



**COMILLAS**

UNIVERSIDAD PONTIFICIA

ICAI

# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

## DESARROLLO DE UN SISTEMA DE MONITOREO Y CONTROL BASADO EN MQTT PARA REDES DE SENSORES Y ACTUADORES

Autor: Carlos Marín Tutau

Director: Raúl Robledo Cabezuela

Madrid, 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título  
Desarrollo de un Sistema de Monitoreo y Control Basado en MQTT para Redes de  
Sensores y Actuadores en la ETS de Ingeniería - ICAI de la Universidad Pontificia  
Comillas en el curso académico 2024/25 es de mi autoría, original e inédito y  
no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido  
tomada de otros documentos está debidamente referenciada.



Fdo.: Carlos Marín Tutau

Fecha: 05/07/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Raúl Robledo Cabezuela

Fecha: 05/07/2025





# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Desarrollo de un Sistema de Monitoreo y Control  
Basado en MQTT para Redes de Sensores y  
Actuadores

Autor: Carlos Marín Tutau

Director: Raúl Robledo Cabezuela

Madrid, 2025



# **Agradecimientos**

A mis padres y amigos, por su apoyo incondicional y por estar a mi lado durante estos años.



# **DESARROLLO DE UN SISTEMA DE MONITOREO Y CONTROL BASADO EN MQTT PARA REDES DE SENSORES Y ACTUADORES**

**Autor: Marín Tutau, Carlos.**

Director: Robledo Cabezuela, Raúl.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

## **RESUMEN DEL PROYECTO**

Se diseñó e implementó un sistema de comunicación inalámbrica para una red de sensores de gran escala basada en ESP32 y módulos LoRa, orientado a la transmisión de datos de un sensor acelerómetro ADXL345. Tras un análisis comparativo de diferentes protocolos de comunicación, se seleccionaron LoRaWAN para la transmisión a larga distancia y MQTT para la gestión de datos en la nube, logrando una arquitectura eficiente, escalable y de bajo consumo.

**Palabras clave:** Protocolos de comunicación, ESP32, LoRaWAN, MQTT, sensores, consumo energético eficiente, arquitectura IoT.

### **1. Introducción**

El avance de las tecnologías IoT ha generado una creciente demanda en sistemas de comunicación eficientes, confiables y de bajo consumo. Este proyecto plantea el desarrollo de un sistema para una amplia red de sensores destinada a monitorizar las vibraciones en seguidores solares de una planta fotovoltaica, con el objetivo de evitar daños estructurales causados por resonancias provocadas por el viento.

### **2. Definición del Proyecto**

Para llevar a cabo este sistema, se seleccionaron tecnologías y protocolos de comunicación que aseguran una transmisión eficiente con un bajo consumo energético, aspecto clave para garantizar la autonomía y operatividad a largo plazo de la red de sensores.

El sistema propuesto integra dispositivos ESP32 con módulos LoRa, empleando LoRaWAN como protocolo de comunicación y MQTT para la gestión y transmisión de información en la nube.

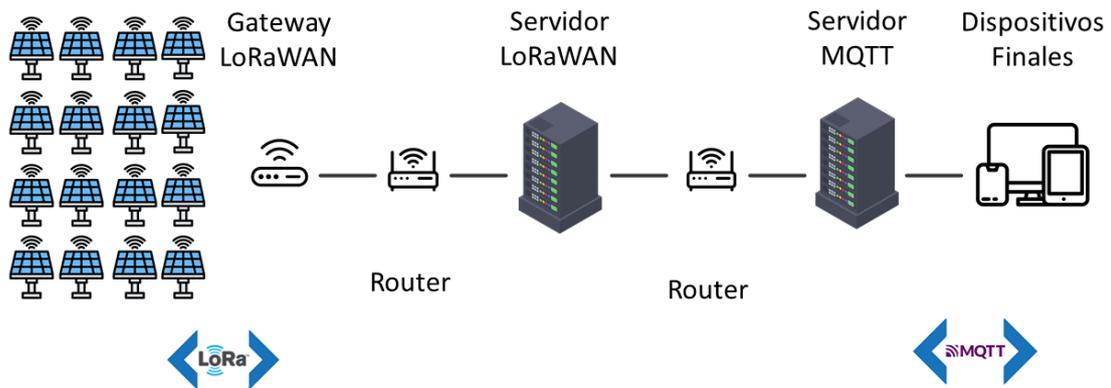
El proyecto incluye tanto la implementación hardware y software del sistema, como el análisis de simulaciones para validar el rendimiento y la viabilidad de la solución planteada.

### **3. Descripción del modelo**

El sistema está compuesto por una amplia red de dispositivos IoT y varios gateways, encargados de transmitir la información recopilada hacia la nube. Estos gateways reciben la información de los nodos, la procesan y adaptan para asegurar una transmisión eficiente.

La comunicación entre el gateway y la red de servidores puede establecerse mediante diferentes tecnologías, como WiFi o conexiones cableadas. En este proyecto se optó por WiFi por su facilidad, aunque el enfoque principal del proyecto está en la elección y justificación de los protocolos empleados.

El sistema cuenta con dos servidores fundamentales: el servidor LoRaWAN, que se encarga de decodificar y gestionar los datos provenientes de los sensores, y el servidor MQTT, responsable de distribuir esta información a los dispositivos finales, de manera eficiente.



#### 4. Resultados

En esta sección se presentan los resultados obtenidos utilizando el simulador LoRaWAN-SIM, así como pruebas físicas del sistema diseñado para una red de 1000 sensores distribuidos en un área de 500 hectáreas. Se evaluaron configuraciones con entre 1 y 5 gateways, obteniendo los siguientes datos:

| Número de Gateways | Consumo energético medio por nodo | PDR (Tasa de entrega de paquetes) | SF (Spreading Factor) medio |
|--------------------|-----------------------------------|-----------------------------------|-----------------------------|
| 1                  | 9.98 mWh                          | 93 %                              | 8.932                       |
| 2                  | 9.14 mWh                          | 97.6 %                            | 8.276                       |
| 3                  | 9.12 mWh                          | 98 %                              | 8.276                       |
| 4                  | 8.9 mWh                           | 98.7 %                            | 8.084                       |
| 5                  | 8.31 mWh                          | 99.3 %                            | 7.592                       |

Por otro lado, se realizaron pruebas con nodos ESP32 y módulos LoRa para validar la comunicación:

- En interiores, a unos 50 metros con obstáculos, se obtuvo una tasa de entrega del 98.68%.
- En exteriores, a 1 kilómetro, en zona urbana sin línea de visión directa, la tasa fue del 100%
- En exteriores, a 2.5 kilómetros, bajo condiciones más complicadas, alcanzó una tasa de 86.9%.

## **5. Conclusiones**

Los resultados obtenidos, tanto en simulaciones como en pruebas físicas, confirman la viabilidad técnica del sistema propuesto para la monitorización distribuida en grandes superficies. La combinación de LoRaWAN para la comunicación de largo alcance y MQTT para la distribución eficiente de datos ofrece una arquitectura fiable, escalable y energéticamente eficiente.

Las simulaciones muestran que el sistema mantiene un buen rendimiento, estimando una autonomía cercana a los 7 años por nodo con baterías de 3000 mAh, lo cual lo hace ideal para aplicaciones IoT de larga duración.

Las pruebas físicas confirman que incluso en entornos urbanos sin línea de visión y con obstáculos, la tecnología LoRa mantiene una elevada tasa de entrega.

# DEVELOPMENT OF A MONITORING AND CONTROL SYSTEM BASED ON MQTT FOR SENSOR AND ACTUATOR NETWORKS

**Author:** Marín Tutau, Carlos.

Supervisor: Robledo Cabezuela, Raúl.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

## ABSTRACT

A wireless communication system was designed and implemented for a large-scale sensor network based on ESP32 and LoRa modules, focused on transmitting data from an ADXL345 accelerometer sensor. After a comparative analysis of different communication protocols, LoRaWAN was selected for long-range transmission and MQTT for cloud data management, achieving an efficient, scalable, and low-power architecture

**Keywords:** Communication protocols, ESP32, LoRaWAN, MQTT, sensors, energy-efficient consumption, IoT architecture.

## 1. Introduction

The advancement of IoT technologies has generated an increasing demand for efficient, reliable, and low-power communication systems. This project proposes the development of a system for a large-scale sensor network aimed at monitoring vibrations in solar trackers of a photovoltaic plant, with the goal of preventing structural damage caused by wind-induced resonances.

## 2. Project Definition

To develop this system, technologies and communication protocols were selected to ensure efficient transmission with low energy consumption, a key aspect to guarantee the autonomy and long-term operation of the sensor network.

The proposed system integrates ESP32 devices with LoRa modules, using LoRaWAN as the communication protocol and MQTT for managing and transmitting information in the cloud.

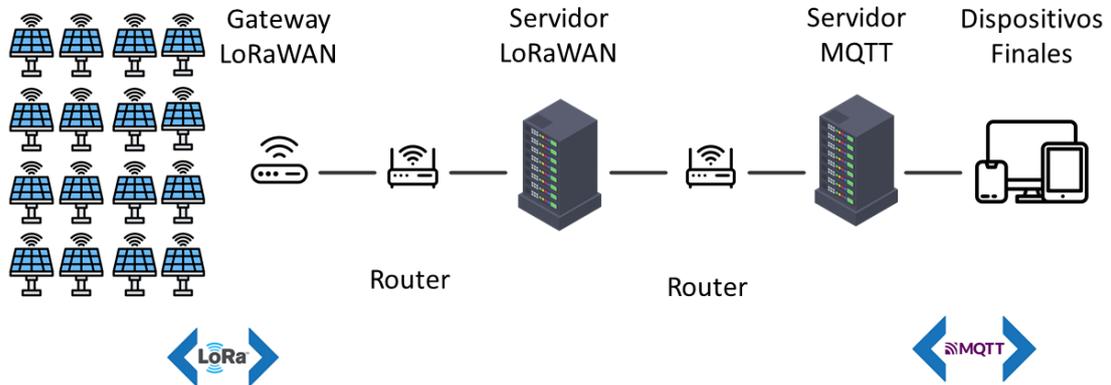
The project includes both the hardware and software implementation of the system, as well as simulation analysis to validate the performance and feasibility of the proposed solution.

## 3. Model Description

The system is composed of a large network of IoT devices and several gateways, responsible for transmitting the collected information to the cloud. These gateways receive data from the nodes, process it, and adapt it to ensure efficient transmission.

Communication between the gateway and the server network can be established through different technologies, such as WiFi or wired connections. In this project, WiFi was chosen for its simplicity, although the main focus of the project is on the selection and justification of the protocols used.

The system includes two fundamental servers: the LoRaWAN server, which is responsible for decoding and managing the data from the sensors, and the MQTT server, responsible for efficiently distributing this information to the end devices.



#### 4. Results

This section presents the results obtained using the LoRaWAN-SIM simulator, as well as physical tests of the system designed for a network of 1,000 sensors distributed over an area of 500 hectares. Configurations with between 1 and 5 gateways were evaluated, yielding the following data:

| Number of Gateways | Average Energy Consumption per Node | PDR (Packet Delivery Ratio) | SF (Average Spreading Factor) |
|--------------------|-------------------------------------|-----------------------------|-------------------------------|
| 1                  | 9.98 mWh                            | 93 %                        | 8.932                         |
| 2                  | 9.14 mWh                            | 97.6 %                      | 8.276                         |
| 3                  | 9.12 mWh                            | 98 %                        | 8.276                         |
| 4                  | 8.9 mWh                             | 98.7 %                      | 8.084                         |
| 5                  | 8.31 mWh                            | 99.3 %                      | 7.592                         |

Additionally, tests were conducted with ESP32 nodes and LoRa modules to validate communication:

- Indoors, at about 50 meters with obstacles, a delivery rate of 98.68% was achieved.
- Outdoors, at 1 kilometer in an urban area without direct line of sight, the delivery rate was 100%.
- Outdoors, at 2.5 kilometers under more challenging conditions, a delivery rate of 86.9% was reached.

## **5. Conclusions**

The results obtained, both from simulations and physical tests, confirm the technical feasibility of the proposed system for distributed monitoring over large areas. The combination of LoRaWAN for long-range communication and MQTT for efficient data distribution provides a reliable, scalable, and energy-efficient architecture.

Simulations indicate that the system maintains strong performance, with an estimated autonomy of nearly 7 years per node using 3000 mAh batteries making it ideal for long-term IoT applications.

Physical tests further validate that even in urban environments without line of sight and with obstacles, LoRa technology sustains a high packet delivery rate.

## *Índice de la memoria*

|   |           |
|---|-----------|
| <i>Capítulo 1. Introducción .....</i>                         | <i>4</i>  |
| <i>Capítulo 2. Descripción de las Tecnologías.....</i>        | <i>7</i>  |
| <i>Capítulo 3. Estado de la Cuestión.....</i>                 | <i>15</i> |
| <i>Capítulo 4. Simulaciones y Validación del sistema.....</i> | <i>36</i> |
| 4.1 Simulación del protocolo LoRaWAN .....                    | 36        |
| 4.2 Implementación física y validación del sistema.....       | 51        |
| 4.3 Integración del módulo mqtt .....                         | 55        |
| <i>Capítulo 5. Alineación del Proyecto con los ODS.....</i>   | <i>60</i> |
| <i>Capítulo 6. Conclusiones y Trabajos Futuros.....</i>       | <i>62</i> |
| <i>Capítulo 7. Bibliografía.....</i>                          | <i>64</i> |

## *Índice de figuras*

|  |    |
|--|----|
| Figura 1. Dispositivos IoT (en billones) esperados a nivel mundial de 2015 a 2030 según STATISTA[1]..... | 7  |
| Figura 2. [2] .....  | 8  |
| Figura 3. Modificación del autor a partir de una imagen descargada [7].....                              | 11 |
| Figura 4. Modificación del autor a partir de una imagen descargada [8].....                              | 13 |
| Figura 5. Diseño realizado por el autor .....  | 13 |
| Figura 6. Capas OSI. Imagen modificada a partir de una imagen extraída [8]. .....                        | 16 |
| Figura 7. Diseño realizado por el autor (Bluetooth).....   | 18 |
| Figura 8. Diseño realizado por el autor (WiFi) .....   | 19 |
| Figura 9. Diseño realizado por el autor (ZigBee).....  | 21 |
| Figura 10. Diseño realizado por el autor (Sigfox).....   | 23 |
| Figura 11. Diseño realizado por autor (NB-IoT).....  | 24 |
| Figura 12. Diseño realizado por autor (LoRaWAN) .....  | 26 |
| Figura 13. Imagen extraída de artículo [20]. .....   | 27 |
| Figura 14. Imagen extraída de artículo [20]. .....   | 28 |
| Figura 15. Imagen extraída de artículo [21] .....  | 28 |
| Figura 16. Diseño realizado por autor (AMQP).....  | 32 |
| Figura 17. Diseño realizado por autor (HTTP) .....   | 33 |
| Figura 18. Diseño realizado por autor (MQTT) .....   | 34 |

## *Índice de tablas*

Table 1. Comparativa con respecto a número de Gateways..... 46

## Capítulo 1. INTRODUCCIÓN

Las telecomunicaciones están presentes en nuestro día a día y son utilizadas en numerosos ámbitos donde realmente facilitan la vida del ser humano. Es muy impactante, como la unión de numerosos sensores pueden salvar vidas humanas, detectando posibles situaciones perjudiciales y aprender de esta información recopilada para saber de qué manera actuar ante dicha situación.

Se plantea una infraestructura, donde debido a externos factores, como puede ser el viento, se inclina un grado con respecto al suelo por año. Los habitantes del edificio no percibirán el cambio que este está experimentando, sin embargo, tras un periodo de 10 años, es probable que la estructura no presente una inclinación de 10 grados, sino que se encuentre derrumbada, lo que implicaría elevados costes económicos y riesgos significativos para la seguridad de las personas. Esta situación no, es imaginaria, es una de las muchas donde este sistema de control y monitoreo a desarrollar en este proyecto ayudará a prevenir y a saber actuar.

Juntar mi pasión por la tecnología a la vez que hacer el bien por la sociedad no es menos que una motivación, es por ello que este proyecto hará uso de las tecnologías más adecuadas, a la vez que abrirá un campo donde personas estudien el rendimiento de estas para conseguir los mejores resultados posibles.

Toda la información recolectada por esta tecnología, será muy valiosa para analizar y entender como este sistema deberá ser desarrollado, intentando encontrar patrones ante situaciones anómalas para anticiparse a cualquier posible situación peligrosa.

Se hará uso de dispositivos IoT “Internet of Things”, dispositivos capaces de transmitir y recibir información. Es interesante como numerosos dispositivos pueden intercambiar datos, sin embargo, es fundamental comprender como se lleva a cabo esta comunicación. Antes de continuar con la explicación, hay que mencionar que esta información captada por los

sensores será transmitida hacia la nube, donde está la información se transmitirá a los dispositivos finales como puede ser el ordenador de una oficina. Tras un profundo estudio sobre que protocolo utilizar, se adelanta que dichos protocolos los cuales forman la base de este proyecto serán los protocolos LoRaWAN y MQTT, donde realmente se hablará en profundidad sobre ellos y se explicará porque son los protocolos utilizados en este proyecto.

Por ello se propone el desarrollo de un sistema de monitoreo y control basado en MQTT para redes de sensores y actuadores. Este proyecto será mayoritariamente teórico, debido a el complicado acceso de todos los recursos necesitados para llevarlo a cabo. Lo cual, no menos, se realizarán numerosas simulaciones para comprobar el funcionamiento de lo propuesto.

Para que esta amplia red de sensores, distribuida a lo largo de un gran espacio físico, funcione de manera eficaz, es fundamental que estos dispositivos cuenten con una larga autonomía. Imaginemos el impacto que conllevaría si estos sensores requirieran revisiones frecuentes. Esto implicaría un gran consumo de tiempo y recursos, convirtiendo este proyecto poco rentable a largo plazo. Por ello, uno de los objetivos clave durante el desarrollo de este proyecto es garantizar un consumo energético eficiente, prolongando así la vida útil de los dispositivos.

El proyecto se ha presentado de manera general para mostrar una idea de las posibles situaciones donde este puede ser aplicado. Una de estas aplicaciones, y la desarrollada en este trabajo es la sensorización de 1.000 seguidores solares de una planta fotovoltaica distribuidos en un área de 500 hectáreas. Esto permite el monitoreo de las vibraciones cuantificando así los posibles daños estructurales y corregir el posicionamiento de los seguidores solares para evitar resonancias en las vibraciones provocadas por el viento.

El coste de un seguidor solar en el mercado es elevado, situándose en torno a 1000 euros por unidad. En nuestro caso, al contar con un total de 1.000 seguidores solares, equivalen a un valor aproximado a 1 millón de euros. Esta cifra es un promedio, ya que en el mercado existen modelos bastante más caros.

El puente de Tacoma Narrows comenzó a balancearse hasta derrumbarse debido al fenómeno de resonancia. Este fenómeno ocurre cuando la frecuencia de las oscilaciones de una estructura coincide o es muy parecida con la frecuencia del viento, lo que provoca un incremento significativo en la amplitud de las vibraciones de la infraestructura. En este caso, las vibraciones crecieron descontroladamente hasta causar el derrumbe del puente.

Este mismo principio se aplica a los seguidores solares de una planta fotovoltaica. Estos pueden entrar en resonancia causando fallos catastróficos. Es por ello que se implementará un sistema de sensorización que monitorice las vibraciones estructurales de los seguidores solares. Cuando dicho sistema detecte condiciones cercanas a la resonancia, modificará la posición de estos seguidores, modificando así la frecuencia natural de las estructuras, alejándola de la frecuencia del viento y por lo tanto evitando entrar en resonancia.

Esta frecuencia natural de la estructura, es básicamente una medida de velocidad de estas vibraciones en dicha estructura tras aplicar una fuerza externa. Existe una relación entre la

frecuencia natural, la masa y la rigidez de una estructura, definida como:  $f = \frac{1}{2\pi} * \sqrt{\frac{k}{m}}$  siendo k la rigidez y m la masa de dicha estructura.

## Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

El internet de las cosas (IoT) es un concepto que hace referencia a la conexión de objetos físicos a internet, lo que les permite transmitir y recibir datos. Estos objetos pueden ser muy diversos, desde relojes inteligentes y sensores hasta un localizador de mascota, siempre que cuenten con la capacidad de conectarse y comunicarse. Todos estos dispositivos inteligentes conforman el núcleo de una nueva era tecnológica, en la que la interacción entre el mundo físico y digital es esencial [1].

Estos dispositivos son la llave para convertir una simple ciudad en una ciudad inteligente, mejorando así la calidad de vida de los ciudadanos. Imaginemos una ciudad donde los semáforos se adapten al tráfico de la carretera, reduciendo tiempos de espera innecesarios.

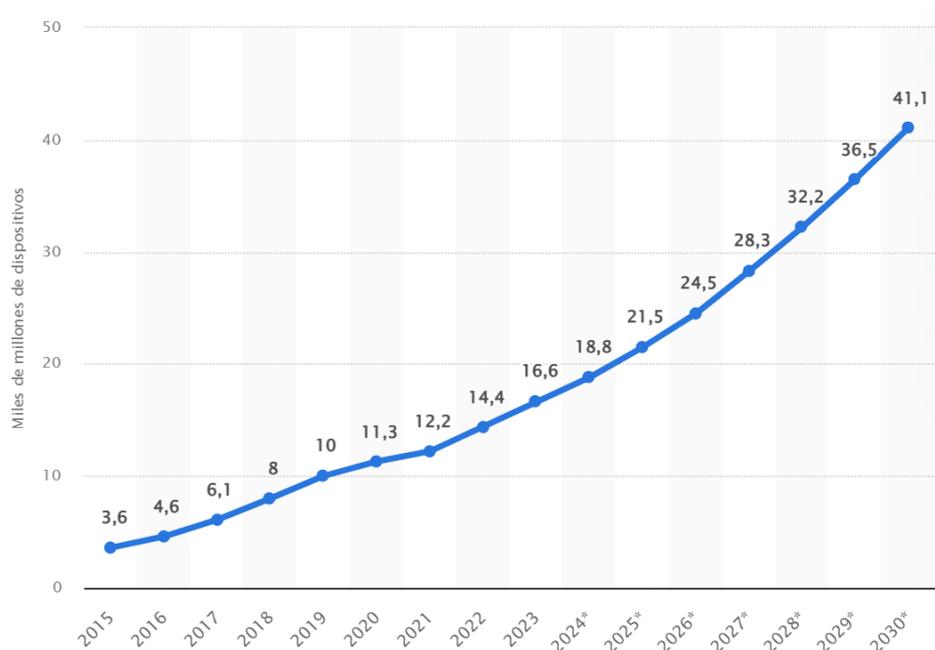
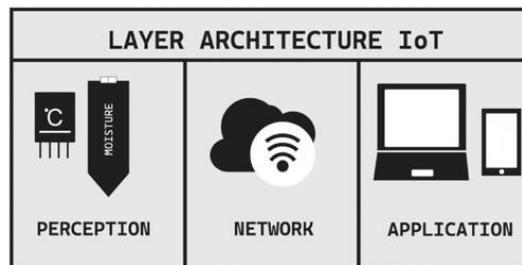


Figura 1. Dispositivos IoT (en billones) esperados a nivel mundial de 2015 a 2030 según STATISTA [2].

Tras mostrar una proyección esperada del crecimiento de dispositivos IoT en los próximos años, es importante destacar el aumento significativo esperado a nivel global. Esto sugiere que estos dispositivos estarán cada vez más integrados en nuestro entorno cotidiano.

Estos dispositivos no están exentos de desafíos. Presentan sus obstáculos relacionados con el uso de la energía, la posible saturación de la red internet debido al elevado volumen de transmisión de datos y sus vulnerabilidades inherentes a su conectividad. Sin embargo, la evolución de los protocolos de comunicación permite optimizar su funcionamiento, adaptarlos a distintos contextos y mejorar tanto su rendimiento como su eficiencia.

Para el desarrollo de este proyecto, es imprescindible entender cómo funciona la arquitectura de estos dispositivos inteligentes.



*Figura 2. [3]*

- **Capa de percepción:** También conocida como capa física, se encuentran los dispositivos de captura y recolección de datos del entorno físico, convirtiendo señales físicas en información digital haciendo posible la transmisión a través del sistema. En este proyecto, esta capa está constituida por el gran número de sensores distribuidos sobre 1000 seguidores solares, encargados de monitorizar las vibraciones y actuar sobre ellos para evitar los grandes efectos negativos que la resonancia puede provocar.

Estos sensores, utilizarán una frecuencia de muestreo de 20Hz, es decir, cada sensor capturará información cada  $\frac{1}{20} = 0.05$  segundos. Imaginemos la cantidad de

información recolectada a lo largo de un periodo de tiempo no excesivamente largo. Es por ello, que este proyecto está centrado en cómo hacer esto de la manera más eficaz posible, enviando toda esta información a la red a través de los protocolos más adecuados, utilizando el menor consumo posible, asegurando la sostenibilidad del sistema.

- **Capa de red:** Como se ha mencionado previamente, esta capa es la responsable de la transmisión y gestión de toda la información recopilada. Su función principal es garantizar el transporte eficiente y fiable de los datos recopilados en la capa de percepción. En este proyecto se ha optado por el protocolo de comunicación LoRaWAN, conocido como (Long Range Wide Area Network), diseñado específicamente para aplicaciones que requieren comunicación a largas distancias con bajo consumo energético, ideal para redes de sensores distribuidos en grandes superficies.

Este protocolo permite la transmisión de datos a grandes distancias haciendo uso de una potencia baja, lo que maximiza la durabilidad de los dispositivos y la cobertura. Es especialmente útil en entornos rurales donde normalmente las tecnologías tradicionales suelen presentar limitaciones. En el caso de este proyecto, se aplicará en un campo fotovoltaico [3].

Dentro de la arquitectura de LoRaWAN, es importante entender el funcionamiento del Gateway, también conocido como puerta de enlace. Actúa como intermediario entre los sensores y el servidor central. Esta estructura evita que cada sensor tenga que comunicarse directamente con Internet, lo que implicaría un elevado consumo energético y, por tanto, iría en contra de los objetivos de eficiencia de este proyecto. Es importante mencionar que toda la información transmitida de los sensores a dicho Gateway se realiza mediante radiofrecuencia en bandas de frecuencia sin licencia como se ha mostrado anteriormente, utilizando una modulación de radio basada en

CSS (Chirp Spread Spectrum) lo que permite transmitir datos a baja potencia y largas distancias [4].

Una vez el Gateway recibe toda la información recopilada por los sensores, este se encarga de transmitir toda esta información a la red. Este enviará toda la información al servidor de LoRaWAN a través de WiFi. Para este enlace, se utiliza UDP/IP, del cual se hablará detalladamente más adelante.

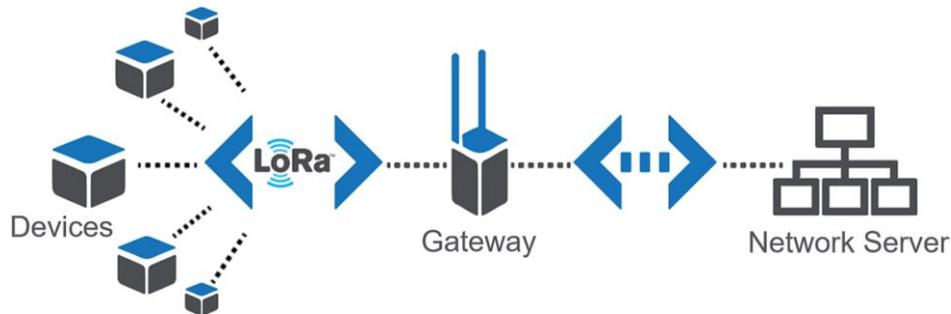
Cuantas más puertas de enlace (Gateways) se agreguen, mayor será su resiliencia en la arquitectura. En caso de que uno de ellos falle, otros gateways pueden asumir la comunicación asegurando la continuidad en la comunicación.

Además, el Gateway es un dispositivo bidireccional, es decir, no solo recibe información de los sensores y la reenvía hasta la red, sino que también es capaz de transmitir información del servidor hacia los sensores.

Un ejemplo sería: Un seguidor solar queda inoperativo tras una avería, este sigue teniendo su sensor asociado. Desde el servidor, se podría transmitir un comando para que el Gateway modifique los parámetros del sensor asociado hasta que el seguidor solar haya sido reparado.

En escenarios donde el Gateway quiera transmitir información a dicho dispositivo, y este dispositivo no esté activo en ese momento, la puerta de enlace (Gateway), retendrá este mensaje hasta que el dispositivo haya sido reactivado, entregando el mensaje tan pronto como se recupere su actividad [5].

Cuando se menciona que la información es enviada a la nube, se hace referencia al servidor de LoRaWAN. Este servidor, no solo gestiona y enruta la comunicación, sino que también aplica medidas de seguridad, como el cifrado de extremo a extremo asegurando la integridad y confidencialidad de la información transmitida [6].



*Figura 3. Modificación del autor a partir de una imagen descargada [7].*

Como se puede observar en la imagen, la información recolectada por los sensores será transmitida al Network Server, que en este caso corresponde al Servidor de Red LoRaWAN. Una vez procesados los datos provenientes de los dispositivos IoT, estos serán enviados al servidor MQTT, dando paso a la siguiente capa dentro de la arquitectura IoT.

- **Capa de aplicación:** Es la capa encargada de presentar toda la información recibida en el servidor de red, que en nuestro caso corresponde al servidor MQTT. Su función es que los usuarios interpreten la información recolectada por los dispositivos IoT, facilitando la toma de decisiones. Esta representación, se visualizará y monitoreará en tiempo real.

Se empleará el protocolo MQTT (Message Queueing Telemetry Transport), que permite la comunicación a través de un modelo de publicación/subscripción. Este enfoque permite que los dispositivos solo reciban la información de aquellos temas a los que estén suscritos, reduciendo el tráfico innecesario y optimizando los recursos.

A continuación, se mostrará un sencillo ejemplo para comprender el modelo:

El servidor LoRaWAN enviará los datos en un formato JSON hacia el servidor MQTT.

```
{  
  
  "dispositivo": "sensor1"  
  
  "vibración": {  
  
    "deltaX" : 2.5,  
  
    "deltaY" : 12  
  
  }  
  
}
```

Este es un ejemplo del formato de datos que enviará el sensor número 1, donde, cada sensor tendrá su propio identificador. Toda esta información, será publicada en una sección o tema, por ejemplo, la sección sensores/medidas.

Únicamente aquellos clientes que estén suscritos a la sección sensores/medidas recibirán esta información. El bróker MQTT actúa como intermediario, distribuyendo los mensajes a los suscriptores correspondientes. Esta arquitectura permite interpretar y presentar los datos de manera clara y comprensible.

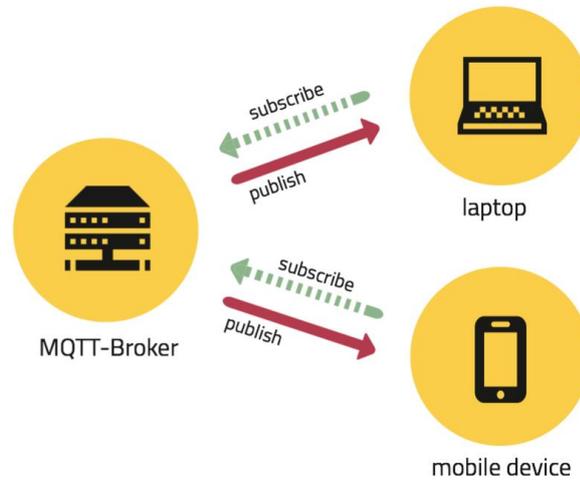


Figura 4. Modificación del autor a partir de una imagen descargada [8].

Una vez explicado el funcionamiento de las distintas capas, y los protocolos a utilizar, se mostrará a continuación una representación gráfica mostrando el funcionamiento del sistema.

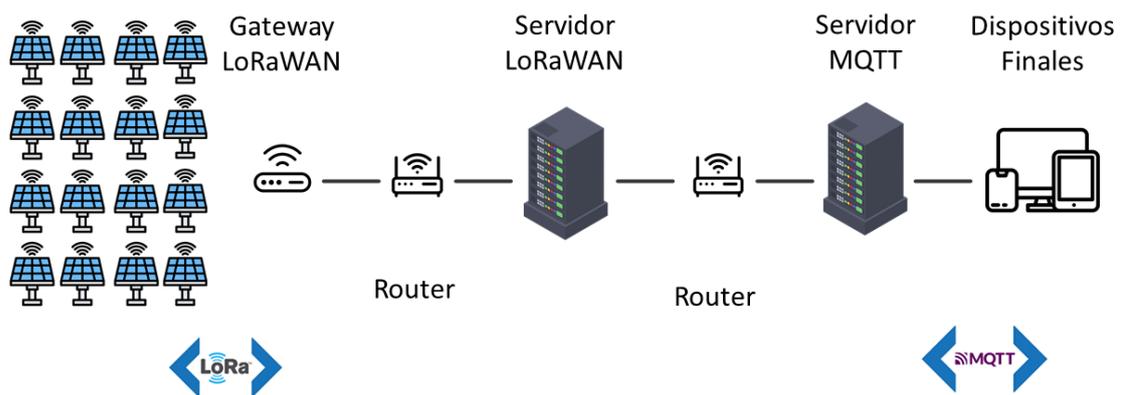


Figura 5. Diseño realizado por el autor

En muchos casos, los dispositivos IoT no utilizan el protocolo IP, lo que hace imprescindible el uso de un gateway, que actúa como puente entre los dispositivos IoT y la nube.

Añadir que la comunicación entre routers se puede realizar de distintas maneras, ya sea conexiones inalámbricas WiFi (Wireless), cableadas u otros tipos de comunicación por radiofrecuencia. En este proyecto, se hará uso de WiFi por simplicidad, pero, hay que destacar que el foco de este proyecto no está en el método de transmisión entre routers, si no en la elección y justificación de los protocolos utilizados.

Como presenta el diseño adjunto, el sistema hará uso de dos servidores, el servidor LoRaWAN y el servidor MQTT. Ambos son necesarios, ya que los sensores transmiten la información al Gateway en un formato específico y este al servidor correspondiente. Una vez recibida, la información es procesada, gestionada y decodificada en el servidor LoRaWAN, que la adapta para ser reenviada al servidor MQTT, encargado de distribuirla de forma eficiente a los dispositivos finales.

## Capítulo 3. ESTADO DE LA CUESTIÓN

En esta parte del proyecto, se hará un análisis exhaustivo de las características de cada una de las tecnologías, protocolos de comunicación y gestión energética aplicable a la solución de este proyecto. El objetivo principal de esta sección será evaluar las posibles opciones disponibles en el ámbito del Internet de las Cosas (IoT), identificando sus ventajas y limitaciones. Además, se aprovecharán casos de uso relevantes, es decir, estudios previos y soluciones existentes en el mercado, donde abordan problemas y se ofrecen soluciones en el ámbito del monitoreo de infraestructuras y el uso de dispositivos IoT, ofreciendo soluciones similares a las buscadas en este proyecto.

Este análisis justificará la necesidad de desarrollar una solución específica basada para el control y monitoreo de seguidores solares, donde la transmisión de datos, eficiencia energética, escalabilidad y la seguridad de estos dispositivos serán aspectos cruciales y determinantes en la elección post análisis. La solución seleccionada, será de gran importancia para garantizar el éxito del sistema propuesto.

La arquitectura IoT permite entender la organización del sistema propuesto este proyecto, pero a la hora de realizar el análisis entre los posibles protocolos a utilizar, es necesario hacerlo con mayor profundidad entendiendo el modelo de capas de red, que divide la comunicación en diferentes capas (Física, Enlace, Red, Transporte, Sesión, Presentación y Aplicación).

A lo largo de esta parte, nos centraremos en las siguientes capas clave de la arquitectura de comunicación:

- Capa Física
- Capa de Enlace de Datos
- Capa de Red
- Capa de Aplicación

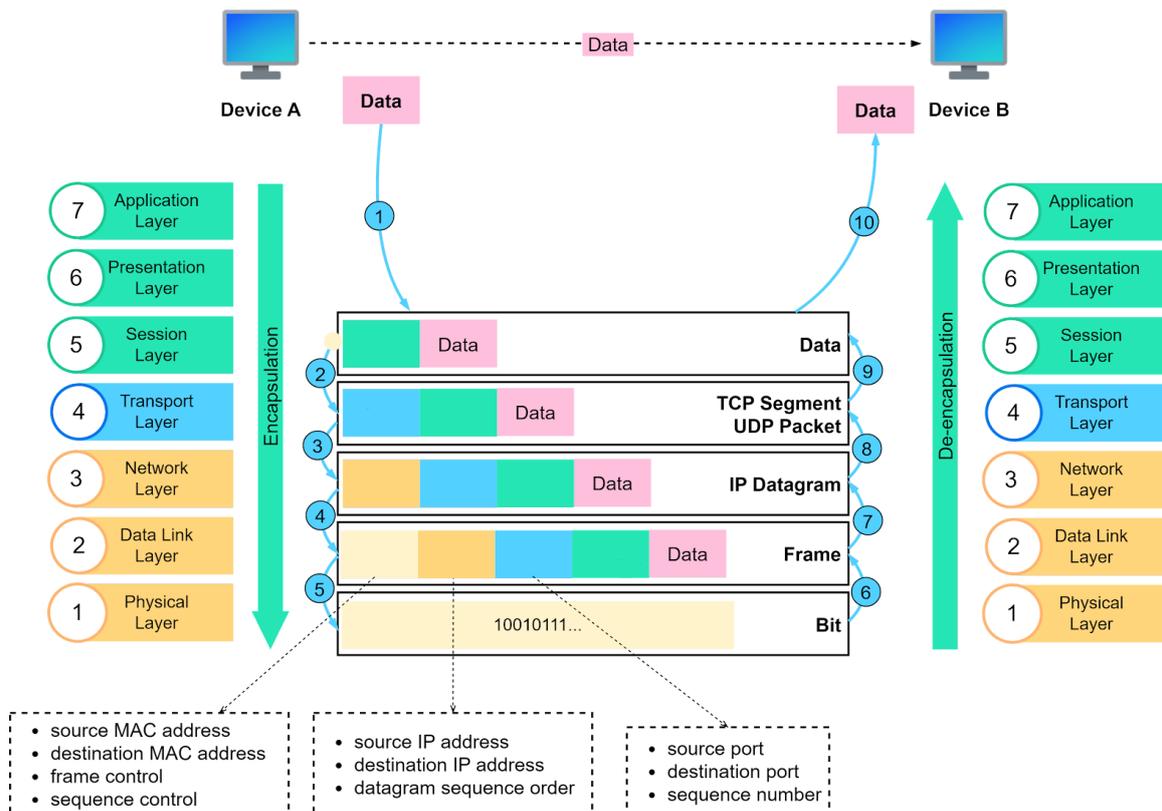


Figura 6. Capas OSI. Imagen modificada a partir de una imagen extraída [8].

La imagen muestra cómo se realiza la comunicación entre dos dispositivos finales, los cuales utilizan todas las capas del modelo OSI. Dependiendo del tipo de dispositivo involucrado, se utilizarán más o menos capas durante la comunicación.

Por ejemplo, en una comunicación entre un sensor, (dispositivo IoT, como los utilizados en este proyecto) y un Servidor de LoRaWAN, la operación se limita únicamente en las capas de Red, Enlace y Física.

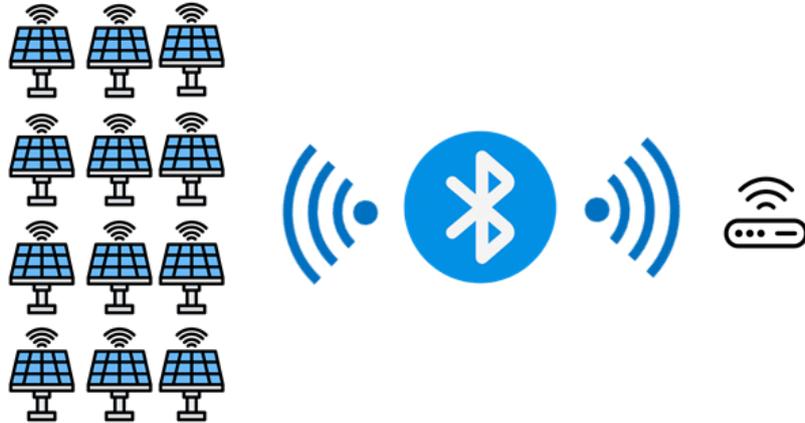
Cuando los datos son enviados, este paquete viajará a través de las diferentes capas del modelo, donde cada una añade una cabecera con información específica para que los

dispositivos puedan procesarlo de manera correcta, este proceso es conocido como encapsulación.

Una vez el paquete llega al destino como se muestra en la imagen superior, este proceso será el inverso, eliminando las cabeceras de las distintas capas hasta recuperar la información original. Este proceso es conocido como desencapsulación.

En la arquitectura de red, cada capa del modelo OSI cumple una función específica para garantizar que los datos se transmitan de manera eficiente y confiable entre dispositivos. En esta sección del proyecto, se harán dos tipos de análisis. En el primero, se hará un estudio de la transmisión de información desde los dispositivos IoT hasta el gateway. El segundo análisis estará centrado en la transmisión de datos entre el servidor de aplicación y los dispositivos finales.

Cabe mencionar que los protocolos de IoT garantizan que la información recopilada por los dispositivos IoT sea leída y comprendida por otro dispositivo, ya sea una puerta de enlace o un servicio [7].



*Figura 7. Diseño realizado por el autor (Bluetooth)*

El protocolo Bluetooth destaca por su bajo consumo energético, especialmente cuando se hace uso de BLE (Bluetooth Low Energy). Tiene un coste de implementación reducido, ya que es posible encontrar Gateways bluetooth cerca de los 10€. Su implementación es bastante sencilla, únicamente emparejando los dispositivos IoT al Gateway y este al router. El proceso es tan intuitivo como conectar unos auriculares inalámbricos al teléfono.

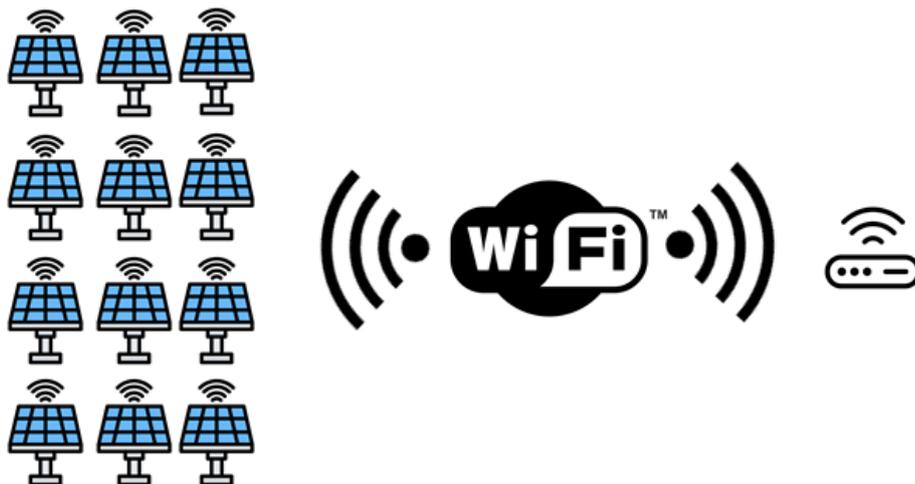
Desde el punto de vista de modelo OSI, Bluetooth opera en las siguientes capas:

1. Capa física: Transmite ondas de radio en la banda de frecuencia de 2.4 GHz entre los dispositivos IoT y el Gateway [8].
2. Capa de enlace: En esta capa se utiliza el protocolo L2CAP, donde aspectos como la sincronización y el control de errores son gestionados.
3. Capa de Red: Se debe destacar que el protocolo Bluetooth, no utiliza el direccionamiento IP. Es por ello que el uso del Gateway es necesario para poder transmitir esta información a lo largo de la red.

Se ha decidido no coger este protocolo para nuestro proyecto debido principalmente a varias limitaciones clave como son:

1. Alcance limitado. Este protocolo tiene un rango de operación corto, entre 10 y 100 metros. Nuestro proyecto requiere que estos dispositivos sean capaces de transmitir información en distancias mayores. Como se ha mencionado, los sensores están distribuidos a lo largo de una producción fotovoltaica en un área de 500 hectáreas.
2. La capacidad limitada de conexiones simultaneas, es otro de los factores que se suma a la decisión de no coger este protocolo para este proyecto [9].
3. Finalmente, este protocolo tiene una baja velocidad de transmisión con respecto a otros protocolos disponibles como son Wifi o LoRaWAN.

Tras este análisis, se concluye que Bluetooth no es el protocolo más adecuado para aplicaciones IoT de largo alcance. Sin embargo, es una gran opción en entornos de corto alcance. En el ámbito de la salud como puede ser rastrear una actividad física a través de un reloj inteligente o una pulsera dedicada al fitness, estos datos son transmitidos a un teléfono móvil cercano. También es ampliamente utilizado en la automatización del hogar, donde la comunicación es en un entorno cerrado y cercano. Un ejemplo de esto sería una cerradura inteligente de una casa, que puede ser desbloqueada desde el teléfono móvil, en vez de utilizar la llave física.



*Figura 8. Diseño realizado por el autor (WiFi)*

El protocolo WiFi es ampliamente utilizado en aplicaciones IoT debido a la capacidad de cubrir distancias más largas que otros protocolos como puede ser Bluetooth. Este tiene la capacidad de transmitir un gran volumen de datos a alta velocidad. Es altamente utilizado para aquellas aplicaciones donde un gran ancho de banda es requerido.

Con respecto a las capas del modelo OSI, el protocolo WiFi opera en las siguientes capas:

1. Capa física: El protocolo WiFi transmite ondas de radio en bandas de frecuencia de 2.4 GHz, 5 GHz y 6 GHz, donde la elección depende de la saturación de la red, la distancia y la velocidad [10].
2. Capa de enlace: El protocolo MAC es utilizado para sincronizar los dispositivos y controlar errores. Este asegura que solo un dispositivo transmita en un momento dado, evitando colisiones.
3. Capa de red: Es importante señalar que el protocolo WiFi sí que utiliza direccionamiento IP. Esto significa que cuando los dispositivos IoT tienen capacidades WiFi, pueden acceder a la red comunicándose directamente con el router o el servidor a través de internet, no necesitando un gateway.

Se ha decidido no utilizar este protocolo debido a:

1. Consumo de energía: Como se ha mencionado en este proyecto, el consumo de energía es uno de los requisitos cruciales. Se tienen distribuidos un gran número de sensores a lo largo de un área de 500 hectáreas. Eso significa que un constante mantenimiento de estos dispositivos llevaría a un gran costo a lo largo del tiempo.
2. Aunque este protocolo presenta una gran mejoría del alcance máximo con respecto al protocolo bluetooth, este alcance sigue sin ser válido para los requisitos de este proyecto. Este protocolo tiene un alcance máximo no superior a los 90 metros en exteriores.
3. Cada punto de acceso puede conectar un máximo de 250 dispositivos. Sin embargo, en la práctica, la mayoría de soluciones permiten la comunicación simultánea de

hasta 50 dispositivos debido a limitaciones de ancho de banda. Esto significa que, para conectar 1000 sensores, se requerirían un mínimo de 4 gateways e incluso puede variar hasta 20 gateways [11].

Cabe mencionar que este protocolo es altamente utilizado en aquellas aplicaciones donde el consumo de energía no sea un factor crítico y un gran ancho de banda sea requerido. Un ejemplo de ello es, una cámara de seguridad, que necesita transmitir un gran volumen de datos en tiempo real.



*Figura 9. Diseño realizado por el autor (ZigBee)*

El protocolo Zigbee es un protocolo ideal para dispositivos IoT, ofreciendo la capacidad de formar topología de redes flexibles (como Malla, Estrella o Árbol), lo que lo hace adecuado para aquellas aplicaciones donde se necesiten conectar un gran número de dispositivos. El uso de estas topologías, hace que aumente la fiabilidad de la transmisión de información debido a que se tienen múltiples caminos de transmisión.

Zigbee cuenta con un bajo consumo energético en redes simples y de corto alcance. Sin embargo, en configuraciones como puede ser Malla, el consumo puede aumentar

significativamente, ya que los nodos actúan como repetidores, recibiendo y retransmitiendo datos de otros dispositivos.

Con respecto a las capas del modelo OSI,

1. Capa física: Zigbee opera en las bandas de frecuencia de 868 MHz, 915 MHz y 2.4 GHz. Cabe señalar que Zigbee utiliza O-QPSK cuando opera en la banda de frecuencia de 2.4 GHz, modulación eficiente en términos de energía. En las bandas de 868MHz y 915 MHz utiliza la modulación BPSK [12].
2. Capa de enlace: Se basa en el protocolo MAC para la gestión de comunicación entre dispositivos, ofreciendo topologías de red como la mencionada Malla, Estrella y Árbol [13].
3. Capa de red: Zigbee no utiliza direccionamiento IP, por lo que sería necesario el uso del gateway.

Este protocolo no ha sido seleccionado para este proyecto debido a:

1. Alcance máximo de hasta 100 metros, lo que por lo tanto no cumple con los requisitos del proyecto.
2. Baja velocidad de transmisión.
3. Elevado consumo energético en redes complejas, debido a que cada nodo recibe y transmite datos de otros dispositivos, lo que incrementa el uso de energía.



*Figura 10. Diseño realizado por el autor (Sigfox)*

Hasta el momento, todos los posibles protocolos a elegir han sido descartados debido al alcance limitado que presentan. En este sentido, el protocolo Sigfox puede parecer una alternativa interesante, ya que está diseñado específicamente para dispositivos IoT que requieren comunicaciones de largo alcance y un bajo consumo energético.

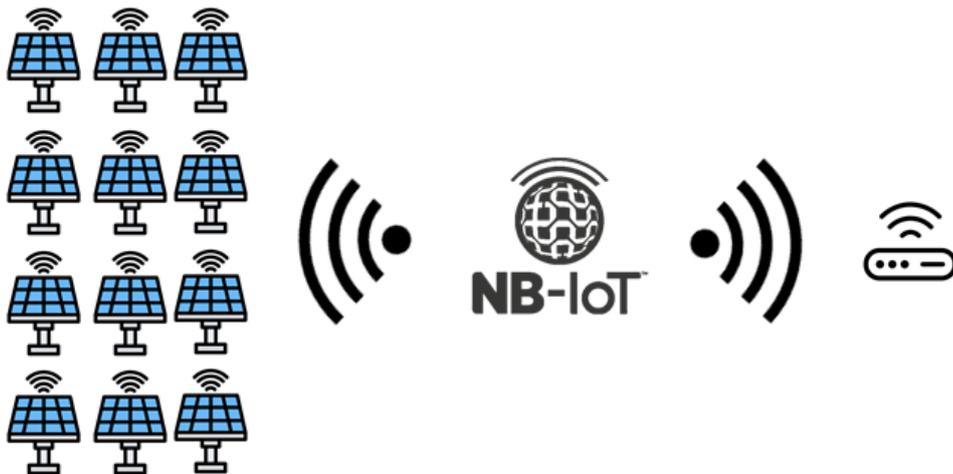
Desde el punto de vista del modelo OSI:

1. Capa física: Sigfox utiliza una modulación de espectro ultraestreccho (Ultra Narrow Band). Opera en bandas de frecuencia libres de licencia, como 868 MHz en Europa y 915 MHz en América [14].
2. Capa de enlace: Sigfox no implementa un protocolo MAC, sino que utiliza el protocolo ALOHA puro, en el que los dispositivos transmiten sin coordinación previa. Esto puede conllevar colisiones si hay una gran cantidad de dispositivos transmitiendo.
3. Capa de red: No utiliza direccionamiento IP, por lo que los dispositivos no son directamente accesibles a la red. Debido a la arquitectura que este protocolo utiliza, los dispositivos IoT envían datos directamente a las estaciones base de Sigfox, por lo tanto, no requiere hacer uso de gateways. Para que haya comunicación, es

importante que se tenga cobertura Sigfox para así poder transmitir la información a la nube [15].

Se ha decidido no continuar con este protocolo debido a:

1. Principalmente a la dependencia de la cobertura. Como se ha mencionado, Sigfox requiere que exista cobertura proporcionada por sus operadores externos, lo cual puede no ser el caso en la mayor parte de áreas rurales.
2. Se tiene un coste de suscripción por dispositivo, y presenta una limitación significativa en la cantidad de mensajes que se pueden transmitir al día.



*Figura 11. Diseño realizado por autor (NB-IoT)*

El protocolo NB-IoT (Narrowband IoT), utiliza las bandas LTE con licencia de operadores móviles. Este ofrece buena cobertura en interiores y áreas urbanas, un bajo consumo energético con una gran calidad de servicio.

Analizándolo desde el punto de vista del modelo OSI,

1. Capa Física: Opera en las bandas LTE, cuya frecuencia varía según la región. En Europa se suele utilizar la banda de 800 MHz o 900 MHz. Cabe indicar que la comunicación puede ser tanto ascendente (dispositivo a estación base) o trascendente (estación base a dispositivo) utilizando una modulación OFDMA y SC-FDMA respectivamente.
2. Capa de enlace: Hace uso del protocolo MAC de LTE, optimizado para la transmisión de pequeños volúmenes de datos, con un uso eficiente de energía.
3. Capa de red: Este protocolo utiliza direccionamiento IP, lo que permite comunicación directa con la red y servidores desde los dispositivos IoT, sin la necesidad de implementar un Gateway en su diseño [16].

Se ha decidido no utilizar este protocolo debido a:

1. Consumo Energético: Aunque este protocolo está optimizado para dispositivos IoT, el uso de bandas LTE implica una mayor complejidad y consumo energético comparado con otras tecnologías como LoRaWAN.
2. Cobertura: La cobertura depende exclusivamente de los operadores móviles, por lo que la señal LTE puede presentar limitaciones en zonas rurales, donde la señal LTE no esté garantizada [17].



*Figura 12. Diseño realizado por autor (LoRaWAN)*

El protocolo LoRaWAN (Long Range Wide Area Network) será el protocolo de red utilizado en este proyecto, como se ha definido en la sección Descripción de las Tecnologías. Está diseñado para aplicaciones IoT donde la comunicación a larga distancia, bajo consumo energético y bajo ancho de bandas son requisitos cruciales. Este protocolo es ideal para escenarios donde los dispositivos están distribuidos en grandes áreas geográficas.

Desde la perspectiva del modelo OSI:

1. Capa física: LoRaWAN utiliza la modulación LoRa, basadas en técnicas de espectro ensanchado por chirp (CSS), diseñada con el objetivo de tener una baja transferencia de datos, manteniendo así un consumo bajo y una alta cobertura.

Modulation / Spreading

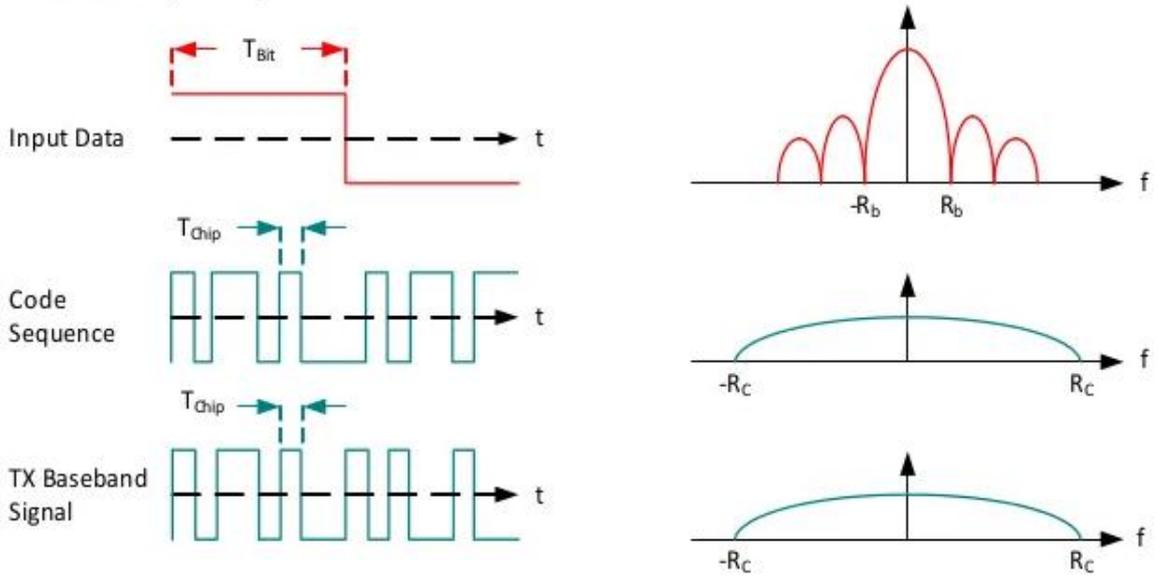


Figura 13. Imagen extraída de artículo [20].

## Demodulación

### Demodulation / De-spreading

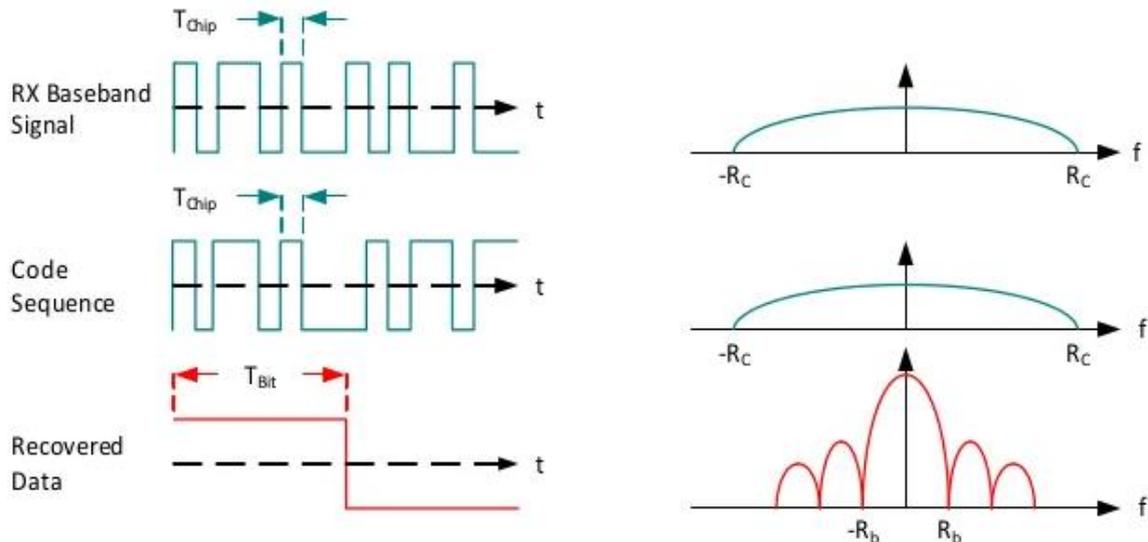


Figura 14. Imagen extraída de artículo [20].

Este protocolo, opera en bandas de frecuencia sin licencia, esto facilita aplicar este proyecto a lo largo de numerosos países sin la necesidad de obtener permisos especiales. A continuación, se muestra una tabla de las bandas de frecuencia sin licencia en distintas localizaciones.

| Región                       | Banda MHz |
|------------------------------|-----------|
| Asia                         | 433       |
| Europa, India, Rusia, Africa | 863-870   |
| Estados Unidos               | 902-928   |
| Australia                    | 915-928   |
| Canada                       | 779-787   |
| China                        | 470-510   |

Figura 15. Imagen extraída de artículo [21]

Este protocolo permite alcanzar distancias de hasta 20km en zonas rurales, lo cual es muy adecuado para este proyecto donde se están distribuyendo 1000 sensores a lo

largo de 500 hectáreas, reduciendo significativamente el número de gateways necesarios, disminuyendo así la complejidad y el coste del despliegue de la red.

2. Capa de Enlace: LoRaWAN implementa su propio protocolo MAC basado en Aloha puro, lo que significa que los dispositivos finales (IoT), pueden transmitir sin establecer una conexión previa con un Gateway específico. En su lugar, los mensajes son enviados a todos los gateways dentro de alcance, estos reenviándolos al servidor de red. En el caso de que el servidor de red reciba alguna copia duplicada, se quedará únicamente con una copia [18].

Cada dispositivo LoRaWAN opera en una de las siguientes clases según sus necesidades de comunicación.

- Clase A: Los sensores de aceleración, que muestrean a 20 Hz solo envían datos cuando se supera un umbral predefinido. Tras cada transmisión ascendente, se abren dos ventanas de recepción cortas, para posibles comandos descendentes (del servidor a el dispositivo IoT). Esta clase es la que menos energía consume.
  - Clase B: Utilizada para dispositivos que requieren comandos descendentes sincronizados. Abre ventana de recepción periódicas, requiriendo un consumo energético intermedio.
  - Clase C: Mantiene la ventana de recepción abierta continuamente, salvo durante transmisiones, pero consume más energía y suele requerir alimentaciones de la red [19].
3. Capa de Red: La red de LoRaWAN, típicamente utiliza una arquitectura, donde los gateways recogen los datos, y los envían a el servidor de LoRaWAN a través de internet. Este protocolo no utiliza direccionamiento IP, lo que simplifica el diseño de los dispositivos y reduce su complejidad.

Este protocolo ha sido seleccionado para este proyecto por las siguientes razones:

1. Alcance: Su capacidad para alcanzar comunicaciones de hasta 20km en entornos rurales lo convierte en una solución ideal para cubrir 500 hectáreas, facilitando el monitoreo distribuido con una infraestructura mínima.
2. Bajo Consumo Energético: Los sensores alimentados por las baterías, operando en Clase A, permanecen en bajo consumo la mayor parte del tiempo, enviando datos únicamente cuando la aceleración supere el umbral. Esto extiende la vida de las baterías, sin necesidad de un reemplazo frecuente, lo que lo convierte en sostenible y económico a largo plazo.
3. Coste: Como se ha mencionado, LoRaWAN utiliza bandas de frecuencia sin licencia, eliminando la necesidad de pagar tarifas, minimizando así los costes de implementación.
4. Escalabilidad: La arquitectura de este sistema, permite conectar una gran cantidad de sensores con pocos Gateways, facilitando la expansión del sistema sin necesidad de modificar la infraestructura diseñada.

Tras finalizar el análisis del protocolo de comunicación entre los dispositivos IoT y el Gateway, se abordará el segundo análisis, enfocado en la transmisión de datos entre el servidor de aplicación y los dispositivos finales.

Antes de entrar en profundidad con los posibles protocolos a utilizar entre el servidor y los dispositivos finales, es importante entender las bases de transporte sobre los que estos protocolos se apoyan, como son TCP o UDP. Cabe comentar que estos protocolos de transporte no son los únicos existentes, pero sí los más ampliamente utilizados y relevantes para este proyecto [17].

#### TCP (Transmission Control Protocol)

- Es un protocolo orientado a conexión, lo que significa que realiza una conexión previa a la transmisión de datos manteniendo la conexión hasta que se complete la comunicación. Esta conexión previa garantiza la entrega correcta de datos, la capacidad de corrección de errores y evita saturaciones de la red. Esta alta fiabilidad se traduce negativamente en una mayor latencia y consumo de recursos.

## UDP (User Datagram Protocol)

-UDP es un protocolo no orientado a conexión, donde se prioriza la velocidad de transmisión y la baja latencia sobre la precisión de información. No garantiza la entrega ni el orden de los paquetes, pero su simplicidad hace adecuado para aplicaciones en tiempo real como es una videollamada o un videojuego en línea.

A nivel de estructura, la cabecera TCP es más compleja, ocupando un mínimo 20 bytes, mientras que la cabecera UDP es mucho más ligera ocupando únicamente 8 bytes, reduciendo así el tamaño del paquete y ofreciendo una velocidad de transmisión más alta.

Cada uno de estos protocolos de comunicación añade su propia cabecera sobre la capa de transporte correspondiente (TCP o UDP). Finalmente, el paquete transmitido estará compuesto por la cabecera del protocolo de transporte (TCP O UDP), la cabecera del protocolo de comunicación (MQTT o cualquier otro seleccionado) y finalmente el mensaje o los datos específicos que se desean comunicar.



*Figura 16. Diseño realizado por autor (AMQP)*

El primer protocolo a analizar dentro de este análisis, será el protocolo AMQP (Advanced Message Queuing Protocol), se traduce en español como Protocolo Avanzado de Cola de Mensajes. Fue diseñado para ser un protocolo de mensajería empresarial, generalmente implementado sobre TCP. Permite la comunicación cifrada y confiable entre aplicaciones y organizaciones, especialmente diseñado para garantizar la entrega de mensajes a pesar de las condiciones de la red.

Este protocolo, hace uso de un bróker, el cual será el intermediario encargado de enrutar y distribuir los mensajes de manera fiable.

Se ha decidido no utilizar este protocolo debido a:

1. Complejidad: Este protocolo ofrece mecanismos avanzados como las colas, el cual no serán necesarios en nuestro sistema. Esta complejidad de implementación es innecesaria, pudiendo alcanzar los mismos objetivos mediante otros protocolos más ligeros y simples.
2. Consumo de energía: Cada mensaje AMQP lleva una cabecera de mínimo de 8 bytes, (MQTT lleva una de mínimo 2 bytes). Esta diferencia, aunque aparentemente sea pequeña, se traduce en un mayor volumen de datos transmitidos afectando negativamente al consumo energético de los dispositivos finales [20].



*Figura 17. Diseño realizado por autor (HTTP)*

El siguiente protocolo considerado para el análisis debido a sus amplias capacidades es HTTP (Hypertext Transfer Protocol), ampliamente utilizado para la transferencia de información entre servidores y clientes.

Opera bajo un modelo cliente-servidor, basado en peticiones y respuestas, lo que lo hace ideal para la navegación en servicios web tradicionales y APIs.

HTTP resulta una opción válida para la entrega de información a dispositivos finales como PCs o móviles, ya que estos soportan el protocolo de forma nativa. Sin embargo, se ha decidido no utilizar HTTP como protocolo de aplicación debido a:

1. HTTP requiere una cabecera mínima de 26 bytes por mensaje (2 bytes en MQTT), donde, además, cada transmisión de información establece y cierra una conexión TCP, implicando así una mayor carga de red. Esto se transmite en un mayor uso de ancho de banda y un consumo energético considerablemente más alto, especialmente en conexiones móviles.
2. La falta de inmediatez del protocolo HTTP es otra de las limitaciones que este presenta. Este protocolo obliga a los dispositivos finales a realizar sondeos periódicos, verificando si hay nueva información disponible, generando latencias y transmisiones innecesarias.
3. El protocolo HTTP también presenta limitaciones en entornos distribuidos, donde múltiples dispositivos reciben actualizaciones simultáneamente. HTTP requiere que cada uno de estos, solicite la información por separado. En cambio, el modelo publicación-suscripción permite una distribución más eficiente, facilitando la escalabilidad del sistema [21] [22].



*Figura 18. Diseño realizado por autor (MQTT)*

El último protocolo considerado para este análisis es el protocolo MQTT (Message Queue Telemetry Transport), protocolo basado en el modelo de publicación-suscripción. Diseñado para facilitar la comunicación eficiente entre dispositivos en redes con recursos limitados o con conectividad intermitente.

Como se explicó en el capítulo 2 de este documento (Descripción de las Tecnologías), MQTT opera mediante un bróker que gestiona la recepción y entrega de mensajes de aquellos clientes que estén suscritos a determinados temas.

Este protocolo mantiene una conexión persistente basada en TCP, garantizando la entrega correcta de datos junto a la capacidad de corrección de errores, importante para la aplicación de este proyecto después de todo el proceso.

El protocolo MQTT es elegido el más adecuado para la comunicación entre el servidor de red y los dispositivos finales como móviles y PCs debido a sus siguientes características:

1. MQTT utiliza cabeceras muy ligeras, con un tamaño mínimo de 2 bytes por mensaje. Por otro lado, mantiene la conexión TCP abierta, evitando así el sobrecoste que supone abrir y cerrar la conexión cada vez que se transmita información. Esto conlleva a un menor uso de ancho de banda y un menor consumo energético [21].
2. Gracias al modelo que este protocolo utiliza, la información será enviada a los dispositivos finales suscritos tan pronto como esta llega al servidor, evitando así la necesidad de realizar sondeos periódicos, lo que se traduce en una mejor latencia y experiencia de usuario.

3. Como se ha mencionado anteriormente, HTTP presentaba grandes dificultades con respecto a la transmisión de información en entornos distribuidos. MQTT permite la distribución eficiente mediante el bróker, facilitando la escalabilidad del sistema cuando varios clientes necesitan acceder a la información simultáneamente [23].
4. Otra característica que hace más adecuado MQTT es la flexibilidad que este ofrece, pudiendo seleccionar tres niveles QoS (calidad de servicio), ajustando así la fiabilidad de la transmisión de información dependiendo de la necesidad del sistema, algo que por ejemplo el protocolo HTTP no ofrece [24].
5. MQTT también presenta funcionalidades adicionales como la retención de mensajes para nuevos suscriptores, de manera que el bróker almacena el último mensaje publicado en cada tema, y lo entrega automáticamente a cualquier nuevo cliente que se suscriba [25].

Gracias a su arquitectura ligera, flexibilidad, bajo consumo energético, eficiencia en el uso del ancho de banda y escalabilidad, MQTT ha sido seleccionado como el protocolo de aplicación más adecuado para este proyecto.

## Capítulo 4. SIMULACIONES Y VALIDACIÓN DEL SISTEMA

Este capítulo tiene como objetivo demostrar la viabilidad del sistema propuesto a través de simulaciones virtuales y pruebas reales. Se busca validar tanto el comportamiento teórico definido previamente como su implementación física, con el fin de comprobar que los objetivos y requisitos definidos previamente se pueden alcanzar en un entorno realista.

### *4.1 SIMULACIÓN DEL PROTOCOLO LORAWAN*

Se ha utilizado el simulador LoRaWAM-SIM, una herramienta de código abierto diseñada para evaluar el rendimiento de redes LoRaWAN en escenarios realistas. Aunque este simulador no utiliza la modulación CSS (Chirp Spread Spectrum) a nivel físico, sí que implementa con precisión los efectos prácticos del protocolo LoRaWAN.

Este simulador, cuenta con muchas características que hacen esta simulación única, algunas de estas características son:

- Múltiples gateways half-duplex
- Restricción del 1% de duty cycle en transmisiones uplink de acuerdo con las normativas europeas (como EU868), lo que es esencial para evaluar la escalabilidad.
- Ventanas de recepción RX1 y RX2 para confirmaciones ACKs
- Manejo de colisiones
- Cálculo del consumo energético por nodo
- Modelo de atenuación por pérdida de trayectoria
- ADR (Adaptive Data Rate), lo que ajusta automáticamente la potencia de transmisión
- Tamaño y tasa de paquetes configurables

Este simulador ha sido utilizado en diferentes investigaciones, lo que demuestra una vez más, su fiabilidad y capacidad para modelar escenarios reales. Entre dichas investigaciones, se incluye la optimización de recolección de datos, políticas de transmisión y la coexistencia

de redes LoRaWAN en la banda EU868. Además, distintas organizaciones han hecho uso de este simulador en sus propios análisis, como PV Hardware y Kazakh Railways. [26]-[28]

Para utilizar este simulador será necesario entender los siguientes dos comandos a ejecutar:

1. `perl generate_terrain.pl terrain_side_size_(m) num_of_nodes num_of_gateways > terrain.txt`

Este primer comando, generará el archivo `terrain.txt` que contiene la descripción del tamaño del terreno, número de nodos, número de Gateways y la posición de cada uno de estos nodos y gateways.

2. `perl LoRaWAN.pl packets_per_hour simulation_time_(hours) terrain.txt`

Aquí se incluye el número de paquetes a transmitir por hora y el número de horas que se pretende ejecutar la simulación. Cabe destacar que la simulación no requiere esperar el número de horas seleccionadas, ya que procesa los eventos de forma rápida.

A continuación, se muestra paso a paso el proceso:

1. Descargar el repositorio y las librerías necesarias.

<https://github.com/deltazita/LoRaWAN-SIM>

2. Una vez instalado, nos dirigimos a la carpeta del repositorio

3. Se ejecuta el primer comando con los parámetros deseados.

```
VMCarlos@VMCarlos:~/LoRaWAN-SIM$ perl generate_terrain.pl 2236 1000 5 > prueba_tfg.txt
VMCarlos@VMCarlos:~/LoRaWAN-SIM$
```

En nuestro caso, nuestro proyecto está pensado para una distribución de 1000 sensores (1 por cada placa fotovoltaica) a lo largo de 500 hectáreas. Como 1 hectárea son 10.000  $m^2$ , 500 hectáreas serán 5000000  $m^2$ . Por lo que cada lado será  $\sqrt{5000000} = 2236.06 m$ . Es por ello que el primer parámetro es 2236. El segundo y tercer parámetro

indican que se quiere realizar la simulación con 1000 sensores y 5 gateways respectivamente.

4. Ejecutar el segundo comando:

```
VMCarlos@VMCarlos:~/LoRaWAN-SIM$ perl LoRaWAN.pl 12 10 prueba_tfg.txt
```

En este caso, se hará la simulación transmitiendo 12 paquetes por hora, en un total de 10 horas.

5. A continuación, se muestra los detalles de la simulación:

```
VMCarlos@VMCarlos:~/LoRaWAN-SIM$ perl LoRaWAN.pl 12 10 prueba_tfg.txt
Simulation time = 35999.124 secs
Avg node consumption = 33.47776 J
Min node consumption = 26.81585 J
Max node consumption = 101.42797 J
Total number of transmissions = 119747
Total number of unique transmissions = 119747
Stdv of unique transmissions = 0.43
Total packets received = 119374
Total unique packets acknowledged = 0
Total confirmed packets dropped = 0
Total unconfirmed packets dropped = 373
Packet Delivery Ratio = 0.99689
Packet Reception Ratio = 0.99689
Unconfirmed uplink fairness = 0.007
-----
# of nodes with SF7: 550, Avg retransmissions: 0.00
# of nodes with SF8: 151, Avg retransmissions: 0.00
# of nodes with SF9: 149, Avg retransmissions: 0.00
# of nodes with SF10: 97, Avg retransmissions: 0.00
# of nodes with SF11: 48, Avg retransmissions: 0.00
# of nodes with SF12: 5, Avg retransmissions: 0.00
Avg SF = 7.957
Script execution time = 6.1689 secs
```

Donde se puede observar información muy interesante a tener en cuenta, como es el consumo medio energético por nodo, el número total de transmisiones y la tasa de

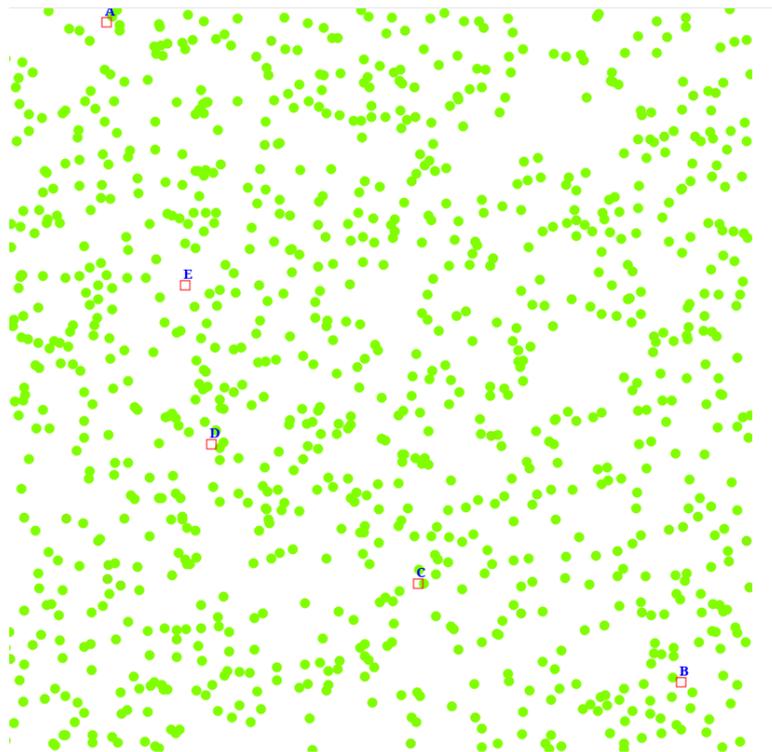
paquetes recibidos correctamente. Estos parámetros serán analizados en mayor profundidad más adelante.

Una de las funcionalidades que este simulador ofrece es la posibilidad de visualizar gráficamente el entorno desarrollado según los parámetros introducidos. A la hora de ejecutar el segundo comando, el simulador crea por defecto un archivo .svg.

Para visualizar la plantilla generada se puede utilizar el siguiente comando:

```
VMCarlos@VMCarlos:~/LoRaWAN-SIM$ firefox prueba_tfg.txt-img.svg
```

Acto y seguido, se abrirá la imagen en el navegador elegido, en este caso Firefox.



En esta imagen, cada círculo verde representa un nodo LoRaWAN, y cada cuadrado rojo representa un Gateway.

Es importante mencionar que este simulador coloca aleatoriamente tanto los nodos LoRaWAN como gateways. Es por ello que, para tener una simulación todavía más aproximada a nuestro proyecto, se han modificado concretamente el archivo generate-terrain.pl simulando así nuestro proyecto con nodos y sensores distribuidos en filas y columnas.

Como se mencionó anteriormente el primer comando a ejecutar, es el encargado de generar el terreno con la dimensión, número de nodos y gateways y sus posiciones en el terreno.

El archivo generate-terrain.pl genera la posición de los nodos y Gateways en la siguiente parte del código:

```
my @sensors;  
my @gws;  
my %coords;  
  
for(my $i=1; $i<=$nodes; $i++){  
    my ($x, $y) = (int(rand($tx*10)), int(rand($tx*10)));  
    ($x, $y) = ($x/10, $y/10);  
    while (exists $coords{$x}{$y}){  
        ($x, $y) = (int(rand($tx*10)), int(rand($tx*10)));  
        ($x, $y) = ($x/10, $y/10);  
    }  
    $coords{$x}{$y} = 1;  
    push(@sensors, [$x, $y]);  
}  
for(my $i=1; $i<=$gws; $i++){  
    my ($x, $y) = (int(rand($tx*10)), int(rand($tx*10)));  
    ($x, $y) = ($x/10, $y/10);  
    while (exists $coords{$x}{$y}){  
        ($x, $y) = (int(rand($tx*10)), int(rand($tx*10)));  
        ($x, $y) = ($x/10, $y/10);  
    }  
    $coords{$x}{$y} = 1;  
    push(@gws, [$x, $y]);  
}
```

Se ha modificado esa parte del código, de manera que la posición de los nodos sea fija:

```
# 40 columnas x 25 filas = 1000 (nodos)
my $cols = 40;
my $rows = 25;

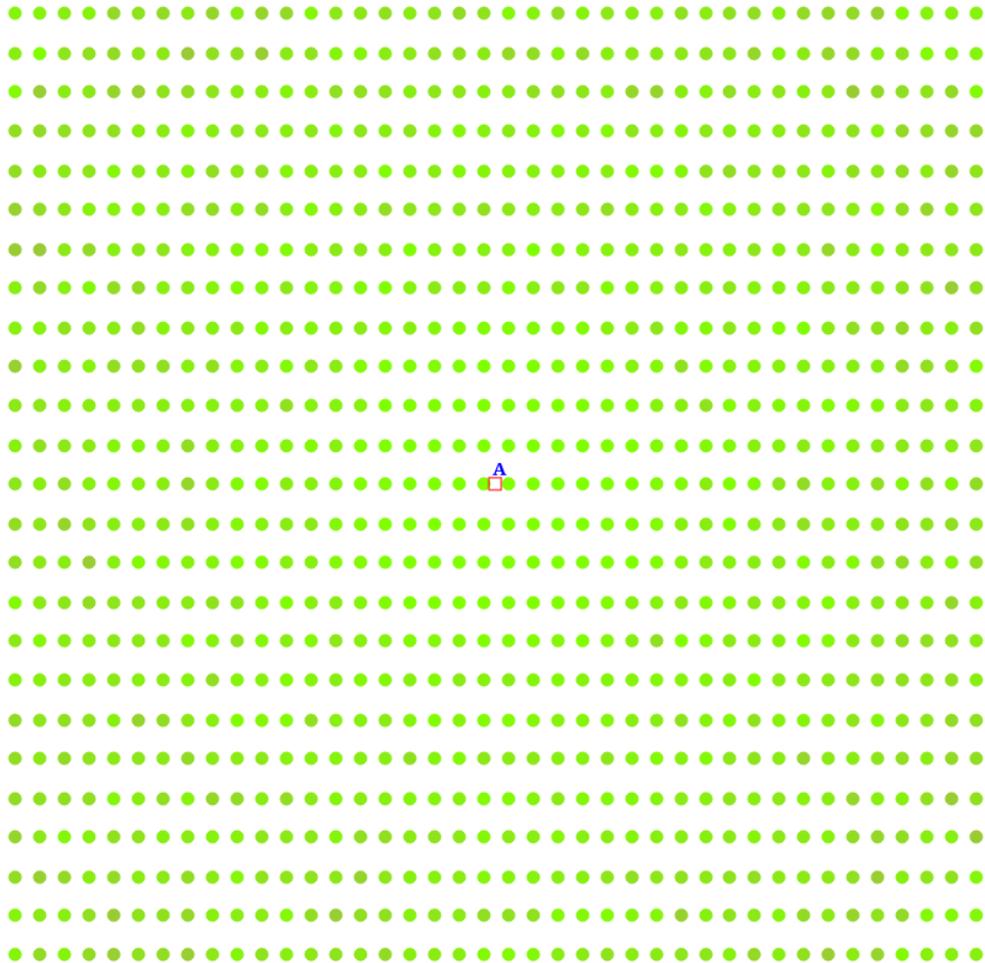
# Espaciado entre sensores
my $dx = $tx / $cols;
my $dy = $ty / $rows;

my @sensors;
my $id = 1;
for my $row (0..$rows-1) {
    for my $col (0..$cols-1) {
        my $x = ($col + 0.5) * $dx;
        my $y = ($row + 0.5) * $dy;
        push @sensors, [$id++, $x, $y];
    }
}
```

Lo mismo para los gateways:

```
# Un único gateway situado en el centro del terreno
my @gws = (
    ['A', $tx/2, $ty/2],
);
```

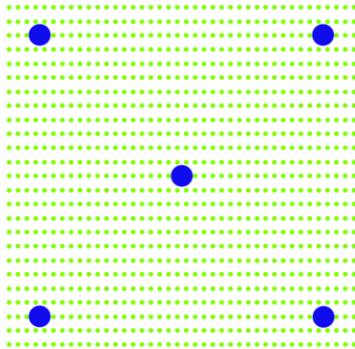
El resultado de esta modificación en el archivo encargado de generar el diseño es el siguiente:



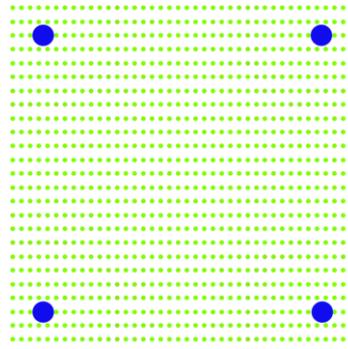
Esto empieza a cobrar sentido, donde solo queda determinar el número de gateways a utilizar en este proyecto. Para analizar el comportamiento de la red en función del número de gateways, se han definido 5 configuraciones distintas, donde cada una de ellas mantiene los 1000 sensores distribuidos en 25 filas y 40 columnas, en un área de 500 hectáreas.

Cada configuración varía únicamente en el número y disposición de los gateways, permitiendo así analizar cómo estos varían el rendimiento general de la red. Se compararán parámetros clave como la tasa de entrega de paquetes y el consumo medio energético por nodo. El análisis se hará de manera progresiva, comenzando con 5 gateways y reduciendo su número hasta llegar a un solo gateway.

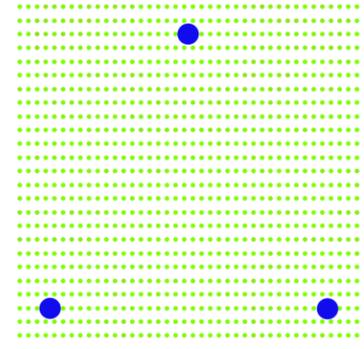
A continuación, se muestran las 5 configuraciones, donde cada círculo azul representa un gateway:



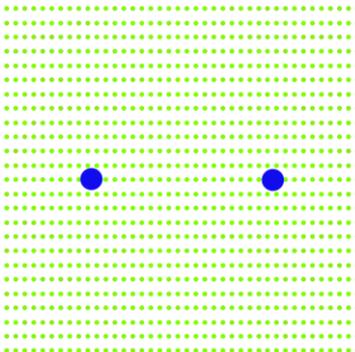
5 gateways



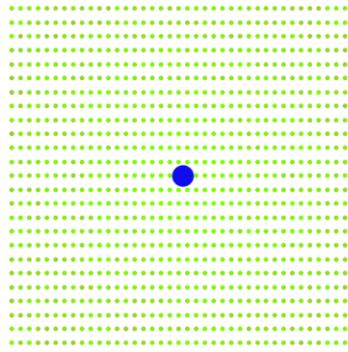
4 gateways



3 gateways



2 gateways



1 gateway

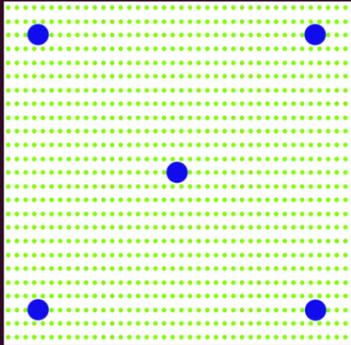
Una vez definidas las configuraciones, comienza la simulación. Se ha seleccionado transmitir 12 paquetes por hora por nodo, con el objetivo de simular un escenario realista típico en aplicaciones IoT basadas en LoRaWAN, donde los dispositivos transmiten con una baja frecuencia con el objetivo de conservar energía.

El tiempo real de simulación se ha fijado en 10 horas. Esto es debido a que es un parámetro suficiente para medir el rendimiento de la red. Si cada uno de los nodos envía 12 paquetes por hora, se tienen 10 horas de simulación y 1000 nodos. Habrá un total de  $12 * 10 * 1000 =$

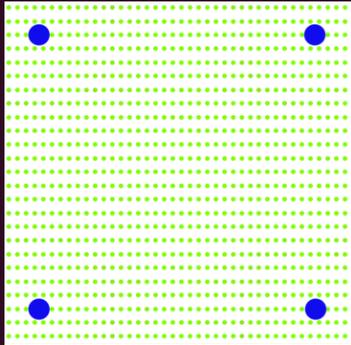
120.000 transmisiones, por lo que aumentar el número de horas, únicamente multiplica el número de datos sin necesariamente mejorar las conclusiones.

Tras ejecutar los comandos de simulación para cada una de las 5 configuraciones empezando de más a menos gateways se obtienen los siguientes resultados:

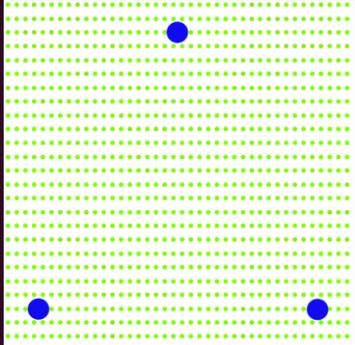
```
VMCarlos@VMCarlos:~/LoRaWAN-SIM$ perl LoRaWAN.pl 12 10 terrain_5gateways.txt
Simulation time = 36000.077 secs
Avg node consumption = 29.94587 J
Min node consumption = 26.41424 J
Max node consumption = 35.76260 J
Total number of transmissions = 119753
Total number of unique transmissions = 119753
Stdv of unique transmissions = 0.43
Total packets received = 118916
Total unique packets acknowledged = 0
Total confirmed packets dropped = 0
Total unconfirmed packets dropped = 837
Packet Delivery Ratio = 0.99301
Packet Reception Ratio = 0.99301
Unconfirmed uplink fairness = 0.012
-----
# of nodes with SF7: 550, Avg retransmissions: 0.00
# of nodes with SF8: 308, Avg retransmissions: 0.00
# of nodes with SF9: 142, Avg retransmissions: 0.00
Avg SF = 7.592
Script execution time = 8.2422 secs
```



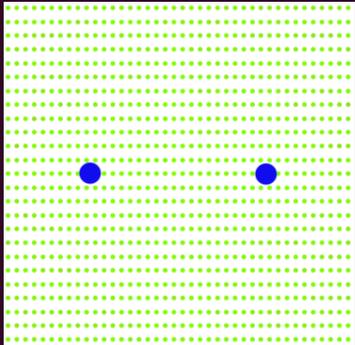
```
VMCarlos@VMCarlos:~/LoRaWAN-SIM$ perl LoRaWAN.pl 12 10 terrain_4gateways.txt
Simulation time = 35999.976 secs
Avg node consumption = 32.06694 J
Min node consumption = 25.54092 J
Max node consumption = 44.52144 J
Total number of transmissions = 119749
Total number of unique transmissions = 119749
Stdv of unique transmissions = 0.43
Total packets received = 118233
Total unique packets acknowledged = 0
Total confirmed packets dropped = 0
Total unconfirmed packets dropped = 1516
Packet Delivery Ratio = 0.98734
Packet Reception Ratio = 0.98734
Unconfirmed uplink fairness = 0.017
-----
# of nodes with SF7: 384, Avg retransmissions: 0.00
# of nodes with SF8: 216, Avg retransmissions: 0.00
# of nodes with SF9: 332, Avg retransmissions: 0.00
# of nodes with SF10: 68, Avg retransmissions: 0.00
Avg SF = 8.084
Script execution time = 7.4295 secs
```



```
VMCarlos@VMCarlos:~/LoRaWAN-SIM$ perl LoRaWAN.pl 12 10 terrain_3gateways.txt
Simulation time = 36000.390 secs
Avg node consumption = 32.86599 J
Min node consumption = 25.11594 J
Max node consumption = 44.52144 J
Total number of transmissions = 119731
Total number of unique transmissions = 119731
Stdv of unique transmissions = 0.44
Total packets received = 117445
Total unique packets acknowledged = 0
Total confirmed packets dropped = 0
Total unconfirmed packets dropped = 2286
Packet Delivery Ratio = 0.98091
Packet Reception Ratio = 0.98091
Unconfirmed uplink fairness = 0.021
-----
# of nodes with SF7: 320, Avg retransmissions: 0.00
# of nodes with SF8: 200, Avg retransmissions: 0.00
# of nodes with SF9: 364, Avg retransmissions: 0.00
# of nodes with SF10: 116, Avg retransmissions: 0.00
Avg SF = 8.276
Script execution time = 7.3821 secs
```

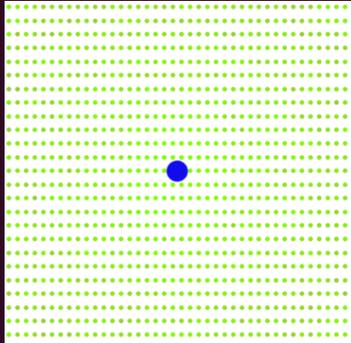


```
VMCarlos@VMCarlos:~/LoRaWAN-SIM$ perl LoRaWAN.pl 12 10 terrain_2gateways.txt
Simulation time = 36000.125 secs
Avg node consumption = 32.93487 J
Min node consumption = 24.90345 J
Max node consumption = 44.52144 J
Total number of transmissions = 119703
Total number of unique transmissions = 119703
Stdv of unique transmissions = 0.46
Total packets received = 116868
Total unique packets acknowledged = 0
Total confirmed packets dropped = 0
Total unconfirmed packets dropped = 2835
Packet Delivery Ratio = 0.97632
Packet Reception Ratio = 0.97632
Unconfirmed uplink fairness = 0.023
-----
# of nodes with SF7: 332, Avg retransmissions: 0.00
# of nodes with SF8: 220, Avg retransmissions: 0.00
# of nodes with SF9: 288, Avg retransmissions: 0.00
# of nodes with SF10: 160, Avg retransmissions: 0.00
Avg SF = 8.276
Script execution time = 6.7166 secs
```



```

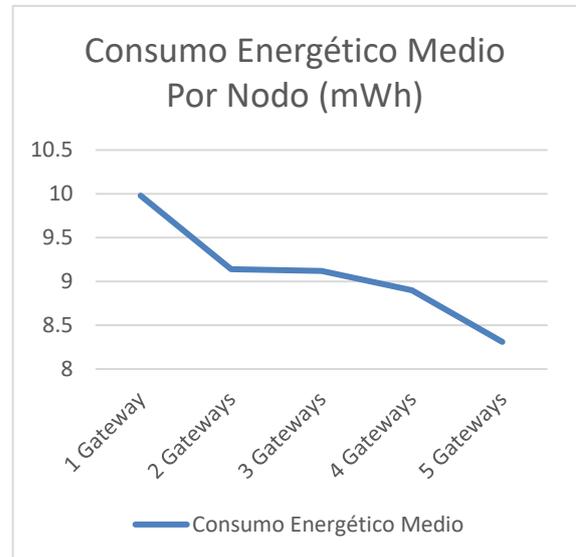
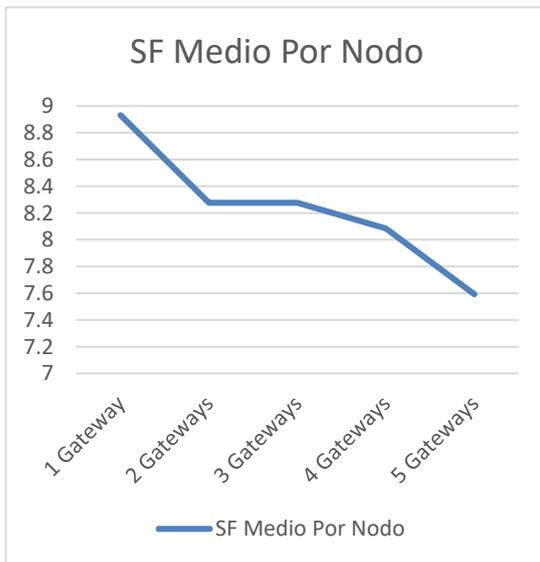
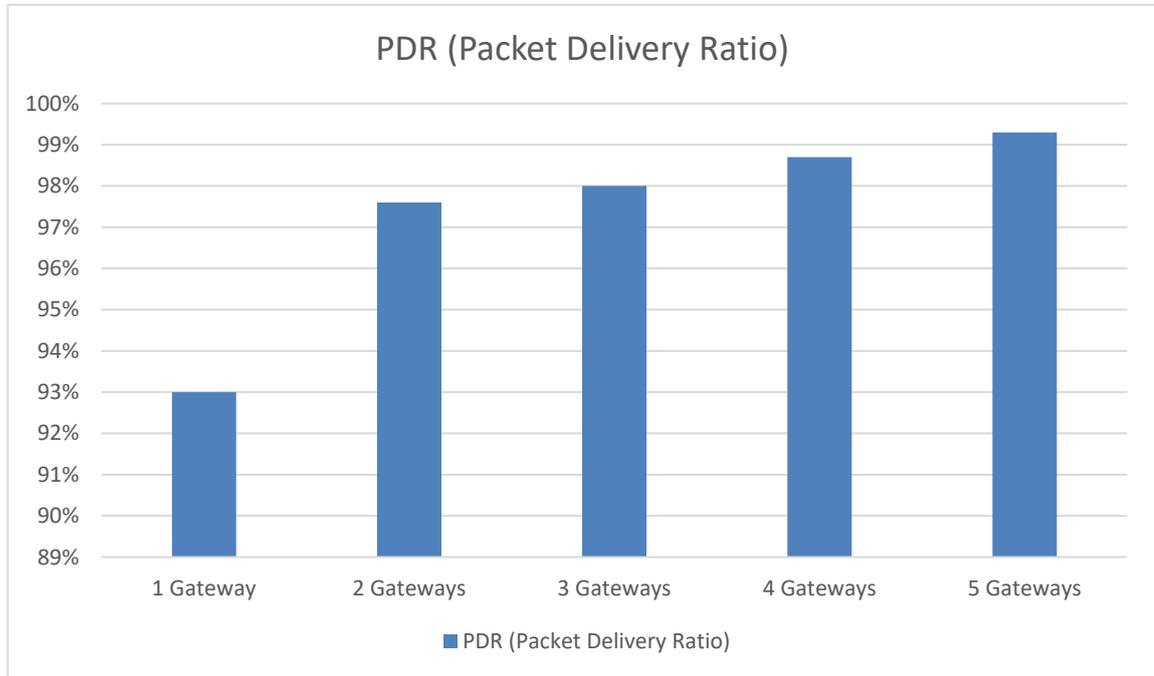
VMCarlos@VMCarlos:~/LoRaWAN-SIM$ perl LoRaWAN.pl 12 10 terrain_1gateway.txt
Simulation time = 36000.093 secs
Avg node consumption = 35.93658 J
Min node consumption = 24.90345 J
Max node consumption = 67.24011 J
Total number of transmissions = 119719
Total number of unique transmissions = 119719
Stdv of unique transmissions = 0.45
Total packets received = 111404
Total unique packets acknowledged = 0
Total confirmed packets dropped = 0
Total unconfirmed packets dropped = 8315
Packet Delivery Ratio = 0.93055
Packet Reception Ratio = 0.93055
Unconfirmed uplink fairness = 0.044
-----
# of nodes with SF7: 166, Avg retransmissions: 0.00
# of nodes with SF8: 148, Avg retransmissions: 0.00
# of nodes with SF9: 306, Avg retransmissions: 0.00
# of nodes with SF10: 348, Avg retransmissions: 0.00
# of nodes with SF11: 32, Avg retransmissions: 0.00
Avg SF = 8.932
Script execution time = 7.6313 secs
  
```



Se muestra a continuación una tabla con las características más relevantes para el análisis:

| Número de Gateways | Consumo energético medio por nodo | PDR (Tasa de entrega de paquetes) | SF (Spreading Factor) medio |
|--------------------|-----------------------------------|-----------------------------------|-----------------------------|
| 1                  | 9.98 mWh                          | 93 %                              | 8.932                       |
| 2                  | 9.14 mWh                          | 97.6 %                            | 8.276                       |
| 3                  | 9.12 mWh                          | 98 %                              | 8.276                       |
| 4                  | 8.9 mWh                           | 98.7 %                            | 8.084                       |
| 5                  | 8.31 mWh                          | 99.3 %                            | 7.592                       |

*Table 1. Comparativa con respecto a número de Gateways*



Se puede observar cómo a medida que se aumenta el número de gateways en la red, se produce una mejora general en el rendimiento del sistema:

- La tasa de entrega de paquetes (PDR) aumenta.
- El consumo energético medio por nodo disminuye.
- El Spreading Factor (SF) medio reduce.

Este comportamiento es coherente con la lógica de LoRaWAN, a mayor número de gateways, la distancia media entre los nodos y su gateway más cercano tiende a ser menor. Esto se traduce en el uso de una menor potencia y, por lo tanto, reducir el consumo energético.

Además, el Spreading Factor (SF) en LoRaWAN varía desde 7 hasta 12. Un SF más alto permite comunicaciones a mayor distancia a cambio de una transmisión más duradera y un mayor consumo energético. Por el contrario, un SF más bajo, implica comunicaciones más rápidas a cambio de menor alcance y un consumo de energía menor.

Este parámetro está totalmente relacionado con el objetivo de este proyecto, ya que al aumentar el número de gateways:

- Nodos pueden usar Spreading Factors más bajos.
- Se reduce el tiempo en el aire de cada transmisión.
- Mejora el rendimiento energético de la red.

Por otro lado, el PDR (Packet Delivery Ratio) es mayor a medida que el número de gateways aumenta, ya que hay más puntos de recepción, aumentando así la probabilidad de que los paquetes lleguen correctamente.

Cabe mencionar que el simulador asigna un SF concreto a cada nodo, en función de la distancia del nodo al gateway más cercano. Por ello se muestra el SF medio de todos los 1000 nodos, tras la asignación previa del simulador.

La decisión sobre el número de gateways a utilizar en este proyecto, es importante, ya que afecta directamente al rendimiento de la red como se ha mostrado previamente.

A primera vista, cuanto mayor sea el número de gateways, mejor será el rendimiento global de la red. Sin embargo, uno de los objetivos de este proyecto es buscar un sistema equilibrado, que combine un buen rendimiento junto a una viabilidad económica razonable. Es decir, un sistema que ofrezca parámetros suficientemente óptimos, pero sin incurrir en costes.

El análisis detallado de los datos, muestra como el salto de 1 a 2 gateways, mejora considerablemente el rendimiento, no obstante, el paso de 2 a 3 gateways no aporta mejoras significativas, lo cual podría indicar que hacer uso de 2 gateways puede ser la opción adecuada. Sin embargo, al introducir un cuarto Gateway, se vuelve a observar una mejora relevante, con una tasa de entrega del 98.7% y un SF medio de 8.084 y un menor consumo energético medio por nodo.

Desde la perspectiva de tolerancia a fallos, utilizar 2 gateways podría presentar un riesgo, ya que, si uno de ellos falla, el rendimiento podría degradarse bruscamente, con un nuevo PDR de 93%. Con 4 gateways en cambio, aparte de alcanzar un alto rendimiento, ofrece un margen de resiliencia.

En conclusión, se ha optado por hacer uso de 4 gateways al considerar que es la opción más equilibrada, eficiente y fiable, que además ofrece capacidad de escalabilidad y garantiza un alto rendimiento con una tolerancia adecuada ante posibles fallos.

Con respecto a la duración del sistema en términos energéticos, estos nodos suelen estar compuestos por placas ESP32 integradas con módulos LoRa, alimentadas normalmente mediante baterías externas aproximadamente de 3000 mAh. Según la simulación realizada, durante un periodo de 10 horas con una transmisión de 12 paquetes por hora (120 transmisiones en total), el consumo energético medio haciendo uso de 4 gateways, es de 8.9 mWh.

Para hacer un cálculo aproximado de la duración de estas baterías, transmitiendo por ejemplo 2 paquetes por hora, sería:

$$\text{Sabiendo que } E (Wh) = V(V) * \text{Capacidad (Ah)}$$

La energía total almacenada de una batería de 3000 mAh (3 Ah) a 3.3 V es de  $E = 3.3 V * 3 = 9.9 Wh$

Según el simulador, el consumo energético medio por nodo es de 32.06 J (8.9 mWh) al enviar 120 paquetes, cada uno de ellos con un tamaño de 16 bytes, (tamaño suficiente para transmitir las aceleraciones de los tres ejes medidas por el sensor).

El consumo energético por paquete es de:

$$\frac{8.9 mWh}{120 \text{ paquetes}} = 0.0741 mWh/paquete$$

Transmitiendo 2 paquetes por hora:

$$2 \frac{\text{paquetes}}{\text{hora}} * 0.0741 \frac{mWh}{\text{paquete}} = 0.1483 \frac{mWh}{\text{hora}}$$

La duración estimada es:

$$\frac{9.9 Wh}{0.1483 mWh/hora} = 66.741 \text{ horas} = 7.61 \text{ años}$$

Esto demuestra que los nodos pueden operar de manera autónoma durante varios años, sin necesidad de recarga o sustitución de batería. Esto convierte al sistema sostenible y de bajo mantenimiento ideal para entornos IoT.

No obstante, este cálculo debe entenderse como una aproximación, ya que, como se ha mencionado a lo largo del proyecto, únicamente se transmitirá cuando los valores del acelerómetro superen un determinado umbral. Esto significa que la frecuencia de envío

puede variar, pudiendo no ser necesario transmitir en ciertos periodos de tiempo, mientras que en otros la actividad puede ser superior.

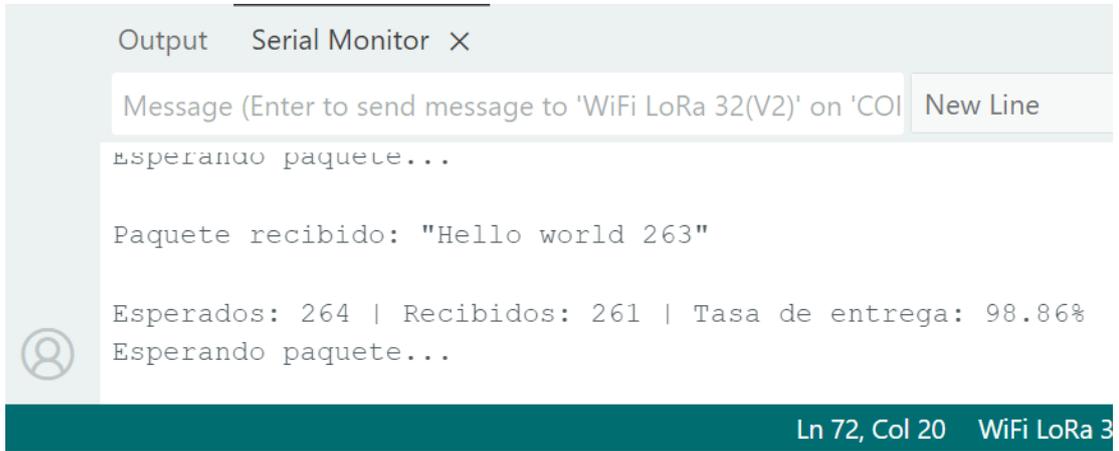
#### ***4.2 IMPLEMENTACIÓN FÍSICA Y VALIDACIÓN DEL SISTEMA***

Con el fin de validar el funcionamiento del sistema diseñado, se procedió a realizar una implementación física a pequeña escala, simulando los tres componentes principales del sistema final. Un nodo transmisor, un Gateway (nodo con capacidad de recibir y transmitir), y un nodo receptor final. Para ello, se utilizaron tres microcontroladores ESP32 con un módulo LoRa integrado, y un sensor acelerómetro ADXL345 integrado en el nodo transmisor.

La primera etapa consistió en realizar pruebas básicas de comunicación LoRa entre dos módulos ESP32, el transmisor y receptor. Se implementó un simple código para enviar mensajes de prueba “Hello World” de forma periódica desde el nodo emisor, y, se verificó en el receptor la recepción de los paquetes. Este código cuenta con la capacidad de modificar parámetros como la potencia de transmisión, el tamaño del paquete y el spreading factor.

Una vez probada la comunicación, se implementó un contador de paquetes recibidos correctamente, con el objetivo de evaluar la tasa de recepción. Para ello se realizaron múltiples transmisiones dentro de un entorno cerrado, más concretamente, la universidad.

En particular, se probó la comunicación entre pasillos, aulas y numerosos obstáculos físicos. En una distancia de aproximadamente 50 metros, el sistema logró mantener una comunicación estable, como se muestra en la imagen adjunta. Durante esta prueba, el nodo transmisor operaba con una potencia de 5 dBm y un spreading factor de 7.



```
Output Serial Monitor X
Message (Enter to send message to 'WiFi LoRa 32(V2)' on 'COI New Line
esperando paquete...

Paquete recibido: "Hello world 263"

Esperados: 264 | Recibidos: 261 | Tasa de entrega: 98.86%
Esperando paquete...
```

Nodo Receptor

Como muestra la imagen, un total de 264 paquetes fueron transmitidos, donde únicamente 3 paquetes fueron recibidos de manera incorrecta, lo que significa una tasa de entrega exitosa del 98.96%.

Una vez validada la comunicación LoRa y confirmada su fiabilidad en condiciones reales, se procedió con la integración del sensor acelerómetro ADXL345 en el nodo transmisor.

Para ello se utilizó la comunicación I2C entre el ESP32 y el acelerómetro, utilizando librerías que facilitan la lectura de datos del sensor. Gracias a estas librerías, fue posible obtener el ángulo de inclinación en cada eje, aprovechando la relación directa entre las aceleraciones medidas por el sensor y el ángulo de inclinación del mismo. Esto permitió implementar un control basado en variaciones angulares.

El código fue desarrollado, de manera que solo enviará datos cuando la variación angular en el eje x o en el eje y, fuese superior a un umbral establecido en 10°. Se exigió que dicha variación se mantuviera al menos en 5 lecturas consecutivas, para evitar falsas transmisiones por movimientos breves.

La parte del código encargado de calcular esta variación angular, es la siguiente:

```
void loop() {
  xyzFloat angle;
  myAcc.getAngles(&angle);

  // Calcular variaciones absolutas respecto a la lectura anterior
  float deltaX = abs(angle.x - prevX);
  float deltaY = abs(angle.y - prevY);

  // Actualizar angulo previo para próxima iteración
  prevX = angle.x;
  prevY = angle.y;

  // Contar variaciones consecutivas de X
  if (deltaX > angulo_umbral) {
    cuentaDeX++;
  } else {
    cuentaDeX = 0;
  }

  // Contar variaciones consecutivas de Y
  if (deltaY > angulo_umbral) {
    cuentaDeY++;
  } else {
    cuentaDeY = 0;
  }

  // Si no se ha transmitido y supera el conteo, se transmite
  if (!transmitido && (cuentaDeX >= conteoRequerido || cuentaDeY >=
  conteoRequerido)) {
    Serial.println("Variación detectada: Transmitiendo LoRa");

    transmitido = true;

    // Resetear para nuevas detecciones
    cuentaDeX = 0;
    cuentaDeY = 0;
  }

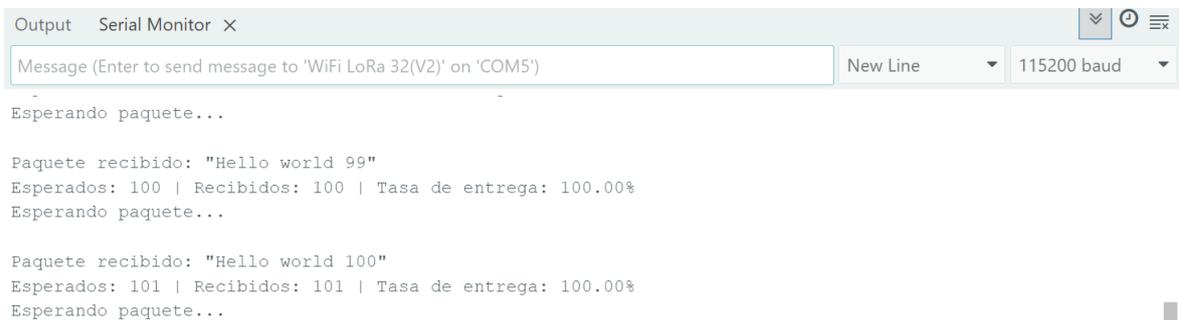
  // Resetear flag para nuevos eventos
  if (cuentaDeX == 0 && cuentaDeY == 0) {
    transmitido = false;
  }
}
```

```
delay(300);  
}
```

Una vez validadas las pruebas de comunicación en un entorno cerrado, se procedió a evaluar el alcance del sistema bajo condiciones reales. Para ello se instaló un uno de los microcontroladores ESP32 con módulo LoRa en la ventana de una habitación, actuando como nodo transmisor. Posteriormente, se realizaron desplazamientos a distintas ubicaciones, para comprobar la capacidad del receptor.

Cabe mencionar que, durante estas pruebas, ambos, tanto el transmisor como el receptor se configuraron con una potencia de 20 dBm y un spreading factor máximo, de 12. Parámetros configurados para maximizar el alcance de estos dispositivos.

El primer desplazamiento se ubicó a 1 kilómetro del transmisor. La zona correspondía a una zona residencial, donde no existía línea de visión directa entre los dispositivos, debido a la presencia de numerosos obstáculos como árboles, muros y casas. A pesar de ello, se obtuvo una tasa de entrega del 100%, con 100 paquetes transmitidos y 100 recibidos correctamente. Este resultado fue sorprendente, debido a las condiciones desfavorables. Superó de gran manera las expectativas en cuanto al rendimiento del sistema en escenarios urbanos no despejados.

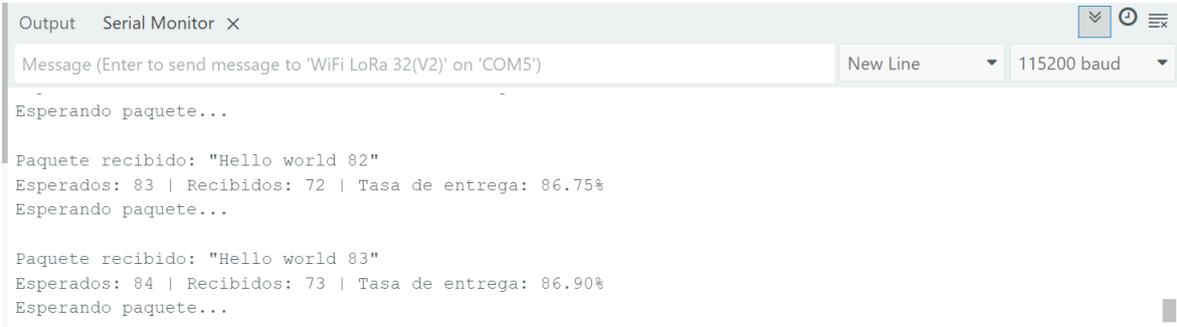


The screenshot shows a Serial Monitor window with the following content:

```
Output Serial Monitor x  
Message (Enter to send message to 'WiFi LoRa 32(V2)' on 'COM5') New Line 115200 baud  
Esperando paquete...  
Paquete recibido: "Hello world 99"  
Esperados: 100 | Recibidos: 100 | Tasa de entrega: 100.00%  
Esperando paquete...  
Paquete recibido: "Hello world 100"  
Esperados: 101 | Recibidos: 101 | Tasa de entrega: 100.00%  
Esperando paquete...
```

Posteriormente se repitió la prueba, en una segunda ubicación, situada a aproximadamente 2.5 kilómetros del transmisor. En esta zona, las condiciones de visibilidad entre nodos fueron incluso más desfavorables, con una mayor cantidad de obstáculos entre transmisor y

receptor. No obstante, la tasa de entrega fue de 86.90%, valor igualmente notable considerando la complejidad de la situación.



```
Output Serial Monitor x
Message (Enter to send message to 'WiFi LoRa 32(V2)' on 'COM5') New Line 115200 baud
-
Esperando paquete...
Paquete recibido: "Hello world 82"
Esperados: 83 | Recibidos: 72 | Tasa de entrega: 86.75%
Esperando paquete...
Paquete recibido: "Hello world 83"
Esperados: 84 | Recibidos: 73 | Tasa de entrega: 86.90%
Esperando paquete...
```

### **4.3 INTEGRACIÓN DEL MÓDULO MQTT**

Una vez validada el funcionamiento del acelerómetro, detectando las variaciones angulares mediante el sensor ADXL345 y la transmisión LoRa, se procedió a realizar la conexión final con un servidor MQTT.

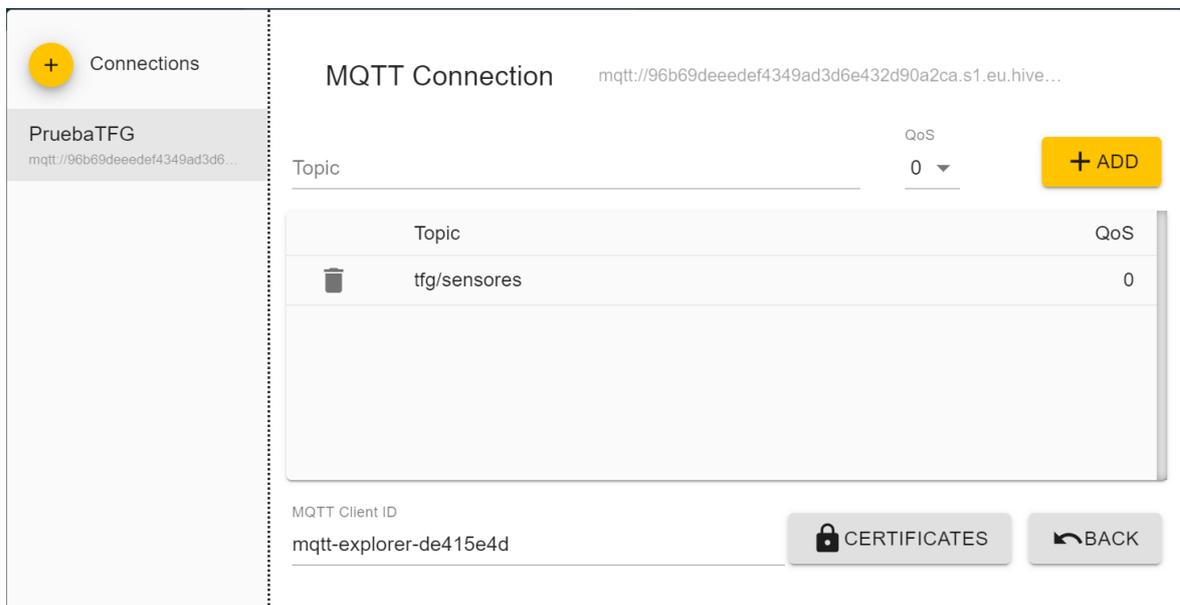
En el diseño original del sistema propuesto a lo largo de este proyecto, se hablaba de que la información es enviada de los sensores al Gateway LoRa, y este hacia un servidor LoRaWAN, encargado de gestionar los dispositivos y paquetes recibidos.

Sin embargo, para la implementación física llevada en esta parte del proyecto, se ha omitido la parte del servidor LoRaWAN, ya que se tiene un módulo ESP32 con módulo LoRa que actúa como un Gateway LoRa, y, por lo tanto, se ha optado por una arquitectura simplificada, transmitiendo la información directamente a un servidor MQTT a través de una conexión Wifi.

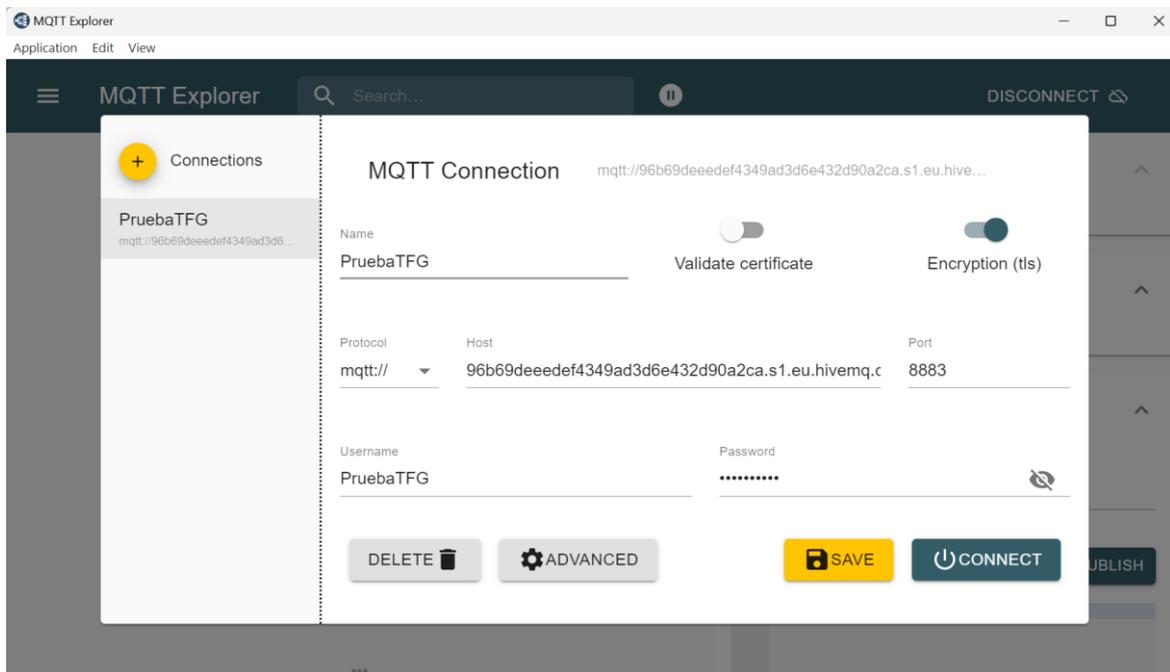
Esta decisión, se tomó únicamente para reducir la complejidad del sistema en esta fase de validación, permitiendo enfocarse en la transmisión de información desde el sensor hasta la nube. A pesar de ello, el diseño se mantiene con el servidor LoRaWAN, para una implementación futura más completa.

Para esta implementación, se utilizó el servidor MQTT proporcionado por HiveMQ. El nodo ESP32 realiza la conexión al bróker MQTT utilizando las credenciales configuradas en la página del servidor, y, se transmite un mensaje sobre la variación detectada cada vez que se supere el umbral definido. Con ayuda de las librerías <PubSubClient.h>, <WiFiClientSecure.h> y <WiFi.h> la comunicación hacía el servidor MQTT se realizó mediante el comando `client.publish(topic, message)`.

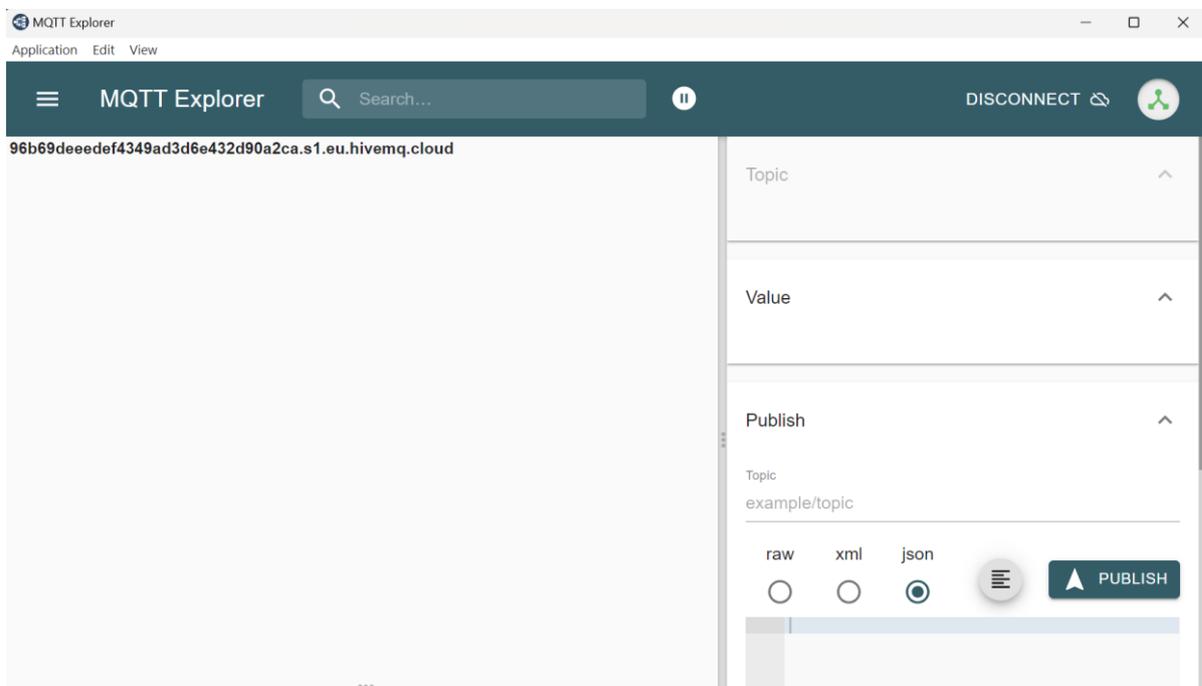
A continuación, se procedió a verificar la recepción de los mensajes en el servidor MQTT, mediante el software MQTT Explorer, herramienta para visualizar en tiempo real los tópicos y mensajes publicados en el servidor. Se muestra la interfaz gráfica afirmando que el canal de comunicación está correctamente establecido.



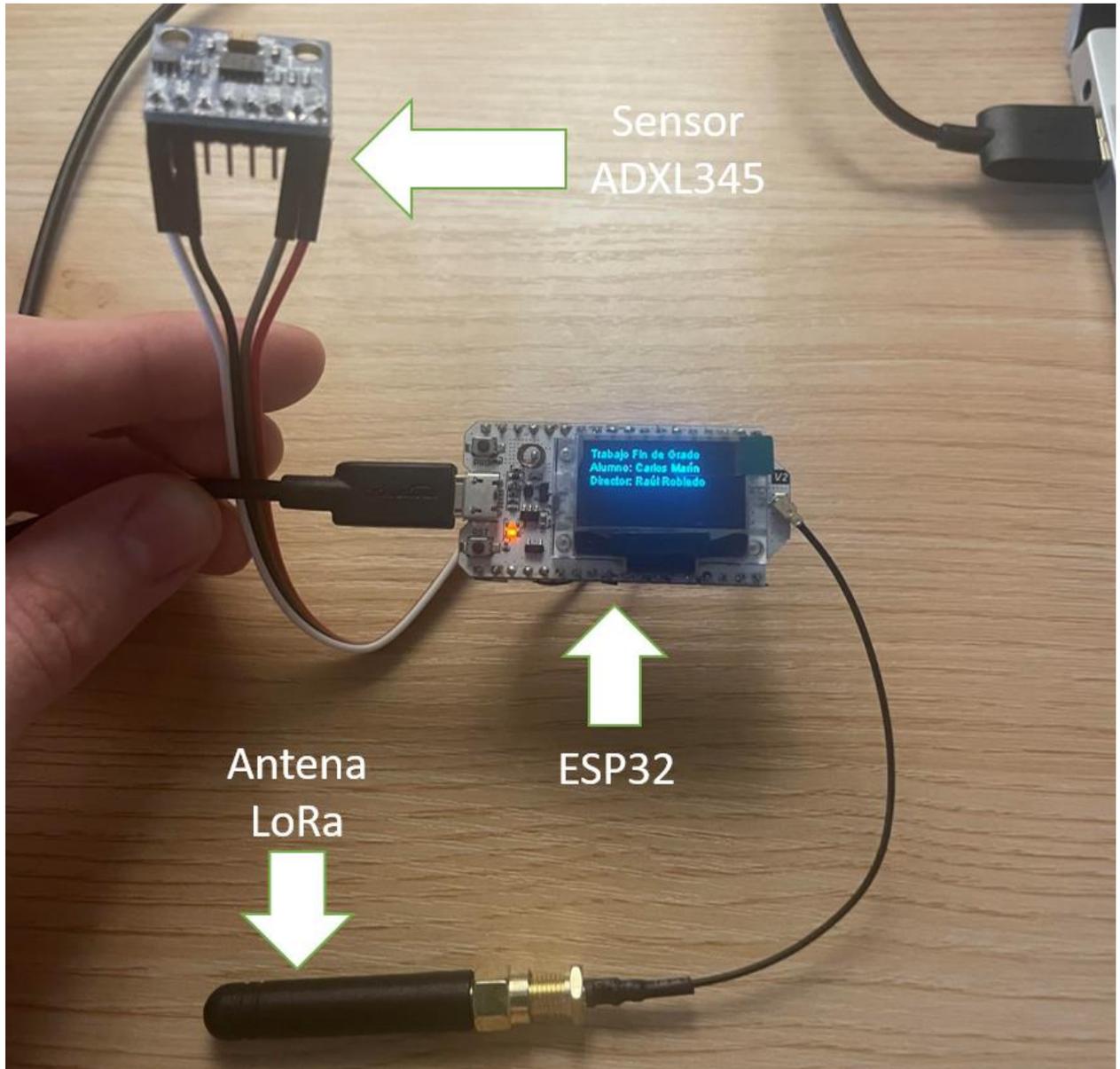
Suscripción al tópico tfg/sensores



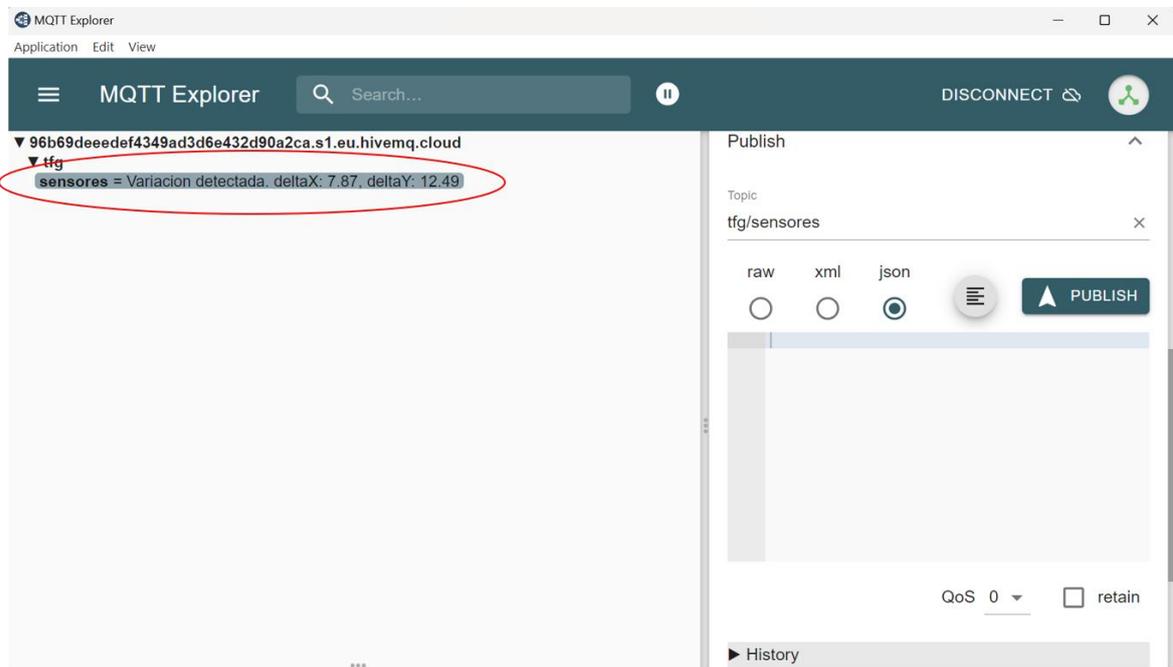
Conexión con el Broker de HiveMQ a través de MQTT Explorer



Interfaz gráfica de MQTT Explorer



Si se procede a agitar el sensor, se detecta y transmite la variación del ángulo



Se recibe la información en el servidor MQTT

## **Capítulo 5. ALINEACIÓN DEL PROYECTO CON LOS ODS**

### **ODS 3 – Salud y Bienestar**

El uso de sensores para anticipar fallos estructurales, reduce los riesgos asociados, protegiendo vidas humanas y promoviendo un entorno más seguro.

### **ODS 4 – Educación de Calidad**

Desarrollar este tipo de proyectos en la universidad promueve la formación en tecnologías emergentes como sensores, comunicaciones LoRa y protocolos de red como MQTT.

Este trabajo ha requerido un estudio teórico previo para alcanzar una comprensión profunda del funcionamiento del sistema, que posteriormente se ha complementado con la implementación física de los componentes.

### **ODS 7 – Energía Asequible y No Contaminante**

El sistema ha sido diseñado para ser energéticamente eficiente, usando microcontroladores de bajo consumo junto a la tecnología LoRa para transmisiones de largo alcance con mínimo gasto energético.

### **ODS 9 – Industria, Innovación e Infraestructura**

Este proyecto fomenta la digitalización de las infraestructuras, mediante el uso de dispositivos IoT y protocolos para transmitir información a la nube. Se impulsa así la innovación tecnológica, mejorando la eficiencia y vida útil de las infraestructuras.

### **ODS 11 – Ciudades y Comunidades Sostenibles**

La capacidad de detección temprana de inclinaciones o deformaciones estructurales, permite adoptar medidas preventivas, mejorando así la seguridad y resiliencia de las ciudades ante riesgos estructurales. En este sentido, el proyecto contribuye a la creación de entornos más seguros e inteligentes.

#### ODS 12 – Producción y Consumo Responsable

El proyecto promueve un uso eficiente de los recursos electrónicos, mediante la selección de componentes de bajo consumo para minimizar el consumo energético. Esta estrategia, reduce el impacto ambiental, fomentando un diseño respetuoso con el entorno.

## Capítulo 6. CONCLUSIONES Y TRABAJOS FUTUROS

Este proyecto ha desarrollado una solución para la monitorización de vibraciones en seguidores solares, con el objetivo de prevenir posibles daños estructurales causados por fenómenos de resonancia. A lo largo del trabajo se han cumplido todos los objetivos planteados, alcanzando avances tanto a nivel teórico como práctico.

Los principales objetivos alcanzados han sido:

- Selección e integración de protocolos de comunicación eficientes, empleando LoRaWAN para la comunicación de largo alcance y MQTT para la distribución de datos en la nube.
- Diseño de una arquitectura IoT, basada en ESP32 con módulos LoRa y sensores ADXL345.
- Validación del rendimiento del sistema mediante simulaciones con LoRaWAN-SIM, evaluando diferentes configuraciones hasta 5 gateways con 1000 nodos.
- Implementación física, realizando pruebas reales y confirmando el buen rendimiento de estos dispositivos.
- Estimación de la autonomía energética, demostrando que los nodos podrían alcanzar hasta 7.6 años con baterías estándar de 3000 mAh.

Además, el análisis detallado de los resultados ha permitido concluir que una configuración con 4 gateways representa un punto óptimo, combinando eficiencia energética, alta tasa de entrega y tolerancia a fallos.

Este sistema cumple con los requisitos de bajo consumo, bajo coste, autonomía prolongada y fiabilidad, lo que lo convierte en una solución robusta para aplicaciones reales.

Entre las posibles líneas de desarrollo y mejora del sistema destacan:

- Despliegue en una planta fotovoltaica real, para evaluar el comportamiento en condiciones operativas.
- Optimización del algoritmo de detección, incorporando nuevas técnicas para identificar patrones de vibración anómalos.
- Integración completa con un servidor de LoRaWAN, para una gestión más avanzada de la red.

## Capítulo 7. BIBLIOGRAFÍA

- [1] Santander Universidades, "Dispositivos IoT: qué son y ejemplos," 2024. Available: <https://about:blank>.
- [2] Anonymous . *Internet de las cosas (IoT): dispositivos conectados en el mundo 2015-2030*. Available: <https://es.statista.com/estadisticas/517654/prevision-de-la-evolucion-de-los-dispositivos-conectados-para-el-internet-de-las-cosas-en-el-mundo/>.
- [3] Anonymous "LoRaWAN: La guía más completa | Sistelec," 2024. Available: <https://sistelec.es/blog/lorawan-guia-completa/>.
- [4] Y. K. Huang, "¿Cuál es la tecnología detrás de la frecuencia LoRa? - MOKOSmart," 2020. Available: <https://www.mokosmart.com/lora-frequency/>.
- [5] Anonymous . *Wyld Networks*. Available: <https://wyldnetworks.com/blog/what-is-a-lorawan-gateway>.
- [6] C. Román, "LoRaWAN: Qué es, Cómo Funciona y Sus Aplicaciones en IoT," 2025. Available: <https://www.monolitic.com/lorawan-que-es-como-funciona/>.
- [7] Anonymous . *Protocolos y tecnologías de IoT | Microsoft Azure*. Available: <https://azure.microsoft.com/es-mx/solutions/iot/iot-technology-protocols>.
- [8] de 2, La tecnología de radio inalámbrica de corto alcance opera en la banda de frecuencia, auriculares, 4 GHz La tecnología Bluetooth permite que las PCs y los dispositivos móviles sincronicen con los periféricos como and Compatible, Teclados O. Ratones Inalámbricos Sin Necesidad De Utilizar Cables O. Una Infraestructura De Red. *¿Cómo funciona la tecnología Bluetooth®?*. Available: <https://www.intel.com/content/www/xl/es/products/docs/wireless/how-does-bluetooth-work.html>.
- [9] J. Arellano. (-06-01). *Bluetooth vs. Wi-Fi for IoT: Which is Better?*. Available: <https://www.verytechnology.com/insights/bluetooth-vs-wifi-for-iot-which-is-better>.
- [10] Anonymous (-11-13). *Qué bandas de frecuencias WiFi hay: Explicación 2.4 GHz, 5 GHz y 6 GHz*. Available: <https://www.redeszone.net/tutoriales/redes-wifi/bandas-frecuencias-wi-fi/>.
- [11] Anonymous . *IoT Wireless Network Protocols*. Available: <https://es.farnell.com/iot-wireless-network-protocols>.

[12] Anonymous "untitled," Available: [https://scdn.rohde-schwarz.com/ur/pws/dl\\_downloads/dl\\_application/application\\_notes/rsi03/RSI03\\_0e.pdf](https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/rsi03/RSI03_0e.pdf).

[13] Anonymous (-06-08). *Networking methods / ZigBee Networking Guide*. Available: <https://development.libelium.com/zigbee-networking-guide/networking-methods>.

[14] Anonymous "Sigfox - wireless ultra narrow band communication solution," Available: <https://stackforce.com/en/portfolio/technologies/sigfox>.

[15] Anonymous "cobertura – SIGFOX," Available: <https://sigfox.com.py/cobertura/>.

[16] Anonymous "¿Qué es NB-IoT y cómo funciona?" 2023. Available: <https://www.telefonica.com/es/sala-comunicacion/blog/que-es-nb-iot-y-como-funciona/>.

[17] N. Huang, "LoRaWAN versus NB-IoT: ¿Cómo se comparan y difieren?" 2024. Available: <https://www.mokosmart.com/lorawan-vs-nb-iot-how-do-they-compare-and-differ/>.

[18] Anonymous . *LoRaWAN Architecture*. Available: <https://www.thethingsnetwork.org/docs/lorawan/architecture/>.

[19] Anonymous . *Device Classes*. Available: <https://www.thethingsnetwork.org/docs/lorawan/classes/>.

[20] Anonymous . *What is MQTT and When You Should Use It*. Available: <https://www.pubnub.com/blog/what-is-mqtt-use-cases/>.

[21] A. H, "MQTT vs HTTP: ¿qué protocolo es mejor para IoT?" 2020. Available: <https://borrowbits.com/2020/04/mqtt-vs-http-que-protocolo-es-mejor-para-iot/>.

[22] Anonymous . *¿Qué es HTTP?*. Available: <https://www.cloudflare.com/es-es/learning/ddos/glossary/hypertext-transfer-protocol-http/>.

[23] Anonymous "MQTT y la escalabilidad del broker," 2021. Available: <https://www.smartnet.com.co/mqtt-y-la-escalabilidad-del-broker/>.

[24] Anonymous (-05-20). *What is MQTT Quality of Service (QoS) 0,1, & 2? – MQTT Essentials: Part 6*. Available: <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>.

[25] Anonymous . *¿Qué es MQTT? Definición y detalles*. Available: <https://www.paessler.com/es/it-explained/mqtt>.

[26] D. Zorbas *et al*, "Optimal Data Collection Time in LoRa Networks—A Time-Slotted Approach," *Sensors*, vol. 21, (4), pp. 1193, 2021. Available: <https://www.mdpi.com/1424-8220/21/4/1193>. DOI: 10.3390/s21041193.

[27] D. Zorbas, "LoRaWAN network coexistence in the EU868 spectrum," in November 25, 2024, Available: [https://www.researchgate.net/publication/385389121\\_LoRaWAN\\_Network\\_Coexistence\\_in\\_the\\_EU868\\_Spectrum](https://www.researchgate.net/publication/385389121_LoRaWAN_Network_Coexistence_in_the_EU868_Spectrum). DOI: 10.1109/CSCN63874.2024.10849695.

[28] S. Javed and D. Zorbas, "LoRaWAN downlink policies for improved fairness," in November 27, 2022, Available: [https://www.researchgate.net/publication/365604000\\_LoRaWAN\\_Downlink\\_Policies\\_for\\_Improved\\_Fairness](https://www.researchgate.net/publication/365604000_LoRaWAN_Downlink_Policies_for_Improved_Fairness). DOI: 10.1109/CSCN57023.2022.10050924.