ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Universidad Politécnica de Cartagena

Trabajo Fin de Máster

MÁSTER EN INGENIERÍA TELEMÁTICA

Diseño y desarrollo de un sistema IoT basado en comunicaciones LoRa punto a punto

AUTORA: Magdalena Martos Núñez

DIRECTOR: Alejandro S. Martínez Sala

Septiembre/2021

Autor	Magdalena Martos Núñez
E-mail del autor	Magdalenamartos1313@gmail.com
Director	Alejando S. Martínez Sala
E-mail del director	alejandros.martinez@upct.es
Título del TFM	Diseño y desarrollo de un sistema IoT basado en comunicaciones LoRa punto a punto
Resumen	El proyecto llevado a cabo se basa en implementar una red LoRa mediante una comunicación punto a punto, que conste de un receptor y un número pequeño de emisores. Para llevar a cabo la implementación se van a realizar una serie de pruebas de cobertura, comprobando que la señal es capaz de transmitirse. Estas evaluaciones serán el primer paso para finalmente implementar una solución IoT para empresas de ganadería. El sistema para implementar sirve para monitorizar el ambiente en el que se encuentra el alimento que ingieren los animales, esto se lleva a cabo conectando sensores que capturen los datos en recintos donde el alimento esté encerrado. Con esta monitorización se puede realizar una trazabilidad del alimento, pudiendo conocer su estado y evitar posibles sobrecostes.
Titulación	Máster en Ingeniería Telemática
Departamento	Tecnología de la Información y las Comunicaciones.
Fecha de presentación	Septiembre-2021

Agradecimientos

A mi familia, por confiar en mí y ser mi motor principal.

A mi pareja, porque me ha mantenido en calma y su gran apoyo.

A mis amigos, por hacerme desconectar fácilmente.

A Alejandro, por enseñarme a organizarme y a no rendirme ante cualquier adversidad.

A Neoradix Solutions, por abrirme la puerta de su empresa y dejarme dar mis primeros pasos en sector.

Contenido

Índice ilustraciones

Índice de tablas

Capítulo 1. Introducción y objetivos

1.1 Introducción

Desde hace unos años estamos viviendo una cuarta revolución industrial, que ha dado lugar al término Industria 4.0; esta no es más que la introducción de nuevas tecnologías en el ámbito laboral tanto a nivel de manufacturación como a nivel industrial. Este proyecto está enfocado en el uso de IoT (Internet of Things) siendo una solución emergente que integra múltiples tecnologías. El fin es proveer a las "cosas" (objetos cotidianos, maquinaria industrial, animales…) de Internet, mejorando así la calidad de vida de los seres vivos y cualquier objeto tangible. Esta propuesta realizada en colaboración con la empresa Neoradix Solutions, consiste en el uso de esta tecnología para realizar mediciones en el ambiente del pienso, para comprobar la calidad del grano. Con el fin de evitar que los seres vivos consuman pienso en mal estado o para detectar cualquier anomalía en el recinto que altere el ambiente y haga perder kilos de pienso.

Se va a realizar la implementación de una red LPWAN (Low Power Wide Area Network) que consiste en un tipo de red que ofrece una gran cobertura con un consumo mínimo de potencia. Entre este tipo de redes se encuentra LoRa (Low Range), esta tecnología tiene la finalidad de poder conectar múltiples dispositivos siendo capaz cubrir distancias de hasta 12 km. Los dispositivos que se conectan con LoRa para emitir son los módulos transmisores, que se encargarán de realizar lecturas periódicas de los sensores (temperatura y humedad) y serán transmitidas a través de la red LoRa hacia un receptor que recibirá los datos. El receptor es el otro módulo LoRa que reenviará los mensajes hacia un gateway mediante un puerto serie. El gateway además de almacenar los datos, será el encargado de establecer conexión con Internet a través de un router IoT; para así enviar los datos recogidos al backend (servidor de Neoradix). Por lo tanto, cada vez que se envía un nuevo dato por la red LoRa, el gateway debe desempeñar múltiples tareas como podría ser una petición al servidor para almacenar cada nuevo valor o generar logs para poder depurar la información recibida. Esta arquitectura se puede visualizar en la Ilustración 1.

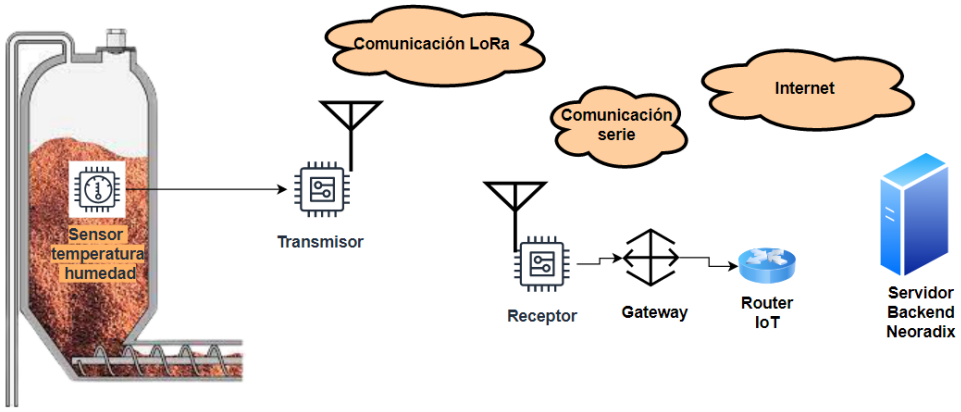

Ilustración 1. Esquema general

1.2 Herramientas software y elementos usados

El proyecto se ha llevado a cabo utilizando distintas tecnologías en una misma arquitectura con el fin de realizar un sistema de recogida de datos completo. De forma que ha sido necesario el conocimiento y manejo de diferentes herramientas para poder realizar cada tarea de forma fructífera.

La implementación de los módulos de comunicación LoRa se va a realizar sobre las placas de desarrollo Lopy 4, las cuales cuentan con el microprocesador ESP32. La programación de estos módulos se ha llevado a cabo con MicroPython, un lenguaje que nos permite implementar soluciones sobre microcontroladores. MicroPython incluye librerías básicas análogas a las usadas en Python como son machine, time, network, entre otras. Cada librería cuenta con una funcionalidad determinada.

Además de las librerías ya existentes, es necesaria una librería específica para el sensor que vamos a utilizar. Este sensor debe ser capaz de realizar lecturas de temperatura y humedad, se va a usar el sensor AM2315 y el cual establece comunicación con el módulo Lopy 4 a través del protocolo I2C (Inter-Integrated Circuit). La librería usada es *AM2320* en la que se implementa el código necesario para realizar las lecturas y otras funciones sobre el sensor.

El código que se va a procesar dentro de los módulos es necesario programarlo previamente y depurarlo, para ello se usa Visual Studio Code con el plugin de Pymakr. Pymakr es una herramienta muy potente que permite desde conectarse a un módulo, subir un proyecto, descargarlo o incluso obtener una lista de los puertos serie abiertos en nuestro ordenador.

Para habilitar el gateway se utiliza una Raspberry Pi, donde se implementa la solución con un script en Python. La programación de este script sobre la Raspberry PI se realiza con el editor de texto Nano, que está integrado en el sistema operativo de la Raspberry. La finalidad del script es leer del puerto serie y enviar los datos al servidor, además de generar logs para detectar cualquier error. Con el fin de poder modificar o consultar información de forma remota se realiza una conexión mediante ssh, aunque también pueden usarse otras herramientas que permitan el acceso remoto. Es necesario configurar mecanismos de seguridad para que el sistema permanezca en ejecución ante cualquier caída de tensión y se pueda depurar, para ello se va a

utilizar el daemon PM2. Este no es más que un proceso que se puede ejecutar sobre la Raspberry para ayudarnos a manejar una aplicación y poder recoger logs de forma automática.

1.3 Estructura y organización del proyecto

El proyecto consta del despliegue de una red LoRa punto a punto estableciendo comunicación a través de los módulos Lopy 4. De forma adicional se configura una Raspberry Pi que permitirá realizar el almacenaje de los datos y conectarse a Internet.

Para asegurar que la solución final funciona correctamente es necesario separarla en distintos hitos, con el fin de verificar que cada uno funciona como se espera de forma independiente. En el capítulo 2 se van a explicar las tecnologías empleadas, que abarcan desde los diferentes módulos hardware que se programan hasta cómo es necesario configurarlos para el funcionamiento del sistema. El capítulo 3 está enfocado a la compresión y evaluación de la capa física a partir de los parámetros que se explotan con LoRa. Una vez comprendida la capa física, se despliega el sistema en su totalidad con todos los componentes. El desglose del prototipo final está expuesto en el capítulo 4. Para finalizar, en el capítulo 5 se analizan las pruebas realizadas tanto para el prototipo de evaluación de la capa física como para el prototipo final. Por último, en el capítulo 6 se exponen las conclusiones y posibles líneas futuras.

Capítulo 2. Tecnologías Empleadas

Internet de las cosas (IoT) se entiende como un conjunto de tecnologías punteras que proveen de soluciones y servicios. Con este nuevo "boom" se espera cambiar la manera en la que vivimos y aportar mayor valor a la producción. Este nuevo paradigma provee la capacidad de conectar múltiples dispositivos que recogen datos de distinta naturaleza y son entregados a clientes/servicios en forma de gráficas o alarmas.

El desarrollo del proyecto se basa en el uso de la tecnología LoRa para la comunicación entre módulos Lopy 4. Para salir a la red de Internet se va a usar un gateway (alojado en una Raspberry Pi) que nos permita enviar los datos generados al servidor, esta comunicación con el servidor se realiza a través del protocolo HTTP.

2.1 Conceptos básicos de LPWAN

El tipo de red que se implementa en soluciones IoT se conoce como low power wide área network (LPWAN), la cual está actualmente en auge. Las redes LPWAN cuenta con unas características muy interesantes como son el bajo consumo de potencia, bajo coste y permiten una cobertura a grandes áreas que pueden alcanzar hasta 20 km de radio. Entre este tipo de tecnologías destacan Sigfox, NB-IoT y LoRa.

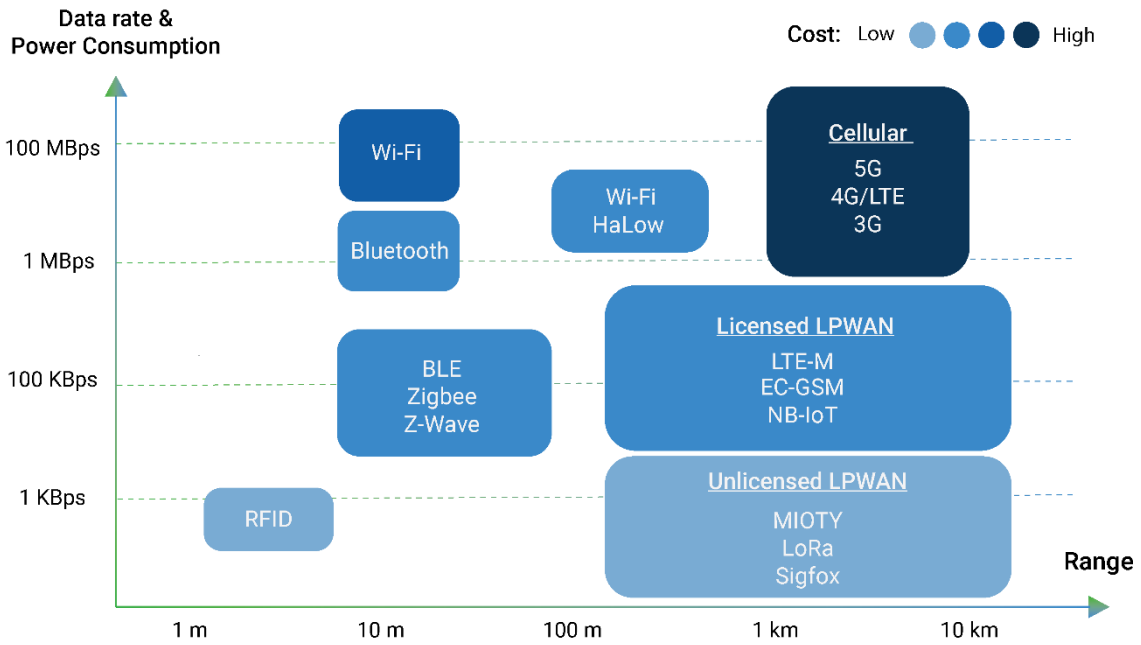

Ilustración 2.Comparativas tecnologías LPWAN

2.2 Conceptos básicos de LoRA

Antes de hablar sobre qué es LoRa, es necesario conocer de dónde surgió esta tecnología. Como ya se ha expuesto anteriormente IoT amplia y mejora el ámbito de las comunicaciones inalámbricas.

LoRa (Low Range) proviene de largo alcance, fue desarrollada por LoRa Alliance que es una asociación tecnológica encargada de definir la tecnología y sus estándares de uso. Para poder implementar soluciones de largo alcance, LoRa usa un esquema de modulación propietario por la empresa Semtech que trata la señal a nivel de capa física, correspondiente a nivel 1 de la capa OSI. Para modular las señales a nivel físico se utiliza las bandas ISM: 433 MHz, 868MHz o 915 MHz. La comunicación punto a punto se realiza a partir del tipo de modulación chrip spread spectrum (CSS). CSS como se puede visualizar en la ilustración 3 permite que la señal ocupe un mayor ancho de banda, proveyendo de mayor robustez en la comunicación y a factores de ruido como el efecto Doppler. Este tipo de modulación se suele usar para fines militares o comunicaciones muy robustas. Esta técnica permite que la señal sea capaz de recibir de múltiples señales al mismo tiempo. Un dispositivo LoRa será capaz de recibir y traducir información de señales en radiofrecuencia (RF).

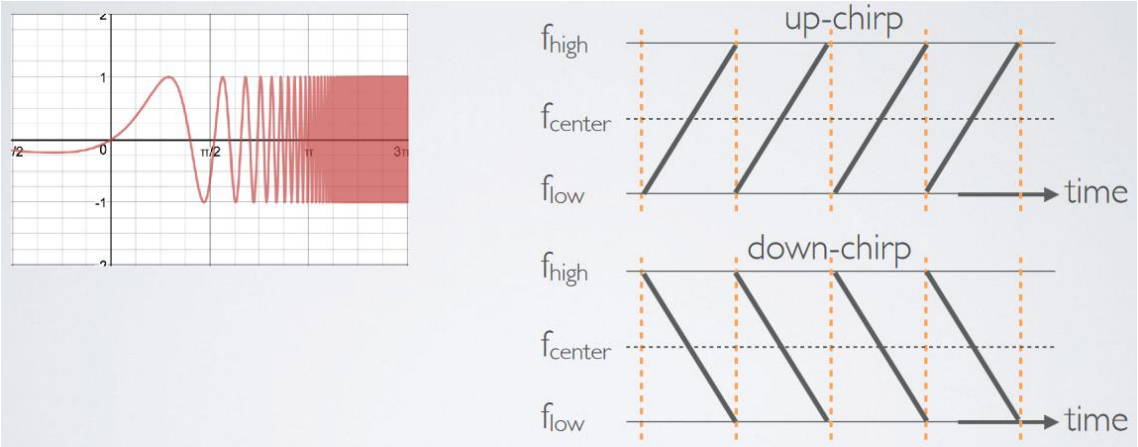

Ilustración 3. Modulación CSS

Los parámetros a analizar para poder comunicar dispositivos a través de la modulación LoRa, son los siguientes:

- Frecuencia portadora, que es la banda de frecuencia en la que se produce la transmisión.
- Ancho de banda es el espacio del espectro en el que se puede transmitir la señal.
- Tasa de codificación que permite definir la tasa de corrección de errores.
- Factor de ensanchado, representa el parámetro chrip ensanchado (chirps/s).

- Potencia transmitida es la energía irradiada por la antena de un nodo LoRa.

Dentro de LoRa tenemos varios dispositivos imprescindibles como son los nodos o también conocidos como dispositivos finales, donde se realizan las lecturas del sensor. Por otro lado, tenemos el gateway que permite integrar los dispositivos LoRa con otras tecnologías como Internet, para que finalmente llegue al servidor donde se van a procesar las peticiones HTTP.

Es imprescindible que exista cobertura entre transmisor y receptor para que se puedan comunicar entre sí, así que se deberán tener en cuenta el valor de las pérdidas en el espacio libre, la zona Fresnel y el RSSI (Received Signal Strength Indicator); que nos permite saber cómo se va atenuando la señal para cada nuevo dato emitido. Existen otros parámetros que nos ayudan a calcular la calidad de la señal. A la hora de elegir los valores a configurar es necesario tener en cuenta la Tabla 1.

DATA RATE	SPREADING FACTOR	BITRATE (BITS/S)	RANGO (KM)	SENSIBILIDAD RX (DBM)	PAYLOAD (BYTES)
0	12	290	12+	-136	51
1	11	440	10	-133	51
2	10	980	8	-132	51
3	9	1760	6	-129	115
4	8	3125	4	-126	222
5	7	5470	2	-123	22

Tabla 1. Valores LoRa

2.3 Módulo Lopy 4

Lopy 4 es una placa de desarrollo que se programa en MicroPython y permite comunicación LoRa, Sigfox, Bluetooth y WiFi. Por lo tanto, es muy fácil de integrar en soluciones IoT. Estos módulos llevan integrado un microcontrolador ESP32 que permite comunicarse con el resto de las componentes de la placa y configurar el pinout. A la hora de desarrollar nuestra solución se han tenido en cuenta ciertas funcionalidades que ofrecen estos microcontroladores como son:

- La gestión del consumo de batería
- Comunicación con sensores I2C
- Envío de datos por la UART
- Envío y recepción de datos a través de las antenas de RF junto con el chip SX1276 de Semtech. Este chip provee de cobertura LoRa y Sigfox y su configuración queda detallada en la tabla 2.

Chip	Potencia Tx (dB)	Sensibilidad Rx (dB)	Freq (Mhz)	BW (KHz)	Bit Rate (Kbps)	SF
SX1276	13	-148	868	125	37.5	12

Tabla 2.Configuración Chip SX1276

Estos módulos serán utilizados para implementar tanto los transmisores como el receptor. En el caso del transmisor, será necesario tener en cuenta la vida útil de la batería y el conexionado de los cables para el sensor. El sensor se encarga de realizar las lecturas de la temperatura y humedad en el ambiente.

El consumo de los módulos transmisores es una de las claves del éxito de este tipo de módulos de IoT, por lo tanto, se puede estimar el consumo para nuestro caso en particular. En la tabla 3 se muestra el consumo para cada modo en el que se encuentra el Lopy 4 y conociendo el tiempo en cada modo se puede hacer una estimación de la duración de las baterías. Evitando así dejar el dispositivo sin batería y poder prever los costes de la solución.

MODO	CONSUMO	TIEMPO	CONSUMO DIARIO
DEEP SLEEP	18.5 uA	30 min	1.296 mAh
TRANSMITIENDO LORA	108 mA	10 sec	7.13 mAh
LEYENDO SENSOR	35.4 mA	5 sec	1.16 mAh

Tabla 3. Consumo Lopy 4

El módulo receptor será el encargado de recibir los datos correctamente y así realizar la transmisión por la UART a través del puerto serie. Para poder procesar el valor de un dato y saber cómo está funcionando la señal LoRa podemos consultar una serie de estadísticas, que nos provee la librería LoRa donde podemos obtener el valor del RSSI. Para las comunicaciones inalámbricas se considera que una señal es audible cuando oscila entre -30 dBm y como mínimo se acepta una señal en torno a -120 dBm.

2.4 Sensor AM2315

La finalidad del Lopy 4 transmisor, es tomar muestras de la temperatura y humedad ambientales, para ello se conectará un sensor. Las lecturas se realizan a través de un sensor, el AM2315, el cual nos va a permitir realizar lecturas de la temperatura y humedad en cada momento. Este sensor nos permite generar una salida digital de los datos con una gran precisión. La comunicación entre el transmisor y el sensor AM2315 es mediante el protocolo I2C, haciendo uso de la señal de reloj y de una señal adicional para permitir la carga de datos. Los pines del sensor y su conexión con el Lopy 4 se muestran en la tabla 4 y en la ilustración 4 se muestra el diagrama de conexiones. Para que las lecturas por parte del transmisor sean realizadas con éxito, es necesario una librería que lea los datos recibidos a través del protocolo I2C para ello se va a usar la librería llamada “AM2320” con la que se pueden realizar las lecturas.

Ilustración 4. Sensor AM2315

PIN	COLOR	NOMBRE	DESCRIPCIÓN	CONEXIÓN LOPY 4
1	Rojo	VDD	Voltaje (3.3V- 5V)	3.3 V
2	Amarillo	SDA	Datos serie (bidireccional)	P10
3	Negro	GND	Tierra	P9
4	Blanco	SCL	Reloj serie	GND

Tabla 4. Pinout AM2315 a Lopy 4

2.5 Raspberry Pi

Este módulo se conoce como el gateway entre la red LoRa e Internet. Una Raspberry Pi es un pequeño computador en el que se puede realizar cualquier tarea de forma análoga a un PC tradicional. Este desempeña la función de capturar datos a través del puerto serie del Lopy 4 receptor, estos datos serán sometidos a un preprocesado; descartando algunos de ellos en caso de que sea necesario. Por último, estos datos serán enviados a través de una petición HTTP al servidor de la empresa Neoradix Solutions. Para realizar el envío mediante HTTP debemos tener una salida a Internet, de este modo la Raspberry estará conectada a un router con conectividad a Internet.

Ilustración 5. Raspberry PI 3

2.6 Router Teltonika 240

El router 240 de Teltonika es un router industrial que permite conectividad 4G, 3G y 2G, este se conecta a través de una tarjeta SIM. Una de las características de este router es que nos permite crear redes WiFi, además cuenta con una gran capacidad para conectarse con aplicaciones IoT. Otras funciones que puede realizar es la creación de una VPN y también permite una conexión remota a los dispositivos conectados a su red, gracias a su sistema RMS (Remote Management System). Este es accesible a través de una URL que apunta a la configuración del router y es ofrecida por la empresa Teltonika, además cuenta con un panel donde se pueden realizar conexiones SSH con los dispositivos de la red.

Ilustración 6. Router Teltonika RUT240

Capítulo 3. Diseño e implementación de capa física

En este capítulo se van a analizar las pruebas realizadas para comprobar el funcionamiento del primer hito del sistema, es decir analizar que la comunicación entre Lopys es la esperada.

Nuestro interés de analizar la capa física viene de los principios de la tecnología LoRa que no es más que comunicar dispositivos a largas distancias. Es necesario realizar estas pruebas para así comprobar que los parámetros de la capa física seleccionados son los óptimos. La cobertura de una antena LoRa va a depender de diversos factores expuestos en el *Capítulo 2* teniendo mayor peso el valor del factor de ensanchamiento (SF).

3.1 Configuración de la capa física

Para poder conectar dos dispositivos a través de LoRa, estos deben tener los módulos LoRa configurados con los parámetros adecuados para poder realizar una comunicación punto a punto. Estos parámetros se comprueban realizando una serie de pruebas para validar la capa física. En nuestro caso el sistema para evaluar las prestaciones de la comunicación punto a punto está formado por un transmisor y un receptor. Ambos dispositivos tienen los mismos parámetros configurados para poder comunicarse y realizar el envío y recepción de los datos, como se puede ver en la Ilustración 7.

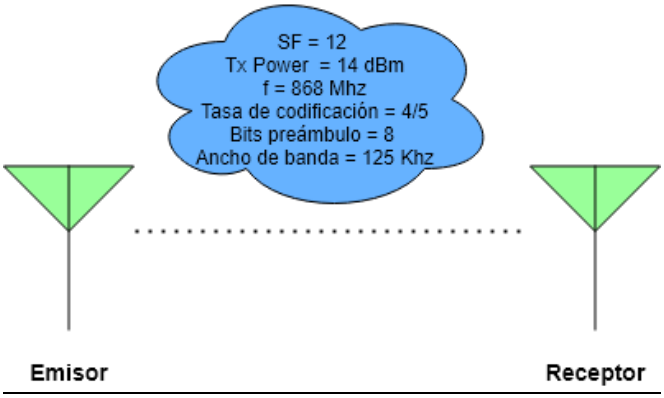

Ilustración 7. Esquema comunicación punto a punto

Los parámetros seleccionados en nuestro caso son aquellos que son capaces de satisfacer la cobertura en torno a 100-200 metro. Algunos valores quedan definidos por la legislación como la frecuencia y la potencia máxima transmitida, los demás pueden variar dependiendo de los requisitos que tenga el sistema a implementar.

	Transmisor/Receptor
Spreading Factor (SF)	12
Potencia Transmitida (P_tx)	14 dBm
Tasa de codificación	4/5
Frecuencia (f)	868 MHz
Bits de preámbulo	8
Ancho de banda	125 KHz

Tabla 5. Parámetros capa física LoRa

En la tabla 5 se muestra de forma detallada cuales son los valores que se han tomado para la validación de la capa física.

El transmisor implementado sobre un módulo Lopy va a realizar una serie de pasos para así poder comprobar la capa física. El primer paso será crear un mensaje longitud fija de 20 bytes que contendrá simplemente un número de secuencia y el resto del mensaje inalterado entre distintos envíos. La estructura de los mensajes transmitidos es: Nº secuencia + Payload. Una vez se crea el mensaje se realiza la conexión con LoRa a través de la antena del Lopy y se envía el mensaje. Como el payload es invariable cada nuevo mensaje incrementará el número de secuencia y se enviará el siguiente mensaje, así de forma periódica.

El flujograma que sigue el módulo Lopy es el mostrado en la Ilustración 8, el primer paso es crear el mensaje con payload invariable. Seguido se conectará la antena a la red LoRa y se enviará el mensaje a través de está. Por otro lado, el receptor tiene como primer objetivo conectarse a la red para poder recibir mensajes LoRa en cualquier momento. Una vez que se ha recibido el mensaje, este es preprocesado y se enviará por la UART (Universal Asynchronous Receiver-Transmitter) del receptor como se ve en la Ilustración 9.

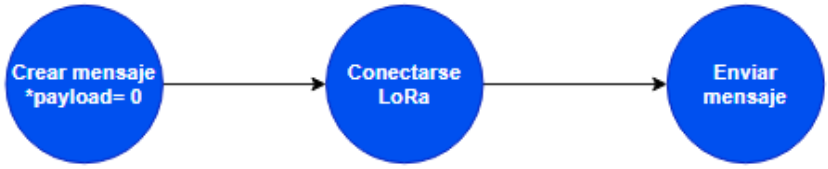

Ilustración 8. Flujograma Transmisor

Ilustración 9. Flujograma receptor

3.2 Recogida de muestras

Con el fin de realizar una recogida de muestras precisa y comprobar cómo está comportándose la comunicación LoRa entre el módulo Lopy transmisor y receptor. Se va a conectar el módulo receptor a un PC mediante un cable de datos, como se muestra en la Ilustración 10. De forma que se va a poder depurar y analizar la cobertura en la red LoRa. En nuestro caso se considera que la comunicación ha sido exitosa siempre y cuando el receptor sea capaz de recibir el dato que ha sido enviado por el transmisor. Esto se puede verificar con la comprobación del número de secuencia de cada mensaje.

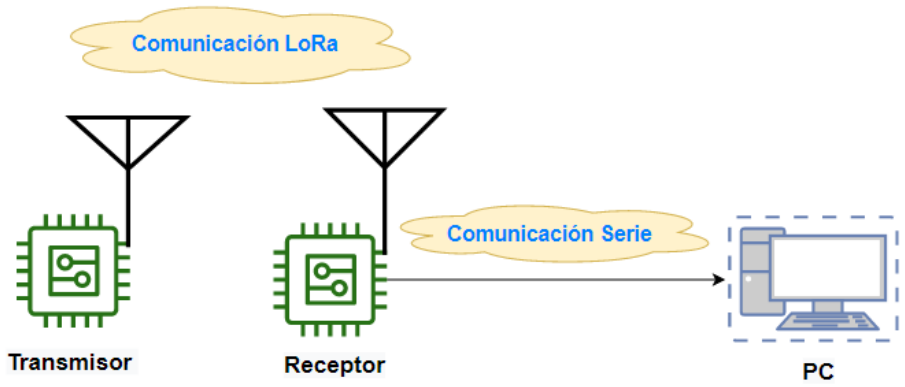

Ilustración 10. Arquitectura para validar capa física

Para realizar la funcionalidad de depuración en el PC se ejecutará un script “Check_coverage.py” que consta de las siguientes funciones: abrir el puerto serie del ordenador para que sea capaz de capturar los datos que llegan de la comunicación LoRa, interpretarlos y almacenarlos.

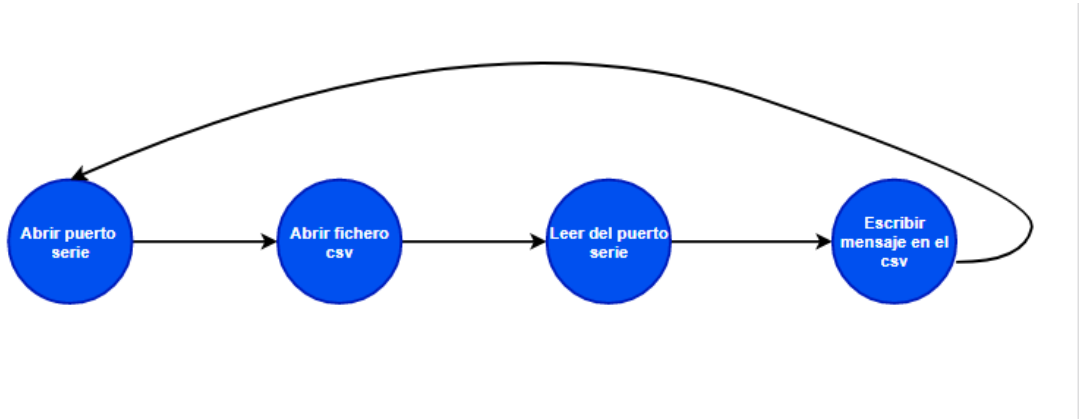

Ilustración 11. Flujograma Check_coverage

3.3 Análisis de la capa física

Para evaluar LoRa se ha decidido diseñar los mensajes con la estructura explicada previamente, es decir que cada uno contenga un número de secuencia. Siendo este primordial para analizar la fiabilidad del enlace entre transmisor y receptor. La tasa de éxito del enlace se calculará teniendo en cuenta la ratio de mensajes que llegan con éxito al receptor entre el número total de mensajes. La fórmula que se muestra a continuación nos ayuda a definir la tasa de éxito en la comunicación por LoRa.

$$Tasa\ éxito = 1 - \frac{Nº\ mensajes\ enviados - Nº\ mensajes\ recibidos}{Nº\ mensajes\ enviados}$$

3.4 Análisis de la duración de la batería

La batería en este tipo de sistemas es un elemento crucial, puesto que supone un gran cambio respecto a otras tecnologías y pretende ahorrar la máxima energía posible. Para ello, se van a realizar una serie de cálculos una vez que el sistema esté completo, de forma que se pueda evaluar el consumo de la batería por cada muestra recogida y entregada al servidor.

$$Consumo\ total\ diario = 1.296 + 1.16 + 7.13 = 9.586\ mA$$

Capítulo 4. Diseño e implementación de la arquitectura del sistema

En el desarrollo de esta aplicación tendremos que contemplar el proyecto como un sistema compuesto por cada componente hardware y con la implementación requerida para cada uno. Este capítulo deja atrás la exploración de las prestaciones de la capa física, basándonos a partir de ahora en evaluar la resiliencia a fallos del sistema. Se debe tener en cuenta todo el recorrido que hace un "dato" desde que parte de un sensor hasta que llega finalmente al servidor Neoradix.

4.1 Arquitectura del sistema

El sistema por implementar cuenta con varios puntos dentro de la red donde existen distintos protocolos y dispositivos.

La arquitectura del sistema quedará tal y como se ve en la Ilustración 12 compuesta por:

- Transmisores
- Receptor
- Raspberry Pi
- Servidor Neoradix

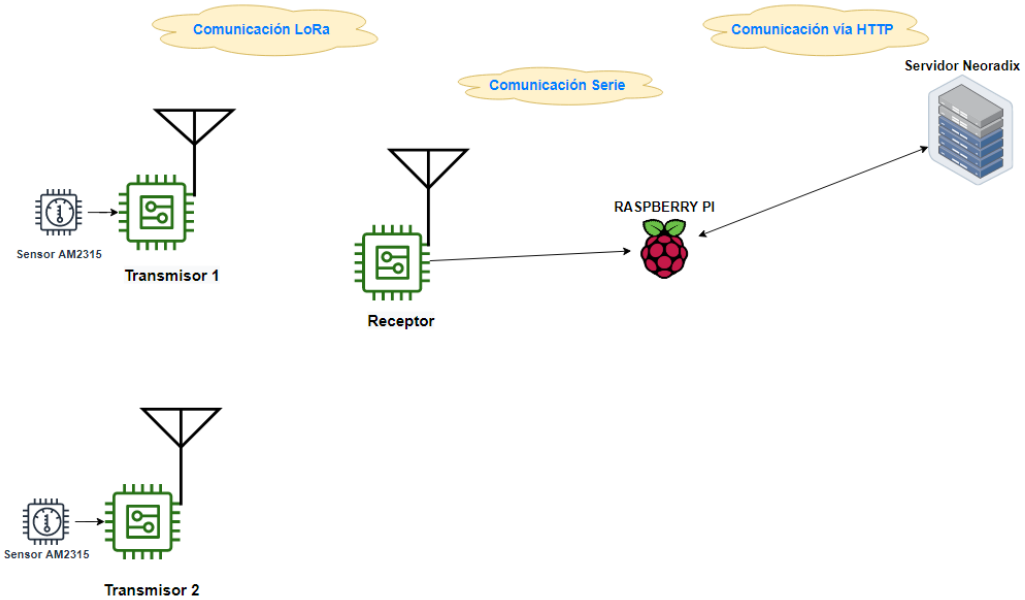

Ilustración 12. Arquitectura del sistema

Dentro de la arquitectura del sistema se dan múltiples intercambios de mensajes dónde se pueden distinguir entre los mensajes enviados por la red LoRa, por el puerto serie y peticiones

HTTP. Este conjunto de mensajes son los que conforman nuestra comunicación y se muestran de forma general en la Ilustración 13.

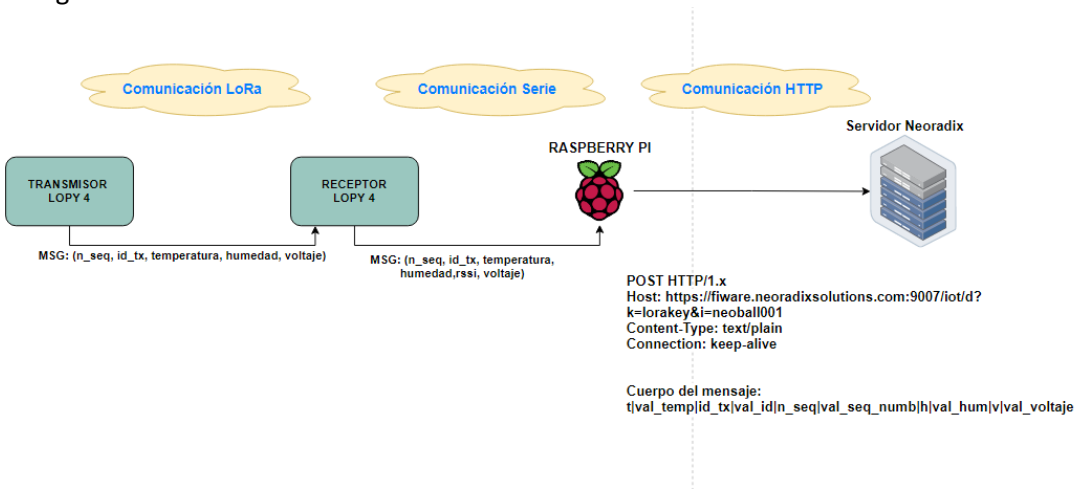

Ilustración 13. Intercambio de mensajes en la arquitectura

Los componentes se comunican mediante múltiples protocolos. La primera interacción es mediante LoRa, donde se ven involucrados los transmisores y el receptor, que es el punto de intercambio de mensajes LoRa. Por otro lado, tenemos la comunicación a través de la UART del receptor hacia el puerto serie de la Raspberry Pi, donde se intercambian mensajes UART a través de un cable de datos. El último tramo es una comunicación mediante el protocolo HTTP entre la Raspberry Pi y el servidor, aquí se almacenan y tratan los datos enviados por los transmisores.
El intercambio de mensajes entre los diferentes componentes de la arquitectura queda detallado de forma genérica en la Ilustración 14. Además, en la Ilustración 15 se visualizar un ejemplo de un mensaje con origen en el transmisor 1 y destino en el servidor de Neoradix, en el receptor se observa que se añade el valor de la potencia de la señal que recibe el receptor.

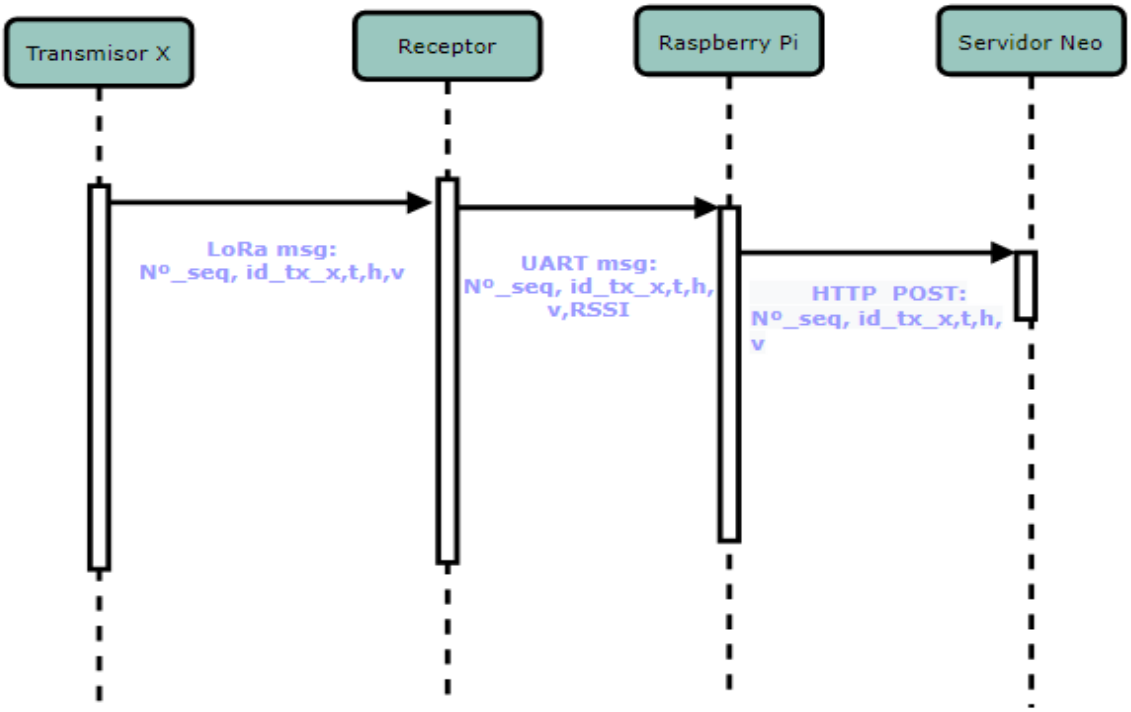

Ilustración 14. Intercambio mensajes

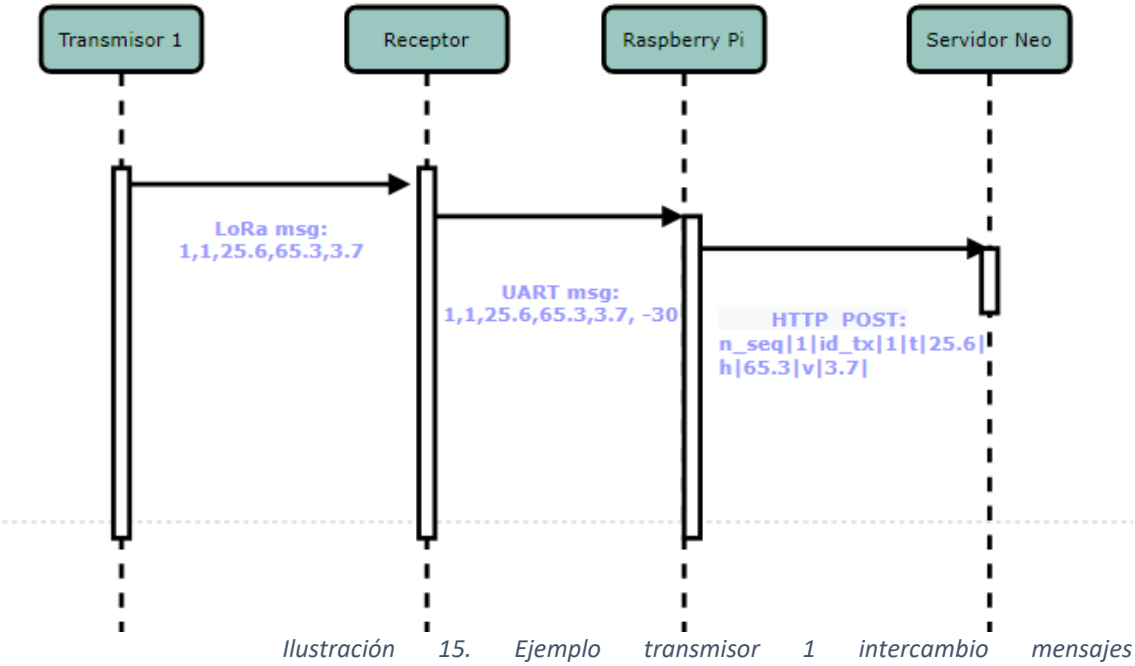

Ilustración 15. Ejemplo transmisor 1 intercambio mensajes

4.2 Funcionamiento Lopy transmisor

El transmisor está compuesto por un módulo Lopy 4, el cual estará alimentado por una batería tipo LiPo de 2000mA. La finalidad de este módulo es encargase de realizar la lectura del sensor AM2315, conectarse a la red LoRa, enviar los datos al receptor a través de la red y finalmente, permanecer en reposo con el fin de optimizar el consumo de batería. En la Ilustración 16 se muestra el flujograma que describe el comportamiento del transmisor.

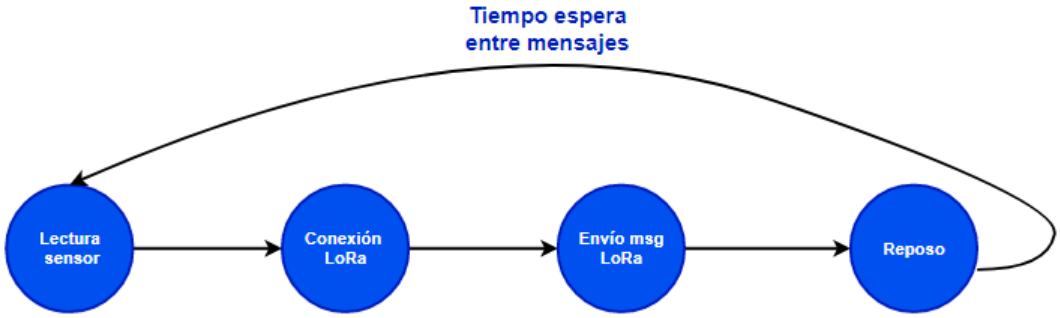

Ilustración 16. Flujograma código transmisor

Para realizar la conexión a través de LoRa es necesario configurar la capa física como se ha expuesto en el capítulo 3. El transmisor es capaz de realizar lecturas de la temperatura y humedad en el ambiente a través de un sensor. Además, se realiza una lectura de un pin del Lopy transmisor con el fin de realizar lecturas del valor de voltaje de la pila, permitiendo así estimar la duración de la misma. Una vez recogidos los datos, nos conectaremos a la red LoRa por donde se enviará el mensaje con la siguiente estructura:

Nº_seq + identificador_transmisor + temperatura + humedad + voltaje

Se van a implementar dos módulos transmisores. Para cargar el código en los módulos se conectan al PC las placas Lopy 4 y se actualizan con el código implementado, cumpliendo con los requisitos del flujograma. A la hora de configurar los transmisores o añadir nuevos, simplemente es necesario modificar el parámetro *identificador_transmisor* con el correspondiente número de transmisor.

4.3 Funcionamiento Lopy receptor

El módulo encargado de la recepción va a recibir los mensajes enviados desde los transmisores, para ello es imprescindible la comunicación entre ambos módulos a través de la red LoRa. Se pretende que estos datos que llegan al receptor se envíen a través de la UART del módulo receptor, como se muestra en el flujo de la Ilustración 17. El mensaje a enviar a través de la UART añade un nuevo parámetro al mensaje recibido, el RSSI cuya importancia se ha explicado en el Capítulo 2. Quedando el mensaje con la siguiente estructura:

Nº_seq + identificador_transmisor + Temperatura + Humedad + Voltaje + RSSI

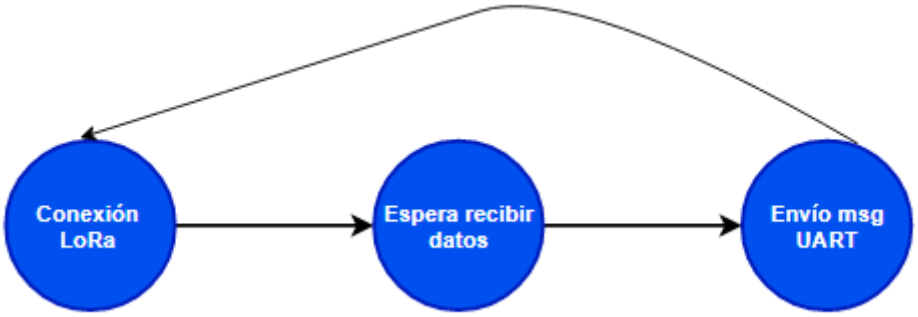

Ilustración 17. Flujograma código receptor

4.4 Funcionamiento Raspberry Pi

La Raspberry Pi va a actuar como gateway entre la red LoRa y la comunicación hacia el exterior. Siendo esta el elemento esencial del sistema y permite tener trazabilidad de los datos generados.

En primer lugar, la Raspberry se va a encargar de recibir los datos a través de uno de sus puertos USB, este puerto va a permitir la comunicación en serie con el módulo receptor mediante un cable de datos. Esta funcionalidad se implementa mediante un script de Python *serial_sniffer.py* de manera que los mensajes que recibe la Raspberry Pi por el puerto USB se almacenan en un archivo *.csv* y finalmente se enviarán al servidor de Neoradix. Los mensajes enviados al backend se implementan mediante una petición POST de HTTP que incluirá el formato deseado para que la información quede almacenada, este flujo del script está descrito en la Ilustración 18. Con el fin de proveer de mayor robustez al sistema se van a generar registros en un fichero de logs, donde podremos contemplar si todos los procesos se realizan con éxito o en su defecto si existe cualquier fallo.

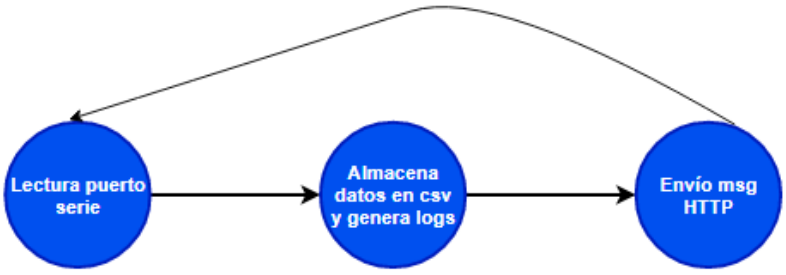

Ilustración 18. Flujograma serial_sniffer.py

Con el fin de aumentar la tolerancia a fallos de la Raspberry Pi se ha integrado PM2, que es un daemon que permite configurar y manejar una aplicación online. En el caso de la Raspberry Pi se va a utilizar PM2 para que la aplicación sea capaz de iniciarse sola ante cualquier caída o corte de tensión en la Raspberry Pi. Además, nos va a permitir monitorizar y diagnosticar todos los eventos que ocurren en nuestra aplicación pudiendo generar un fichero que centralice los logs.

Para hacer uso de PM2 en la Raspberry se requiere Node.js, para ello se va a descargar el archivo .tar.gz directamente de la web de Node.js. Una vez instalado Node, se va a hacer uso de npm que es su gestor de paquetes y ya se podrá instalar PM2. Para ello se lanzará el comando *npm install pm2 -g.* PM2 nos permite demonizar un script, además puede usarse para mantener aplicaciones. En nuestro caso se va a utilizar para arrancar y mantener el script s*erial_sniffer.py*. Con el fin de poder desplegar este servicio primero hay que ejecutar *pm2 start serial_sniffer.py,* donde nos muestra una salida como la de la Ilustración 19. En caso de realiza algún cambio en el script *serial_sniffer.py* es necesario que se almacenen cualquier modificación y para ello se debe ejecutar *pm2 save.* En caso de necesitar monitorizar o mostrar logs se pueden ejecutar los comandos *pm2 monit o pm2 logs,* respectivamente.

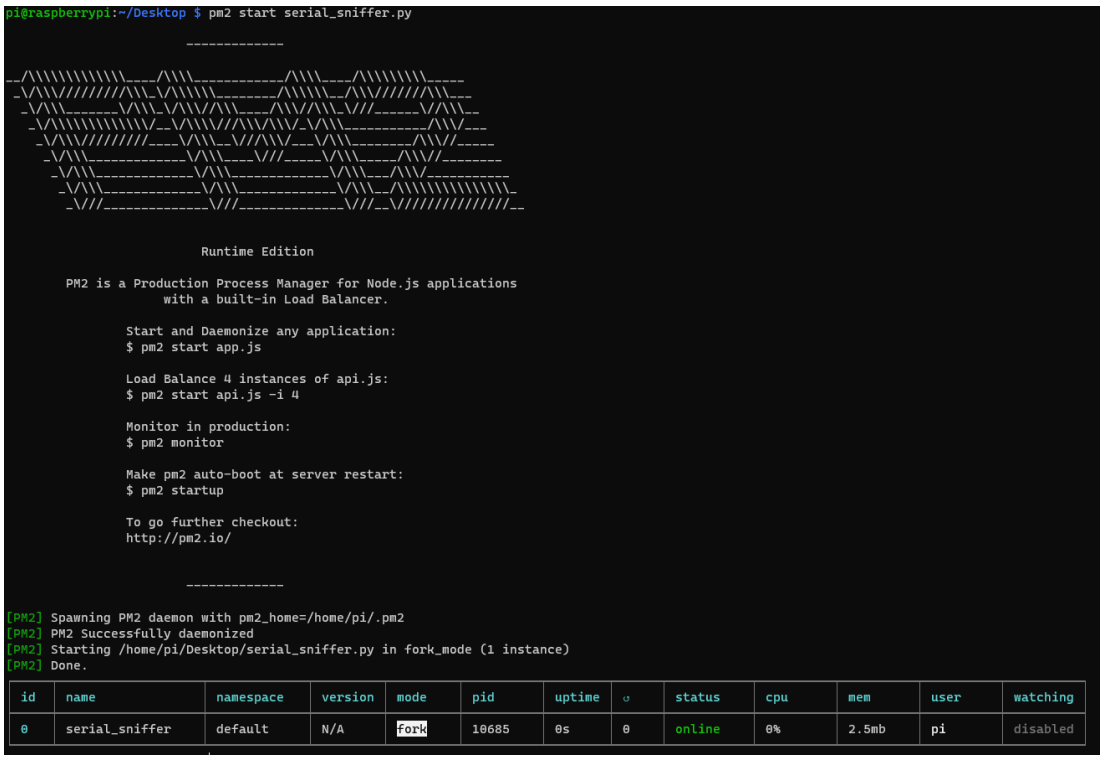

Ilustración 19. Inicio PM2

Este módulo que actúa a la vez de gateway y de servidor local, nos va a permitir almacenar la información transmitida a través de LoRa. El gateway es necesario que sea accesible de forma remota, de modo que en cualquier momento sea posible comprobar los mensajes enviados para validar la recepción o el porcentaje de batería de los transmisores. Para realizar la conexión en remoto será necesario habilitar el protocolo de red SSH. El comando que se ejecuta para realizar este tipo de conexiones es *ssh username@ip*. En la arquitectura no se contempla el acceso desde

una red externa mediante ssh, solo a través de la red local. Así que se va a hacer uso de la herramienta AnyDesk para poder acceder desde el exterior de la red al gateway y poder ver el estado de la arquitectura.

AnyDesk es un programa software que permite realizar conexiones remotas en dispositivos con escritorio gráfico, que se puede implementar con un servidor VNC. Para poder conectarse es necesario que la Raspberry tenga configurada la herramienta y se deben autorizar conexiones remotas sin necesidad de confirmación por la Raspberry Pi. Este programa también será instalado en los equipos para acceder de forma remota.

4.5 Router IoT

El último tramo por el que pasarán nuestros mensajes dentro de la arquitectura es el servidor, por lo tanto, es necesario utilizar un router que provea de salida a Internet. En nuestro caso usaremos el router 240 de Teltonika que estará conectado a la Raspberry Pi mediante un cable Ethernet de forma que los mensajes que envía la Raspberry al servidor Neoradix lleguen de forma correcta.

Este router cuenta con un inconveniente y es que la salida a Internet se provee a través de una tarjeta SIM provista por un operador de red. Resultando una tarea muy ardua realizar conexiones remotas a través del protocolo ssh, ya que el router no es capaz de asignar una IP pública a los dispositivos de la red local. Podría habilitarse dicha acción pidiendo permisos al operador en cuestión o usando técnicas más complejas.

4.6 Análisis de vulnerabilidades de la arquitectura del sistema IoT

La arquitectura del sistema es muy compleja y es necesario cubrir cualquier tipo de vulnerabilidad en el sistema y detectarla cuanto antes. Lo arduo del sistema no es solo mantenerlo en funcionamiento durante todo el tiempo, sino también garantizar que está protegido y no pueda sufrir ningún ataque directamente sobre el hardware.

Una solución para proteger los dispositivos hardware es que queden dentro de recintos donde para acceder sea necesaria una autenticación por PIN u otro método que permita autenticar a los usuarios. Además de un control de acceso físico es interesante aportar mayor seguridad a la arquitectura. Una buena práctica es proteger los servicios a los que se puede acceder desde los dispositivos externos, mediante técnicas de autenticación como puede ser el uso de LDAP o RADIUS. Como ya se ha expuesto previamente la Raspberry Pi permite acceso remoto a través de AnyDesk cuyo acceso está protegido y garantiza solo el acceso al usuario autorizado. De forma análoga el acceso desde la red local mediante ssh queda protegido con una contraseña de 16 caracteres para aumentar la resiliencia. Como se ha expuesto previamente cualquier intrusión quedará reflejada en el sistema centralizado de logs, que es la Raspberry pi, una alternativa más eficaz sería el uso de sistema de detección de intrusos (IDS).

Es muy importante tener en cuenta todos los requisitos que deben de satisfacerse en los componentes que conforman el sistema. Una vez se realiza el montaje en campo es necesario tener en cuenta una serie de pasos que son los que se exponen en Ilustración 20.

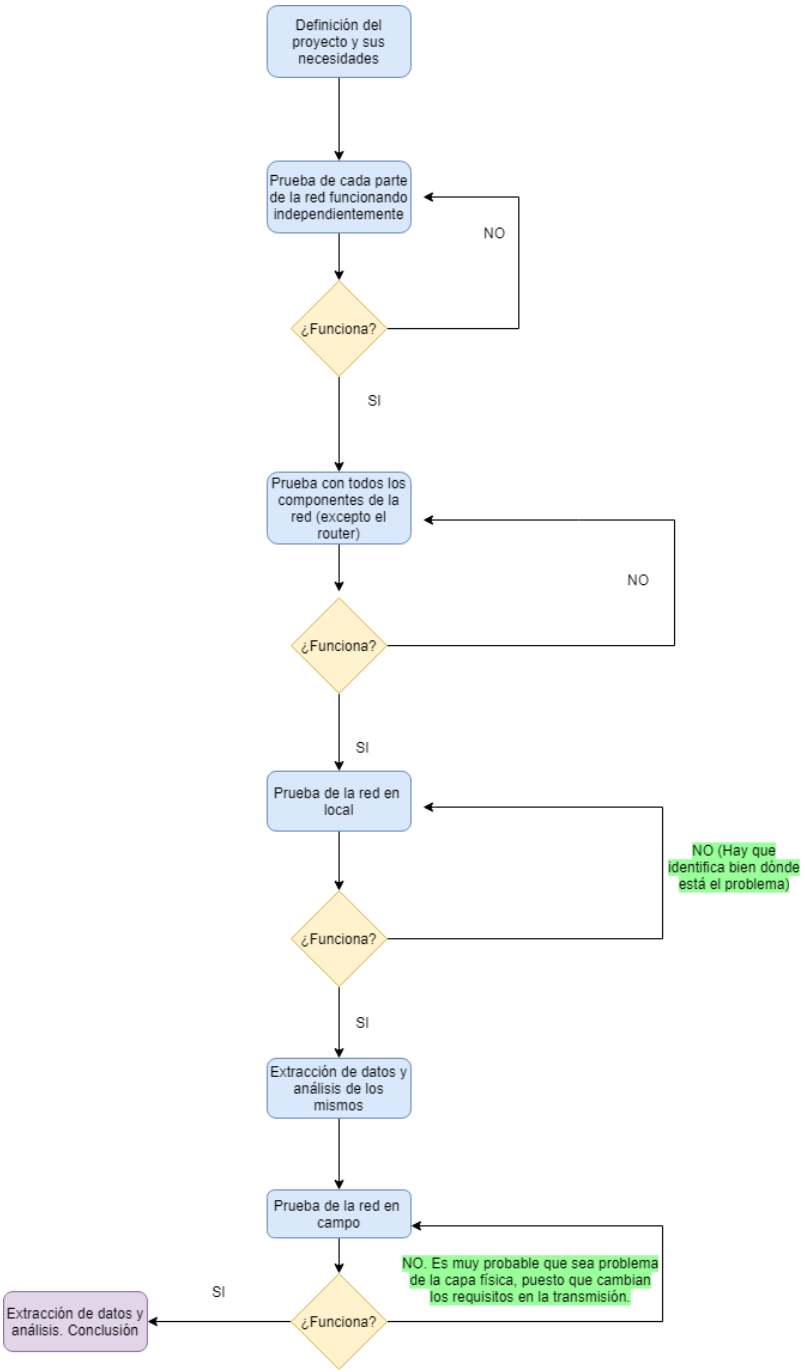

Ilustración 20. Esquema validación de arquitectura

Capítulo 5. Pruebas y resultados

Para el desarrollo de esta solución se ha analizado la robustez de la arquitectura mediante una serie de pruebas. Como se ha visto en los capítulos anteriores, la finalidad principal es analizar el comportamiento de la capa física y el desarrollo de un sistema que pueda trabajar de forma autónoma para la recogida de datos y envío al servidor.

5.1 Pruebas en local

Es necesario, para las pruebas en el laboratorio, realizar un montaje que permita conocer el comportamiento de LoRa, que es la pieza principal de la arquitectura. Para ello, hay que desplegar un módulo receptor, un módulo transmisor y un dispositivo para depurar la comunicación. Por lo tanto, la arquitectura queda como se muestra en la Ilustración 21. El montaje de cada módulo independiente se puede ver en las siguientes figuras.

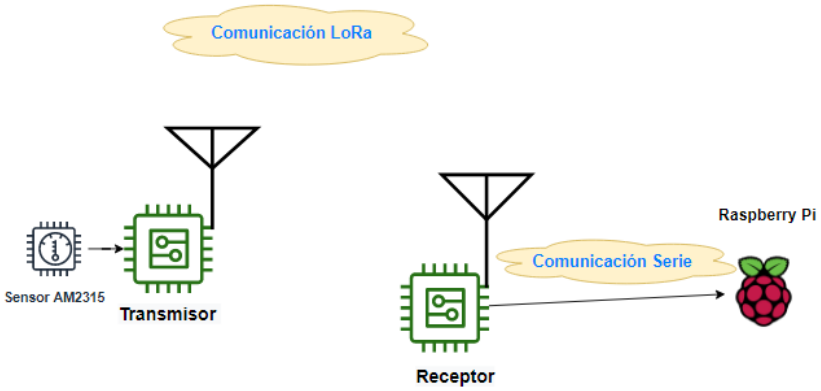

Ilustración 21. Esquema para pruebas en local

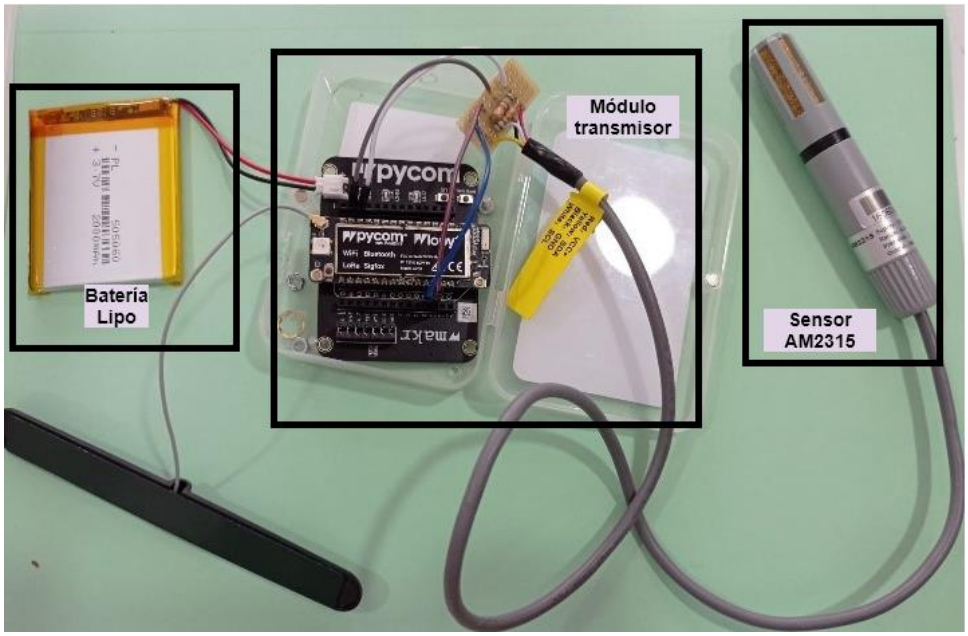

Ilustración 22. Arquitectura del sistema I

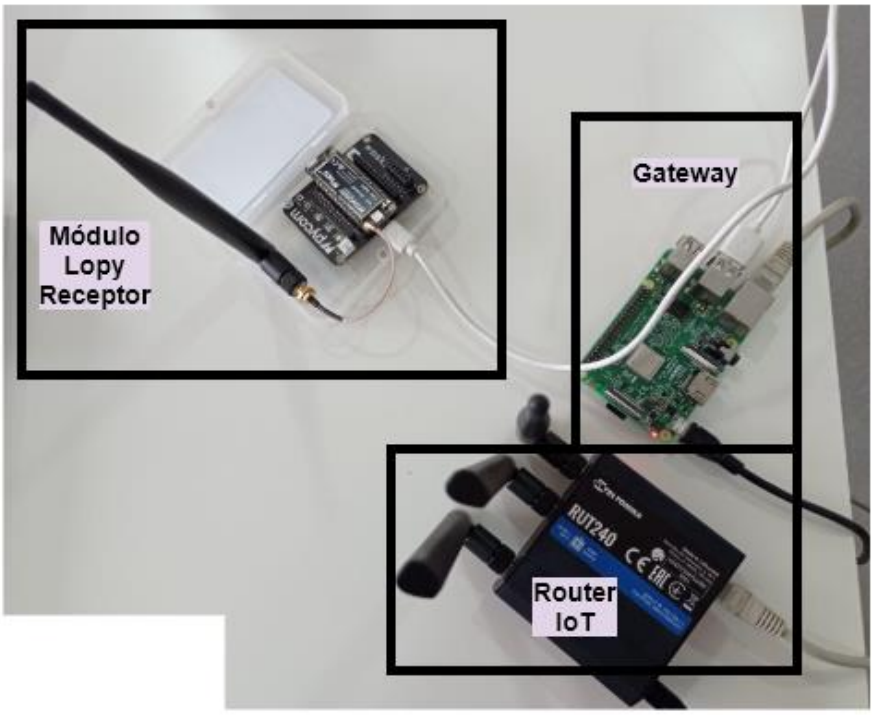

Ilustración 23. Arquitectura del sistema II

Tras unos días de muestreo de datos, se puede observar que el sistema es capaz de estar iniciado de forma continue con las funcionalidades requeridas. Para implementar el sistema y analizar los datos de manera automatizada se ha diseñado un script cuyo flujo se describe en la Ilustración 23, llamado *data_extraction.py;* que se encarga de recoger los datos almacenados del fichero *datos_LoRa.csv*. Este es creado y actualizado por el script *serial_sniffer.py* que permite que cada nueva muestra se almacene en una línea del fichero *datos_LoRa.csv*, donde se alojan los datos que van a analizarse.

Ilustración 24. Flujograma script análisis de datos

En la siguiente gráfica se pueden comprobar las prestaciones de LoRa, analizando el valor de la potencia de la señal que llega al receptor y el porcentaje total de mensajes que llegan con éxito. Como puede observarse en la Ilustración 24, el porcentaje de mensajes recibidos para esta evaluación es de un 100% de paquetes, con una calidad de señal muy buena cuyo valor está en torno a -40 dB de media entre todos los mensajes recibidos, es decir, los enviados. También se analiza de forma empírica cuál va a ser la duración aproximada de la batería. Como resultado obtenemos la estimación de 4 días de duración con la batería de 2000mA, lo que significa una duración suficiente para poder realizar el muestreo de datos.

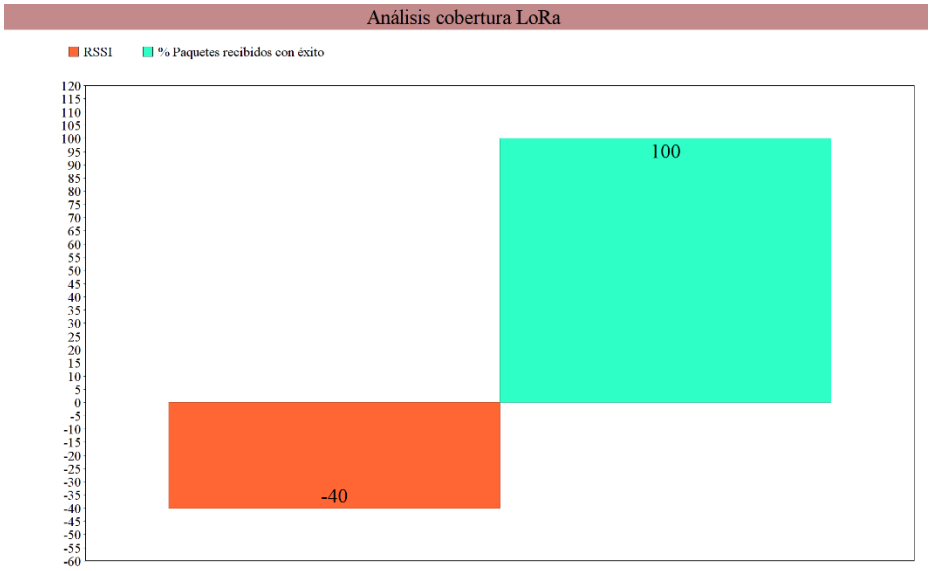

Ilustración 25. Tabla análisis de cobertura LoRa

5.1 Pruebas en campo

Una vez comprobada la cobertura de la red LoRa y el funcionamiento del sistema en local habiendo realizado pruebas durante varios días. Las muestras recogidas se almacenan en la Raspberry Pi, como servidor local. Se va a implementar en esta segunda fase la salida de los datos fuera de la red local, conectándose con un servidor externo. Por lo tanto, para estas

pruebas es necesario contar con dos nuevos componentes: un router IoT y el servidor de Neoradix Solutions. Para así poder ir recogiendo los datos de forma síncrona conforme estos son generados. Una vez que los datos llegan al servidor externo, se puede realizar sobre ellos un análisis más exhaustivo o muestrearlos en una web para dar información sobre cómo es el ambiente donde se encuentra el grano de pienso.

5.1.1 Fase 1

Las pruebas en esta fase se basan en pruebas testear en entornos de indoor, soportando rangos de distancia similares a los que serán necesarios para satisfacer la cobertura necesaria para la fase 2. El montaje es como se describe en la Ilustración 25.

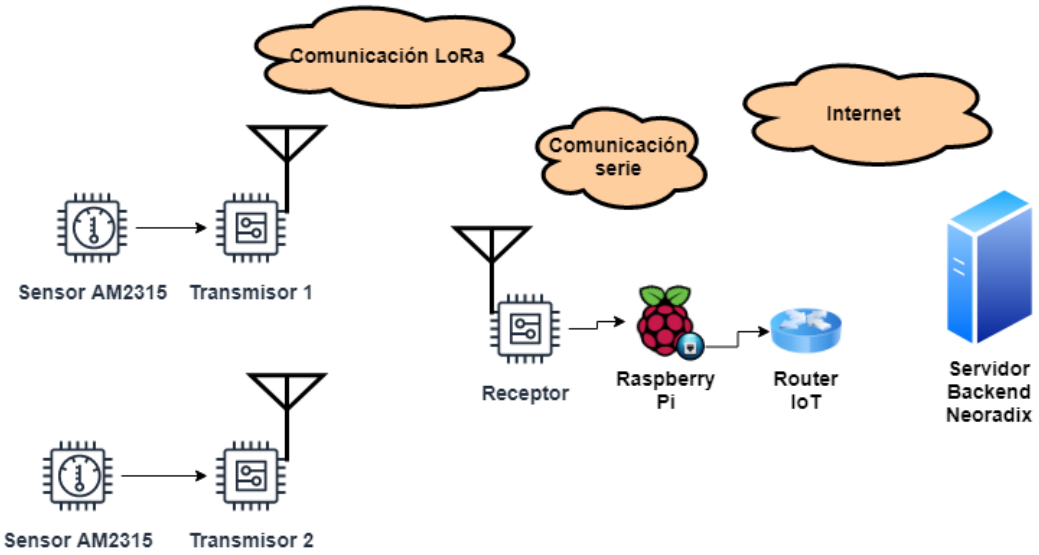

Ilustración 26. Arquitectura completa

Una vez se establece el sistema completo y se haya iniciado la recolección de datos, se lleva a cabo la evaluación de las prestaciones. En nuestro caso, contaremos con dos transmisores espaciados unos 70 metros del receptor. En la Ilustración 26 se muestra los valores del RSSI para ambos, en el caso del transmisor 1 está situado a menor distancia y se consigue una mejor señal en el receptor. Ambos transmisores tienen valores aceptables en esta configuración, aunque el transmisor 2 pierde un 4% de los mensajes que emite como se muestra en la Ilustración 27. Como conclusión se considera que la comunicación es favorable.

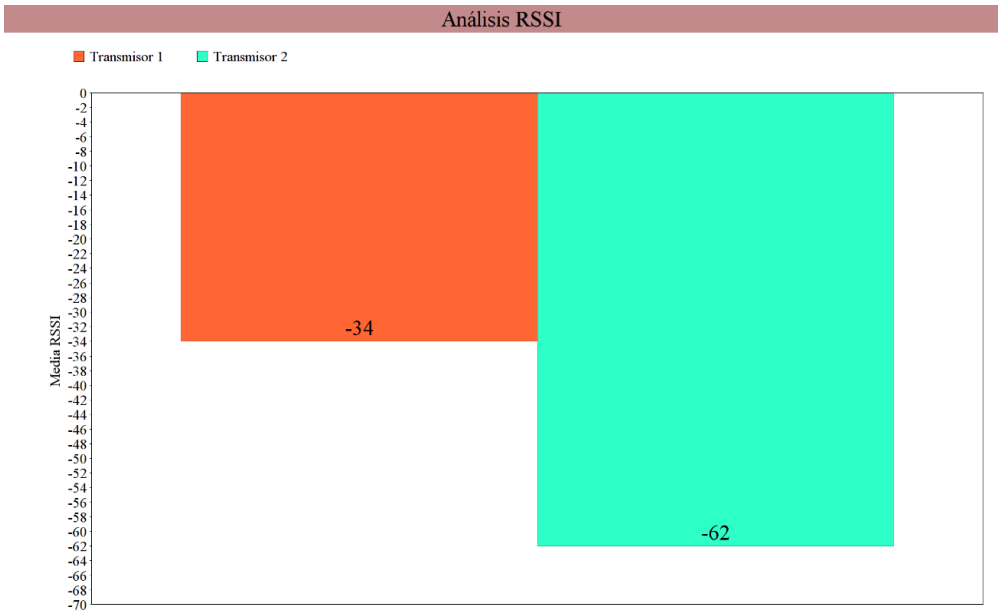

Ilustración 27. Análisis RSSI

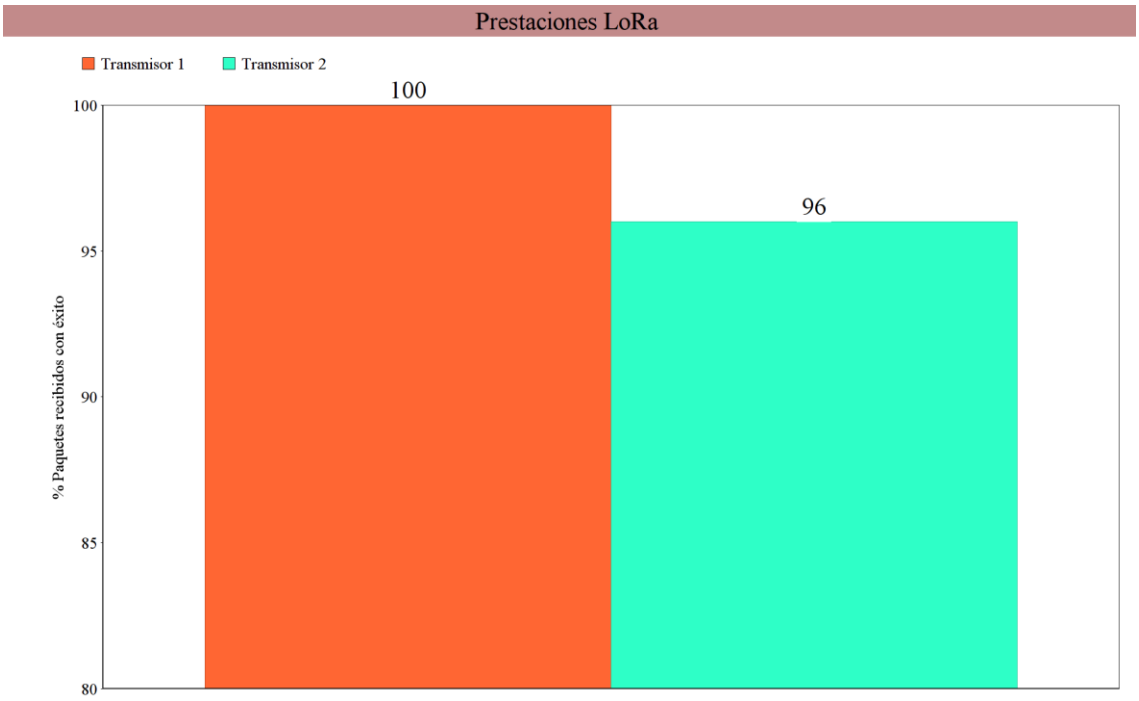

Ilustración 28. Prestaciones LoRa

5.1.2 Fase 2

Esta fase es la implementación en campo para la solución final con la arquitectura que se aprecia en la Ilustración 28 que será implementada en compartimentos de pienso. Este sistema está ideado para que la empresa Neoradix Solutions comience a utilizar estos sistemas y ofrecerlos a los clientes para poder medir la calidad del grano. En este caso, se va a implementar el proyecto en una granja de corderos donde pongamos a prueba la arquitectura para ver cómo se comporta con transmisores embebidos en grano. Las distancias que tiene que cubrir nuestra solución son las mostradas en la Ilustración 29.

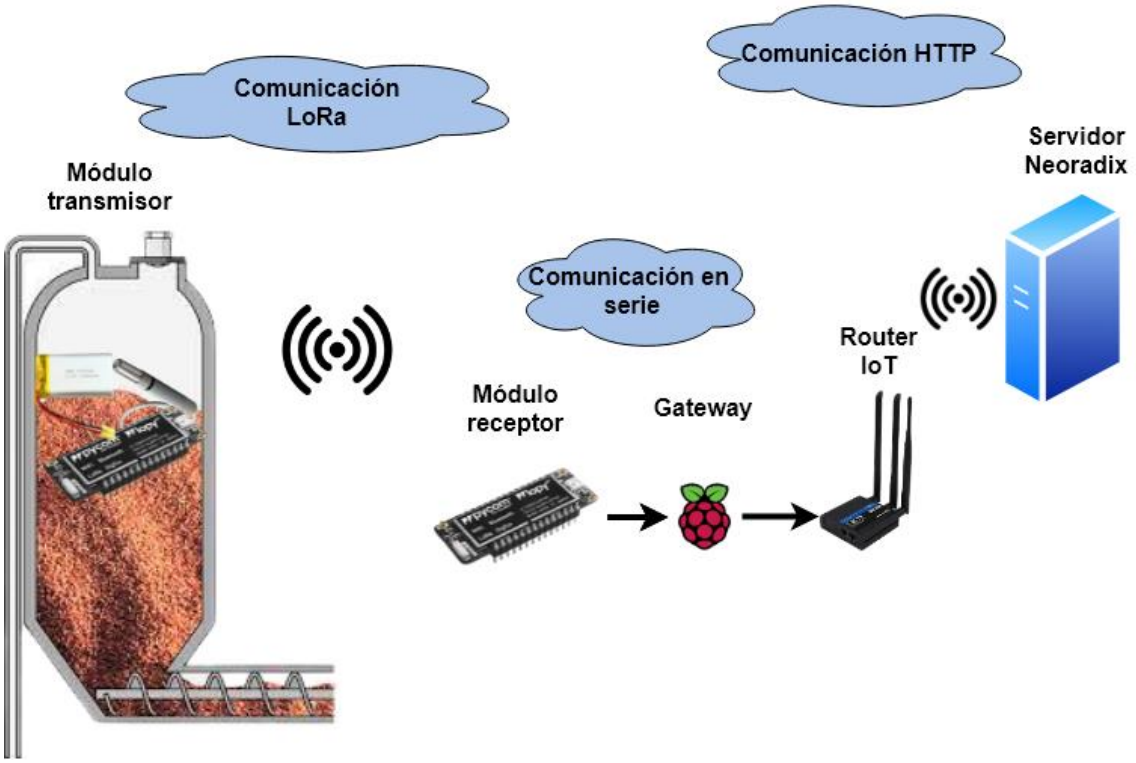

Ilustración 29. Esquema arquitectura en campo

Ilustración 30. Recinto final arquitectura

Capítulo 6. Conclusiones y trabajos futuros

6.1 Conclusiones

Las conclusiones técnicas del funcionamiento del sistema se obtienen a partir del montaje que nos permite analizar cómo de efectiva es la comunicación mediante la red LoRa. Esta tecnología es el principal pilar para analizar nuestro sistema. Verificando que LoRa es efectivo podría ampliarse un mayor número de transmisores y poder controlar de forma eficiente la red. Los parámetros para analizar de esta comunicación son la cantidad de paquetes perdidos y analizando el RSSI.

Después de analizar las muestras recogidas en cada una de las fases del proyecto se ha concluido en que LoRa cumple con creces las expectativas para este tipo de sistemas y es capaz de soportar transmisiones a mayor rango de distancia, aunque se penalizaría otros parámetros como el consumo de los módulos LoRa.

6.2 Grado de consecución de los objetivos y formación adquirida

Resumen de los objetivos propuestos y alcanzados

De los objetivos propuestos para validar la comunicación punto a punto entre módulos LoRa han resultado satisfactorios. Además, se ha ido un paso más allá enfocando este tipo de comunicación a una solución válida como producto para Neoradix Solutions. Con este proyecto se pretende mejorar algunos procesos muy básicos y necesarios en ganadería.

Curva de aprendizaje y principales dificultades encontradas

Para la realización del trabajo ha sido necesario conocer los protocolos de transmisión inalámbrica usados por IoT, para posteriormente ser capaces de implementarlos sobre los módulos elegidos. Estos conocimientos se han ido adquiriendoal realizar el capítulo 3, validación de la capa física.

Conectar todos los componentes de la arquitectura final fue una tarea ardua, puesto que, si fallaba un solo componente en la red, la solución ya no es eficaz. Como conclusión, se establecieron mecanismos de seguridad para comprobar que todo funcionase correctamente.

Reflexión sobre la formación adquirida en conocimientos, metodología y competencias

Durante el desarrollo del proyecto he ido aprendiendo a como depurar este tipo de comunicaciones. Ningún fallo ha sido en vano, puesto que me ha permitido ir aprendiendo y mejorando a lograr los conocimientos y adentrarme dentro de esta tecnología tan novedosa. También me he visto muy motivada con la implementación final, puesto que es un proyecto tangible que podrá ser útil para la sociedad.

6.3 Trabajos futuros

Dentro de este tipo de sistemas hay aún muchas posibles arquitecturas para poder continuar con este proyecto. Algunos posibles trabajos futuros son:

- Añadir un sensor volumétrico, que permite realizar medidas de la cantidad de grano que queda en el recinto y así poder prevenir que se queden sin abastecimiento para los animales.
- Aumentar el número de transmisores, lo que implicará implementar LoRaWAN para poder gestionar este tipo de comunicaciones.
- Realizar un preprocesado de datos dentro de los módulos LoRa realizando tareas de Edge Computing.
- Aplicar algoritmos de predicción para poder predecir el consumo de pienso o cómo va a variar el ambiente del recinto dependiendo de la estación del año o de la hora del día. Para evitar que se llegue a este punto y no malgastar alimento.
- Configurar los módulos transmisores para que solo envíe datos por la red LoRa cuando haya realizado 5 medidas, es decir, estas medidas se deben almacenar en una memoria RAM y luego ser enviadas al receptor para realizar así un consumo de potencia menor.

Bibliografía y referencias

[1] Diapositivas Asignatura Máster Ingeniería Telemática UPCT 2020/2021

[2] LoRa: https://lora-developers.semtech.com/

[3] Sensor AM2315: https://cdn-shop.adafruit.com/datasheets/AM2315.pdf

[4] Lopy 4: https://docs.pycom.io/datasheets/development/lopy4/

[5] Comparativas redes LPWAN: https://behrtechnologies.com/blog/tag/lpwan-comparison/

[6] Semtech chip: https://www.semtech.com/lora

[7] Instalación Anydesk en Raspberry PI: https://pimylifeup.com/raspberry-pi-anydesk/

[8] PM2: https://pm2.keymetrics.io/docs/usage/quick-start/

[9] Router 240 Teltonika: https://teltonika-networks.com/product/rut240/

[10] Anydesk: https://anydesk.com/en/downloads/raspberry-pi