectiva y económica para el monitoreo y control de silos ubicados en entornos rurales sin acceso convencional a la energía eléctrica. Los resultados obtenidos respaldan la viabilidad de esta tecnología, destacando su accesibilidad y la facilidad de configuración de sus nodos.

Capítulo 6

6.1 Introducción

Este capítulo se enfocará en los aspectos a mejorar y/o corregir en el prototipo que fue construido y probado en los capítulos anteriores. Se describirán los problemas encontrados durante las pruebas y se discutirán posibles soluciones. Además, se propondrán mejoras para optimizar el desempeño del prototipo.

6.2 Recomendaciones

6.2.1 *Obtener buena señal con LoRa*

Como se mencionó anteriormente, las pruebas de la red LoRa se llevaron a cabo utilizando una antena exterior que se especifica en el punto 3.2.3. Si bien esta antena fue adecuada para demostrar la hipótesis, existen otras antenas disponibles en el mercado con un alcance mayor. Estas antenas aumentan su ganancia de señal a medida que aumenta su tamaño, es decir, que la calidad de la señal LoRa es directamente proporcional al tamaño de la antena. Por lo tanto, sería recomendable explorar otras opciones de antenas para mejorar la cobertura y la calidad de la señal de la red LoRa.

6.3 Trabajos Futuros

6.3.1 *Implementación de base de datos*

Este trabajo se enfocó en la investigación de la tecnología de comunicación inalámbrica LoRa. Sin embargo, durante el desarrollo del prototipo, fue necesario disponer de un medio de almacenamiento de datos para registrar históricamente las mediciones de temperatura. Para ello, se utilizó temporalmente una hoja de cálculo de Google. Aunque esta solución fue útil para el entorno de pruebas, no es recomendable para una implementación final, ya que no es escalable en el tiempo y tiene un límite de registros relativamente bajo.

Para lograr una solución final robusta y confiable, se recomienda implementar un motor de base de datos capaz de almacenar un gran número de registros. Este motor de base de datos permitirá trabajar con más de un servidor, lo que proporcionará un modelo de alta disponibilidad con tolerancia a fallos. De esta forma, se podrá garantizar la escalabilidad y confiabilidad de la solución en el tiempo.

6.3.2 Ahorro de energía

Se configurará el nodo emisor para que luego de su inicio deje la pantalla oled apagada. La idea es que el nodo que censa la temperatura tenga por default el display apagado para ahorrar energía, al presionar un botón aparecerá por un lapso breve el valor actual en pantalla y luego volverá a apagarse.

6.3.3 Implementación de botón de información

En la configuración del nodo emisor se incluirá una modificación para que la pantalla OLED se apague automáticamente después de su inicio. El objetivo principal de esta modificación es ahorrar energía y prolongar la vida útil de la batería. En caso de ser necesario visualizar el valor actual de la temperatura, se podrá activar la pantalla OLED de forma manual mediante el uso de un botón específico. Este botón estará diseñado para que, al presionarlo, el valor actual de la temperatura aparezca en la pantalla durante un corto período de tiempo, lo suficiente para ser leído, y luego vuelva a apagarse automáticamente para ahorrar energía. De esta manera, se podrá obtener la información necesaria sin comprometer la duración de la batería y sin necesidad de mantener la pantalla OLED constantemente encendida.

6.3.4 Pruebas exhaustivas sobre la seguridad

Se deberán llevar a cabo pruebas que simulen escenarios que vayan más allá del entorno inicialmente utilizado. Por ejemplo, podrían realizarse pruebas en entornos urbanos, suburbanos para evaluar cómo el cifrado responde a diferentes condiciones y posibles amenazas.

Además, sería importante considerar la posibilidad de realizar pruebas de intrusión para intentar identificar vulnerabilidades específicas en el cifrado y en el protocolo de comunicación LoRa. Estas pruebas podrían incluir intentos de interceptación de datos, ataques de denegación de servicio u otros tipos de ataques cibernéticos que podrían comprometer la seguridad de la red LoRa.

Anexos

A continuación, se incluye el código que fue empleado para la programación de los nodos en la red LoRa, tal como se detalló en el capítulo 3. Cada segmento del código va acompañado de un breve comentario diseñado para facilitar el seguimiento y la comprensión de su funcionamiento.

Programación del nodo emisor

Agregar librerías necesarias para el manejo del chip LoRa y la pantalla oled. (Bertoletti, P. 2019).

```
//Librería LoRa
#include <LoRa.h>

//Librería para display Oled
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

Definición de valores para la placa de desarrollo. (Bertoletti, P. 2019).

```
//Pines según placa de desarrollo LoRa
#define MOSI 27
#define SCK 5
#define CS 18
#define DIO 26
#define RST 23
#define MISO 19

//Pins OLED según placa de desarrollo
#define OLED_SDA 21
#define OLED_SCL 22
#define OLED_RST 16
#define SCREEN_WIDTH 128 // Ancho de display OLED en pixeles
#define SCREEN_HEIGHT 64 // Alto de display OLED en pixeles
```

Definición de frecuencia LoRa para América.

```
#define BAND 915E6
```

Inicializar el contador de paquetes y la pantalla oled de la placa de desarrollo, también se configura el tamaño y color de la fuente a mostrar y su posición en pantalla. (Bertoletti, P. 2019).

```

int count = 0;
Adafruit_SSD1306 display (SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RST);

void setup () {
  //Reset display OLED via software
  pinMode (OLED_RST, OUTPUT);
  digitalWrite (OLED_RST, LOW);
  delay (20);
  digitalWrite(OLED_RST, HIGH);

  //Inicializar display OLED
  Wire.begin(OLED_SDA, OLED_SCL);
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3c, false, false)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); }

  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setTextSize(1);
  display.setCursor(0,0);
  display.print("Test Envío LoRa ");
  display.display();

```

Inicializar el monitor serial del IDE Arduino. Esto permitirá controlar el funcionamiento del módulo desde el mismo IDE sin la necesidad de observar la pantalla oled de la placa de desarrollo. También permitirá copiar valores que serán de utilidad para demostrar los resultados en forma gráfica. (Bertoletti, P. 2019).

```

//Inicializando monitor serial
Serial.begin(115200);
Serial.println("Test envío LoRa ");

```

Inicializar el chip LoRa con los valores definidos anteriormente. (Bertoletti, P. 2019).

```

//SPI LoRa pins
SPI.begin(SCK, MISO, MOSI, SS);

//Inicializar módulo LoRa
LoRa.setPins(SS, RST, DIO);
Serial.println("Inicialización LoRa OK!");
display.setCursor(0,10);
display.print("Inicialización OK!");
display.display();
delay(5000);}

```

En esta fase final del código, se da inicio al ciclo de transmisión de paquetes a través de la infraestructura de red LoRa. Con fines de validación práctica, se ha configurado el envío de paquetes cada 5 segundos. Además, se brinda información en la pantalla OLED sobre la cantidad de paquetes que han sido enviados. (Bertoleti, P. 2019).

```
void loop() {
  Serial.print("Enviando paquete: ");
  Serial.println(count);

  //Send LoRa packet to receiver
  LoRa.beginPacket();
  LoRa.print("Paquete ");
  LoRa.print(count);
  LoRa.endPacket();
  display.clearDisplay();
  display.setCursor(0,0);
  display.println("Test envío LoRa ");
  display.setCursor(0,20);
  display.setTextSize(1);
  display.print("Paquete LoRa enviado.");
  display.setCursor(0,40);
  display.print("Contador:");
  display.setCursor(60,40);
  display.print(count);
  display.display();

  count++;
  delay(5000);}

```

Programación del nodo receptor

En esta sección, se introduce una variable adicional de tipo String, destinada a contener el paquete recibido. Acto seguido, se inicia un bucle de ejecución constante diseñado para recibir y analizar el paquete en cuestión, con el propósito de posteriormente presentarlo en la pantalla de visualización.

Nota: La sección inicial del código, que abarca la configuración de las librerías, la definición de los parámetros del protocolo SPI y la inicialización de la pantalla OLED, se omitirá en esta instancia dado que es idéntica al código implementado en el nodo emisor. A continuación, se presenta la porción del código específica del nodo receptor, la cual difiere de la implementación en el nodo emisor.

```

String LoRaData;

void loop() {

    //try to parse packet
    int packetSize = LoRa.parsePacket();
    if (packetSize) {
        //received a packet
        Serial.print("Received packet ");

    //Leer Paquete
    while (LoRa.available()) {
        LoRaData = LoRa.readString();
        Serial.print(LoRaData); }

    //Mostrar RSSI
    int rssi = LoRa.packetRssi();
    Serial.print(" with RSSI ");
    Serial.println(rssi);

    // Mostrar información en pantalla Oled
    display.clearDisplay();
    display.setCursor(0,0);
    display.print("Receptor LoRa");
    display.setCursor(0,15);
    display.print("Mensaje recibido:");
    display.setCursor(0,30);
    display.print(LoRaData);
    display.setCursor(0,45);
    display.print("RSSI:");
    display.setCursor(35,45);
    display.print(rssi);
    display.print("dB");
    display.display(); }
}

```

En cada ocasión en la que se reciba un paquete, se hará una presentación del valor correspondiente al RSSI de la señal LoRa. Esta información reviste una importancia crucial, ya que nos brinda el medio para evaluar el deterioro de la señal a medida que el nodo emisor se distancia del receptor. Mediante este enfoque, se adquiere la capacidad de determinar el alcance máximo alcanzable por la red LoRa. (Bertoletti, P. 2019).

Nodo emisor con sensor de temperatura

Incluir las librerías necesarias para que el IDE compile y ejecute el código correctamente. (Bertoleti, P. 2019).

```
//Librerías para LoRa
#include <SPI.h>
#include <LoRa.h>

// Librerías para Display OLED
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Librerías para BME280
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
```

Definir las variables para configurar el modo sleep. (Bertoleti, P. 2019).

```
#define uS_TO_S_FACTOR 1000000 /* Factor de conversión en Microsegundos */
#define TIME_TO_SLEEP 3540 /* Duración del modo sleep en segundos */
```

Definición de valores para los pines de la placa de desarrollo. (Bertoleti, P. 2019).

```
//Pines según placa de desarrollo LoRa
#define MOSI 27
#define SCK 5
#define CS 18
#define DIO 26
#define RST 23
#define MISO 19

//Pins OLED según placa de desarrollo
#define OLED_SDA 21
#define OLED_SCL 22
#define OLED_RST 16
#define SCREEN_WIDTH 128 // Ancho de display OLED en pixeles
#define SCREEN_HEIGHT 64 // Alto de display OLED en pixeles
```

Definición de frecuencia LoRa para América y configuración de los pines del sensor de temperatura con los pines de la placa de desarrollo. (Bertoleti, P. 2019).

```
#define BAND 915E6

//Parámetros del Sensor BME280
#define SDA 15
#define SCL 13
```

Crear una instancia para el sensor BME 280. (Bertoleti, P. 2019).

```
TwoWire I2Cone = TwoWire(1);
Adafruit_BME280 bme;
```

Crear la variable para almacenar la temperatura y configurar el contador de paquetes. (Bertoleti, P. 2019).

```
//Contador de Paquetes
int readingID = 0;
int counter = 0;
String LoRaMessage = "";
float temperature = 0;
```

Definiremos una función destinada a la inicialización del display OLED en la placa de desarrollo. Dicha función se encargará de borrar el contenido previo del display, posicionar el cursor en su punto de partida y presentar el primer mensaje al arrancar el programa. (Bertoleti, P. 2019).

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

//Inicializar display Oled
void startOLED(){
  pinMode(OLED_RST, OUTPUT);
  digitalWrite(OLED_RST, LOW);
  delay(20);
  digitalWrite(OLED_RST, HIGH);
  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(0,0);
  display.print("LORA SENDER");}
```

El código siguiente establece una función diseñada para inicializar el módulo LoRa, configurando las variables SPI y generando impresiones tanto en el display como en el monitor serie del entorno de desarrollo Arduino IDE para presentar los resultados. (Bertoleti, P. 2019).

```
//Inicializar Modulo LoRa
void startLoRA(){
  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DIO);

  Serial.println("Inicializando LoRa OK!");
  display.setCursor(0,10);
  display.clearDisplay();
  display.print("Inicializando LoRa OK!");
  display.display();
  delay(2000);
}
```

Crear la función para guardar el dato obtenido del sensor en la variable 'temperature'.

```
void getReadings(){
    temperature = bme.readTemperature(); }
```

Almacenar el valor de la temperatura en la variable 'LoRaMessage' y transmitirlo a través de la red. Simultáneamente, en el display OLED y en el monitor serial del entorno de desarrollo Arduino IDE, se mostrará la confirmación del envío del paquete, el valor de la temperatura y el número correspondiente al paquete transmitido. (Bertoletti, P. 2019).

```
void sendReadings() {
    LoRaMessage = temperature;

    LoRa.beginPacket();
    LoRa.print(LoRaMessage);
    LoRa.endPacket();

    display.clearDisplay();
    display.setCursor(0,0);
    display.setTextSize(1);
    display.print("Paquete LoRa Enviado!");
    display.setCursor(0,30);
    display.print("Temperatura:");
    display.setCursor(72,30);
    display.print(temperature);
    display.setCursor(0,50);
    display.print("Contador:");
    display.setCursor(66,50);
    display.print(readingID);
    display.display();

    Serial.print("Enviando paquete: ");
    Serial.println(readingID);
    readingID++;
}
```

Dentro de la sección 'void setup()', se lleva a cabo la inicialización de los módulos correspondientes al display OLED, el sensor de temperatura y el chip LoRa. Además, se efectúa una multiplicación entre el valor de la variable 'TIME_TO_SLEEP' y el contenido de 'uS_TO_S_FACTOR', dado que el período de tiempo durante el cual el módulo deberá permanecer en modo de espera requiere ser configurado en microsegundos. (Bertoletti, P. 2019).

```
void setup() {  
  Serial.begin(115200); //Inicializar monitor Serial  
  startOLED();  
  startBME();  
  startLoRA();  
  esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR);}
```

Finalmente, en la sección 'void loop()', se invocarán las funciones encargadas de llevar a cabo la inicialización de la pantalla OLED, la lectura y el envío del valor capturado por el sensor, y la activación del modo de suspensión en el dispositivo hasta su próxima ejecución. (Bertoletti, P. 2019).

Nodo receptor con envío de datos a internet

Incluir las librerías necesarias para que el IDE compile y ejecute el código correctamente. (Bertoletti, P. 2019).

```
// Librería Wi-Fi  
#include <WiFi.h>  
  
// Librería para sincronizar la hora  
#include "time.h"  
  
// Librerías LoRa  
#include <SPI.h>  
#include <LoRa.h>  
  
// Librería para Display OLED  
#include <Wire.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>
```

Definición de valores para los pines de la placa de desarrollo. (Bertoletti, P. 2019).

```
//Pines según placa de desarrollo LoRa
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 23
#define DIO 26

#define BAND 915E6 // Definición de frecuencia LoRa

// Pines display OLED según placa de desarrollo
#define OLED_SDA 21
#define OLED_SCL 22
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

//Definición de variables
String mV = "";
String mA = "";
String Temperatura = "";
```

Inicialización de variables que se utilizarán para conectar a la red WiFi y obtener la hora desde internet.

```
// Configurar conexion WiFi
const char* ssid      = "WiFi_SSDDi";
const char* password = "*****";

//Obtener la hora
const char* ntpServer = "time1.google.com";
const long  gmtoffset_sec = -10800;
```

Inicialización de variables que se utilizarán para cargar los datos que se reciban por la red LoRa.

```
// Inicializando variables para almacenar datos LoRa
int rssi;
//String readingID;
String LoRaData;
```

Inicialización del display de la placa de desarrollo, los cuales mostraran información sobre la red WiFi y los paquetes que se reciban por la red LoRa.

```

void startOLED () {
  pinMode(OLED_RST, OUTPUT);
  digitalWrite(OLED_RST, LOW);
  delay(20);
  digitalWrite(OLED_RST, HIGH);

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

```

El siguiente código inicializa el módulo LoRa de la placa de desarrollo e informa su estado por el display.

```

//Inicializando Modulo LoRa
void startLoRa(){
  int counter;
  //SPI LoRa pins
  SPI.begin(SCK, MISO, MOSI, SS);
  //setup LoRa transceiver module
  LoRa.setPins(SS, RST, DIO);

  while (!LoRa.begin(BAND) && counter < 10) {
    Serial.print(".");
    counter++;
    delay(500);
  }
  Serial.println("LoRa Inicializado OK!");
  display.setCursor(0,10);
  display.clearDisplay();
  display.print("LoRa Inicializado OK!");
  display.display();
  delay(2000);
}

```

La siguiente función utiliza las variables anteriormente declaradas para conectar la placa de desarrollo a la red WiFi, mostrando en el display el estado de la conexión y la dirección IP obtenida.

```

void connectWiFi(){
  // Conectar a la red WiFi con la contraseña almacenada
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Mostrar dirección IP
  Serial.println("");
  Serial.println("WiFi conectado.");
  Serial.print("Direccion IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  display.display();
}

```

La siguiente función espera la recepción de paquetes por la red LoRa, y carga cada una de las variables separadas por el símbolo '\$'. Luego mostrará por pantalla la recepción del paquete y el valor RSSI.

```

// Recibir los datos del sensor por la red LoRa
void getLoRaData() {
  printLocalTime();
  Serial.print(" Paquete LoRa Recibido: Temp: ");
  // Read packet
  while (LoRa.available()) {

    mV = LoRa.readStringUntil('$');
    mA = LoRa.readStringUntil('$');
    Temperatura = LoRa.readStringUntil('$');
    //LoRaData = LoRa.readString();
    Serial.print(LoRaData) }

  // Mostrar RSSI
  rssi = LoRa.packetRssi();
  Serial.print(" with RSSI ");
  Serial.println(rssi);

  display.clearDisplay();
  display.setCursor(0,30);
  display.print("Paquetes Recibidos ");
  display.setCursor(0,50);
  display.print("RSSI: ");
  display.setCursor(30,50);
  display.print(rssi);
  display.print("dB");
  display.display();
}

```

Ejecución de las funciones de los módulos display, LoRa y WiFi. Por último obtiene la hora desde internet.

```
void setup() {
  Serial.begin(115200);
  startOLED();
  startLoRA();
  connectWiFi();

  // Obtener la hora desde internet
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
}
```

En la función dedicada a ejecutarse en modo 'loop' el código quedará a la espera de recepción de paquetes por la red LoRa.

```
void loop() {
  // Chequea la existencia de paquetes por la red
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    getLoRaData();
  }
}
```

La siguiente función muestra la hora local obtenida por internet.

```
void printLocalTime()
{
  struct tm timeinfo;
  if(!getLocalTime(&timeinfo)){Serial.println("Failed
to obtain time");return;}
  Serial.print(&timeinfo, "%Y/%m/%d %H:%M:%S");
}
```

Empleando los datos adquiridos mediante la red LoRa y la hora obtenida desde internet, se procederá a elaborar una solicitud HTTP. Dicha solicitud enviará los datos hacia una hoja de cálculo alojada en Google Drive, utilizando el servicio IFTTT como puente intermedio. Finalmente, se procederá al cierre de todas las conexiones previamente establecidas.

```
// Crear HTTP request para el servicio IFTTT
void ConnectIFFT() {
    Serial.println();
    Serial.print("Connecting to ");
    Serial.print(server);

    WiFiClient client;
    int retries = 5;
    while(!!!client.connect(server, 80) && (retries-- > 0)) {
        Serial.print("."); }

    String jsonObject = String("{\"value1\":\"" + Temperatura +
    "\",\"value2\":\"" + rssi + "\",\"value3\":\"" + mV + "\"}");

    client.println(String("POST ") + resource + " HTTP/1.1");
    client.println(String("Host: ") + server);
    client.println("Connection: close\r\nContent-Type: application/json");
    client.print("Content-Length: ");
    client.println(jsonObject.length());
    client.println();
    client.println(jsonObject);

    int timeout = 5 * 10; // 5 seconds
    while(!!!client.available() && (timeout-- > 0)){
        delay(100);
    }
    if(!!!client.available()) {
        Serial.println("No response...");
    }
    while(client.available()){
        Serial.write(client.read());
    }

    Serial.println("\nclosing connection");
    Serial.println();
    client.stop(); }
```

Acrónimos

3GPP	3rd Generation Partnership Project.
ABP	Activation by Personalization.
BPSK	Binary Phase Shift Keying.
CSS	Compressed Sensing in Spectroscopy.
IDE	Integrated Development Environment.
IoT	Internet of Things.
LORA	Long Range.
LORAWAN	Long Range Wide Area Network.
LPWAN	Low Power Wide Area Network.
LTE	Long Term Evolution.
M2M	Machine to Machine.
OTAA	Over the Air Activation.
PoC	Proof of Concept.
QPSK	Quadrature Phase-Shift Keying.
RSSI	Received Signal Strength Indicator.
SoC	System on Chip.
SNR	Signal to Noise Ratio.
SPI	Serial Peripheral Interface.

Referencias

- Alldatasheet.com. (2018). BME280 Datasheet - Bosch Sensortec GmbH.
<https://www.alldatasheet.com/datasheet-pdf/pdf/1132060/BOSCH/BME280.html>
- Appert, J. (1993). *El almacenamiento de granos y semillas alimenticios*. Ed. hemisferio sur.
https://www.lea-global.com/uploads/circulares/2015/09/1_combustion_espontanea_de_granos.pdf
- Bertoletti, P. (2019). *Proyectos con ESP32 Y Lora*. Editora NCB.
- Centenaro, M., & Vangelista, L. (2016, October 5). *Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios*. Digital Object Identifier System. <https://doi.org/10.1109/mwc.2016.7721743>
- Esp32. (2022). <https://www.espressif.com/en/products/socs/esp32>
- Fulton, A. (2019, March). *Lora devices for smart agriculture*. Analog and Mixed-Signal Semiconductors. <https://info.semtech.com/smart-agriculture-e-book>
- Gilson, R., Bajic, K., Chaxel, E., & Meyer, F. (2018, April). *LoRaWANTM CAPACITY TRIAL IN DENSE URBAN ENVIRONMENT*. Iot - LoRaWAN - Smart City Solutions. https://www.smart-city-solutions.de/wp-content/uploads/2018/04/machineQ_LoRaWan_Capacity_Trial.pdf
- Grion, F., Petracca, G., & Lipuma, O. (2017). *Lora network coverage evaluation in urban and densely urban enviroment simulation and validation tests in autonomous city of Buenos Aires*. <https://ieeexplore.ieee.org/document/8214345>
- Hernández, R. (2021). *Lora vs LoRaWAN - ¿Cuál es la diferencia?* - Tu fuente experta en IoT. <https://iotconsulting.tech/lora-vs-lorawan/#:~:text=Qu%C3%A9%20es%20LoRaWAN,Area%20Amplia%20de%20Baja%20Potencia%20>

- Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2018, January). *A comparative study of LPWAN technologies for large-scale IoT deployment*. ScienceDirect.com | Science, health and medical journals, full text articles and books. <https://www.sciencedirect.com/science/article/pii/S2405959517302953>
- López, E. (s.f.). *INGENIERÍA EN MICROCONTROLADORES Protocolo SPI*. <https://www.i-micro.com/pdf/articulos/spi.pdf>
- LoRaWAN® specification v1.0.4. (2020, November 17). LoRa Alliance®. <https://resources.lora-alliance.org/technical-specifications/ts001-1-0-4-lorawan-l2-1-0-4-specification>
- LoRaWAN architecture. (2022, April). The Things Network. <https://www.thethingsnetwork.org/docs/lorawan/architecture/>
- Pégulu, M. (2022, April 6). *Environmental conditions and oyster health monitoring with Lora*. Semtech's Corporate Blog. <https://blog.semtech.com/monitoring-environmental-conditions-and-oyster-health-with-lora>
- Seedstudio.com. (2022, April 2). *Lora fiberglass antenna, 860-930MHz, 5dBi, 600mm*. Seed Studio Bazaar, The IoT Hardware enabler. <https://www.seedstudio.com/Lora-Fiberglass-Antenna-860-930MHz-5dBi-600mm-p-4927.html>
- Sensor de Corriente ina219 3v-5v i2c. (2022, August). UNIT Electronics. <https://uelectronics.com/producto/sensor-corriente-ina219-3v-5v-i2c/>
- Shenzhen Xin Yuan electronic technology Co., Ltd. (2018). Shenzhen Xin Yuan Electronic Technology Co., Ltd. <https://www.lilygo.cc/products/loral3>
- Texas Instrument. (2015, December). *INA219 Zero-Drift*. <https://www.ti.com/lit/ds/symlink/ina219.pdf?ts=1590377928988>