

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE MASTER

OPTIMIZACIÓN DE UNA RED IOT LORA MEDIANTE AJUSTES REALIZADOS EN LOS PARÁMETROS AVANZADOS DE LA RED



Estudiante: Sedano Isasi, Iker

Director: Vélez Elordi, Manuel María

Curso: 2023-2024

Fecha: Bilbao, 6, junio, 2024

Resumen

En los últimos años, el IoT ha crecido significativamente, con la tecnología LoRa destacando por su amplio alcance y bajo consumo energético. Itelazpi, una empresa pública vasca, ha desplegado una extensa red LoRa en la CAV, conectando más de 100.000 dispositivos IoT, la gran mayoría siendo dispositivos *metering* de agua. Esta red ha sido clave para la monitorización y gestión eficiente del agua, permitiendo la recopilación de datos en tiempo real y optimizando la administración de recursos hídricos.

Este trabajo de fin de master se enmarca en unas prácticas de cooperación educativa en Itelazpi, desarrolladas durante más de dos años y medio. Este trabajo se propone optimizar la red existente mediante ajustes en los parámetros de calidad de servicio (QoS) del protocolo LoRaWAN, sin necesidad de añadir más hardware. La localidad de Galdakao, con una cobertura adecuada y una alta densidad de dispositivos, ha sido seleccionada como zona de estudio piloto para implementar y validar estas mejoras.

Para ello, primero se definirán los parámetros que se van a utilizar para poder medir la calidad de la red, al igual que los métodos de extracción de estos y la forma en la que se presentarán en forma de gráficos para poder analizarlos.

Una vez definido esto, se desarrollará una aplicación de escritorio capaz de obtener los gráficos resultantes con la mínima interacción del usuario posible, para así poder automatizar la mayoría del proceso del análisis y ser capaz de realizar estos análisis en cualquier zona de la red.

Para poder determinar el impacto de la modificación de ciertos parámetros QoS del protocolo LoRaWAN, primero se realizará un análisis previo a cualquier cambio, para poder observar el estado actual de la red en la zona estudiada.

Una vez hecho esto, se modificarán ciertos parámetros QoS en función de los resultados que se quieran obtener, y se volverá a realizar otro análisis, para observar el estado de la red después de haberle aplicado los cambios.

Finalmente, tras estudiar el impacto de los cambios realizados, se realizarán otros cambios finales con el fin de maximizar las mejoras obtenidas y paliar el impacto negativo que pueda suponer el cambio de ciertos parámetros. Esto servirá para volver a analizar el estado de la red y obtener conclusiones sobre el impacto del cambio de diferentes parámetros, los cuales serán aplicables al resto de la red.

Palabras clave: IoT, metering, LoRa, LoRaWAN, parámetros QoS

Laburpena

Azken urteotan, IoT nabarmen hazi da, LoRa teknologia bere irismen zabalagatik eta energia-kontsumo txikiagatik nabarmentzen delarik. Itelazpi euskal enpresa publikoak LoRa sare zabala zabaldu du EAEn, 100.000 IoT gailu baino gehiago konektatuz, gehienak ur-*metering* gailuak izanik. Sare hori funtsezkoa izan da ura modu eraginkorrean monitorizatzeko eta kudeatzeko, datuak denbora errealean biltzea ahalbidetuz eta baliabide hidrikoen administrazioa optimizatuz.

Master amaierako lan hau Itelazpin bi urte eta erdiz baino gehiagoz garatutako hezkuntza-lankidetzako praktika batzuen barruan kokatzen da. Lan honek egungo sarea optimizatzea proposatzen du, LoRaWAN protokoloeko zerbitzuaren kalitate-parametroetan (QoS) doikuntzak eginez, hardware gehiago gehitu beharrik gabe. Galdakaoko herria, estaldura egokiarekin eta gailu-dentsitate handiarekin, azterketa-eremu pilotu gisa hautatu da hobekuntza horiek inplementatzeko eta baliozkotzeko.

Horretarako, lehenik eta behin, sarearen kalitatea neurtzeko erabiliko diren parametroak definituko dira, bai eta horiek erazteko metodoak eta analizatu ahal izateko grafiko moduan aurkezteko modua ere.

Behin hori definituta, mahaigaineko aplikazio bat garatuko da, lortutako grafikoak erabiltzailearen ahalik eta elkarrekintza txikienarekin lortzeko gai izango dena, horrela analisiaren prozesu gehiena automatizatu ahal izateko eta analisi horiek sareko edozein gunetan egiteko gai izateko.

LoRaWAN protokoloeko QoS parametro jakin batzuen aldaketaren eragina zehaztu ahal izateko, lehenik eta behin, edozein aldaketaren aurreko analisi bat egingo da, aztertutako eremuan sarearen egungo egoera behatu ahal izateko.

Hori egin ondoren, QoS parametro batzuk aldatuko dira lortu nahi diren emaitzen arabera, eta beste analisi bat egingo da, aldaketak aplikatu ondoren sarearen egoera behatzeko.

Azkenik, egindako aldaketen eragina aztertu ondoren, beste azken aldaketa batzuk egingo dira, lortutako hobekuntzak maximizatzeko eta parametro jakin batzuk aldatzeak ekar dezakeen inpaktu negatiboa arintzeko. Horrek balio izango du sarearen egoera berriro aztertzeko eta hainbat parametroren aldaketaren inpaktuari buruzko ondorioak ateratzeko. Parametro horiek sarearen gainerako ataletan aplikatu ahal izango dira.

Gako-hitzak: IoT, metering, LoRa, LoRaWAN, QoS parametroak

Abstract

In recent years, IoT has grown significantly, with LoRa technology standing out for its wide range and low power consumption. Itelazpi, a Basque public company, has deployed an extensive LoRa network in the BAC, connecting more than 100,000 IoT devices, the vast majority being water metering devices. This network has been key to efficient water monitoring and management, enabling real-time data collection and optimising water resource management.

This master's thesis is part of an educational cooperation internship at Itelazpi, carried out over more than two and a half years. This work aims to optimise the existing network by adjusting the quality of service (QoS) parameters of the LoRaWAN protocol, without the need to add more hardware. The town of Galdakao, with adequate coverage and a high density of devices, has been selected as a pilot study area to implement and validate these improvements.

To do this, we will first define the parameters that will be used to measure the quality of the network, as well as the methods for extracting them and the way in which they will be presented in the form of graphs for analysis.

Once this has been defined, a desktop application will be developed capable of obtaining the resulting graphs with as little user interaction as possible, in order to automate most of the analysis process and be able to perform these analyses in any area of the network.

In order to determine the impact of modifying certain QoS parameters of the LoRaWAN protocol, an analysis will first be performed prior to any changes, in order to observe the current state of the network in the area under study.

Once this has been done, certain QoS parameters will be modified according to the results to be obtained, and another analysis will be carried out again, to observe the state of the network after the changes have been applied.

Finally, after studying the impact of the changes made, other final changes will be made in order to maximise the improvements obtained and mitigate the negative impact of changing certain parameters. This will serve to re-analyse the state of the network and obtain conclusions on the impact of the change of different parameters, which will be applicable to the rest of the network.

Keywords: IoT, metering, LoRa, LoRaWAN, QoS parameters

Índice

1. Introducción.....	1
2. Contexto	3
2.1. Zona de estudio	4
3. Objetivos y alcance del trabajo	6
4. Beneficios que aporta el trabajo	7
4.1. Beneficios sociales	7
4.2. Beneficios económicos	7
4.3. Beneficios técnicos	7
5. Descripción de la solución propuesta	8
5.1. Metodología de captura de datos y resultados	8
5.2. Creación de interfaz de usuario	22
5.3. Estudio inicial	57
5.4. Iteraciones de mejora	70
5.5. Configuración final.....	82
6. Planificación del proyecto	94
6.1. Paquetes de trabajo.....	94
6.2. Diagrama de GANTT	101
7. Presupuesto	103
8. Conclusiones.....	105
9. Bibliografía.....	106

Lista de ilustraciones

Ilustración 1: Arquitectura de red LoRa [1]	1
Ilustración 2: Diagrama de la red LoRa de Itelazpi	2
Ilustración 3: Esquema de problemas de densificación de la red LoRa de Itelazpi	3
Ilustración 4: Ubicación de los Gateways en Galdakao.....	5
Ilustración 5: Esquema de planificación del proyecto	8
Ilustración 6: Información básica de un device en Orbiwise.....	9
Ilustración 7: Información detallada de un device en Orbiwise I	10
Ilustración 8: Información detallada de un device en Orbiwise II	10
Ilustración 9: Mensaje MAC enviado por un dispositivo.....	10
Ilustración 10: Información básica de un Gateway en Orbiwise I	11
Ilustración 11: Información básica de un Gateway en Orbiwise II.....	12
Ilustración 12: Información detallada de un Gateway en Orbiwise	12
Ilustración 13: Endpoint de la API de Orbiwise para obtener todos los devices de un grupo	14
Ilustración 14: Endpoint de la API de Orbiwise para obtener los payloads de un device.....	15
Ilustración 15: Extracto de las opciones disponibles del endpoint de obtención de payloads ...	15
Ilustración 16: Respuesta obtenida de una llamada al endpoint de obtención de payloads	16
Ilustración 17: Endpoint para obtener todos los Gateways asociados a un usuario	17
Ilustración 18: Respuesta obtenida de una llamada al endpoint de listado de Gateways	17
Ilustración 19: Endpoint de la API NST para obtener parámetros avanzados de un Gateway ...	18
Ilustración 20: Formato de respuesta de una llamada al endpoint de obtención de parámetros avanzados de un Gateway.....	18
Ilustración 21: Diagrama de bloques de la obtención de parámetros de los devices	23
Ilustración 22: Diagrama de bloques de la obtención de parámetros de los Gateways.....	25
Ilustración 23: Explicación gráfica de la fórmula de Haversine [6]	28
Ilustración 24: Diagrama de bloques de la GUI funcional	33
Ilustración 25: Pantalla principal (GUI funcional)	34
Ilustración 26: Pantalla de selección de grupo (GUI funcional).....	34
Ilustración 27: Pantalla de selección de tiempo (GUI funcional).....	35
Ilustración 28: Pantalla de obtención de payloads (GUI funcional).....	35
Ilustración 29: Pantalla de guardado (GUI funcional)	36
Ilustración 30: Pantalla de datos guardados (GUI funcional)	37
Ilustración 31: Pantalla de ingresar credenciales para los Gateways (GUI funcional).....	37
Ilustración 32: Pantalla con la lista de todos los Gateways (GUI funcional)	38
Ilustración 33: Pantalla con Gateways seleccionados (GUI funcional)	38
Ilustración 34: Pantalla de obtención de parámetros de Gateways (GUI funcional)	39
Ilustración 35: Pantalla de creación de gráficos (GUI funcional).....	39
Ilustración 36: Pantalla de finalización (GUI funcional)	39
Ilustración 37: Pantalla de selección de carpetas (GUI funcional)	40
Ilustración 38: Pantalla de creación de gráficos combinados (GUI funcional)	40
Ilustración 39: Diagrama de bloques de la GUI final.....	41
Ilustración 40: Diagrama de funcionamiento de "Tkinter-Designer"	42
Ilustración 41: Pantalla principal (GUI final).....	43
Ilustración 42: Pantalla de selección de grupo (GUI final)	44
Ilustración 43: Pantalla de selección de tiempo (GUI final)	44
Ilustración 44: Pantalla de obtención de payloads (GUI final)	45

Ilustración 45: Pantalla de guardado (GUI final).....	46
Ilustración 46: Pantalla de datos guardados (GUI final).....	47
Ilustración 47: Pantalla de ingresar credenciales para los Gateways (GUI final)	47
Ilustración 48: Pantalla con la lista de todos los Gateways (GUI final)	48
Ilustración 49: Pantalla con Gateways seleccionados (GUI final)	48
Ilustración 50: Pantalla de obtención de parámetros de Gateways (GUI final)	49
Ilustración 51: Pantalla de creación de gráficos (GUI final)	49
Ilustración 52: Pantalla de finalización (GUI final)	50
Ilustración 53: Pantalla de elección de numero de análisis (GUI final)	50
Ilustración 54: Pantalla de selección de carpetas (GUI final)	51
Ilustración 55: Pantalla de creación de gráficos combinados (GUI final).....	52
Ilustración 56: Resultado de crear el ejecutable	53
Ilustración 57: Ejecutable de la aplicación	54
Ilustración 58: Ventana emergente al introducir credenciales erróneas.....	55
Ilustración 59: Información estática de un Gateway	57
Ilustración 60: Archivos JSON de los parámetros de cada Gateway.....	58
Ilustración 61: Ejemplo de datos de calidad de señal obtenidos de un Gateway	58
Ilustración 62: Información estática de un device	59
Ilustración 63: Información de cada mensaje transmitido de un device	60
Ilustración 64: Ejemplo de datos unificados de un Gateway	61
Ilustración 65: Ejemplo de datos unificados de un device	61
Ilustración 66: Mapa de Gateways y devices de Galdakao	62
Ilustración 67: Zoom del mapa de Gateways y devices de Galdakao	62
Ilustración 68: Gráfico de número de frames de uplink por Gateway (estudio inicial)	63
Ilustración 69: Grafico de la distribución de la media de RSSI (estudio inicial).....	64
Ilustración 70: Distribución de "Average Gateways reached" (estudio inicial)	64
Ilustración 71: Distribución de "Last SF used" (estudio inicial)	65
Ilustración 72: Distribución de "Lost frames" (estudio inicial)	66
Ilustración 73: Distribución de "Received frames" (estudio inicial).....	66
Ilustración 74: Márgenes de potencia de los mensajes (estudio inicial)	67
Ilustración 75: Distribución de "Average SNR" (estudio inicial)	67
Ilustración 76: Media de RSSI por cada SF (estudio inicial).....	68
Ilustración 77: Media de SNR por cada SF (estudio inicial)	68
Ilustración 78: Media de mensajes perdidos por distancia al Main gateway (estudio inicial)....	69
Ilustración 79: Media de mensajes perdidos por SF (estudio inicial)	69
Ilustración 80: Mensaje de configuración del Network Server a un device.....	73
Ilustración 81: Comando MAC enviado en el mensaje de configuración	73
Ilustración 82: Comparativa de la media de RSSI de los mensajes recibidos en cada Gateway .	75
Ilustración 83: Comparación del número de mensajes recibidos por cada Gateway.....	75
Ilustración 84: Comparación del número de Gateways al que llegan los mensajes.....	76
Ilustración 85: Comparación del último SF utilizado por los devices.....	76
Ilustración 86: Comparación del ToA relativo a la distancia al Main Gateway	77
Ilustración 87: Comparación de RSSI en función de SF	77
Ilustración 88: Comparación de SNR en función de SF	78
Ilustración 89: Comparación de la distribución de RSSI.....	78
Ilustración 90: Comparación de la distribución de SNR	79
Ilustración 91: Comparación del número de mensajes recibidos.....	79

Ilustración 92: Comparación de porcentaje de mensajes perdidos en relación a la distancia al Main Gateway	80
Ilustración 93: Comparación de porcentaje de perdidas en relación al SF	80
Ilustración 94: Relación entre el SF y ToA	81
Ilustración 95: SNR mínimo para demodular dependiendo el SF [19]	81
Ilustración 96: Comparativa de la media de RSSI recibida por cada Gateway	85
Ilustración 97: Comparación del número de mensajes recibidos por cada Gateway.....	86
Ilustración 98: Comparación del número de Gateways al que llegan los mensajes.....	86
Ilustración 99: Comparación del último SF utilizado por los devices.....	87
Ilustración 100: Comparación del ToA relativo a la distancia al Main Gateway	87
Ilustración 101: Comparación de RSSI en función de SF	88
Ilustración 102: Comparación de SNR en función de SF	88
Ilustración 103: Comparación de la distribución de RSSI	89
Ilustración 104: Comparación de la distribución de SNR	89
Ilustración 105: Comparación de número de mensajes recibidos	90
Ilustración 106: Comparación de porcentaje de mensajes perdidos en relación a la distancia al Main Gateway	90
Ilustración 107: Comparación del porcentaje de la media acumulada de mensajes perdidos en relación a la distancia con el Main Gateway	91
Ilustración 108: Comparación de porcentaje de perdidas en función del SF.....	91
Ilustración 109: Diagrama de Gantt	102

Lista de tablas

Tabla 1: Resumen de parámetros a obtener de los devices	11
Tabla 2: Resumen de parámetros a obtener de los Gateways.....	13
Tabla 3: Resumen de parámetros que se van a utilizar para analizar la red.....	14
Tabla 4: Parámetros obtenidos listando los devices.....	15
Tabla 5: Parámetros obtenidos utilizando el endpoint de obtención de payloads	16
Tabla 6: Parámetros obtenidos listando los Gateways	17
Tabla 7: Parámetros avanzados obtenidos de la API del NST	18
Tabla 8: Resumen de métodos de obtención de parámetros	19
Tabla 9: Parámetros QoS configurables en el Network Server.....	71
Tabla 10: Nuevos valores de los parámetros QoS modificados (iteración).....	72
Tabla 11: Nuevos valores de los parámetros QoS modificados (final).....	83
Tabla 12: Sueldos y nóminas	103
Tabla 13: Amortizaciones	103
Tabla 14: Resumen de presupuesto.....	104

Lista de acrónimos

IoT	Internet of Things
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
CAV	Comunidad Autónoma Vasca
QoS	Quality of Service
DASS	Data Access Sub-System
NST	Network Supervision Tool
DevEUI	Device Extended Unique Identifier
RSSI	Received Signal Strength Indicator
SNR	Signal-to-Noise Ratio
SF	Spreading Factor
ToA	Time on Air
MAC	Medium Access Control
API	Application Programming Interface
JSON	JavaScript Object Notation
CABB	Consorcio de Aguas Bilbao Bizkaia
URL	Uniform Resource Locators
HTML	HyperText Markup Language
PNG	Portable Network Graphics
JPEG	Joint Photographic Experts Group
GUI	Graphical User Interface
EJIE	Eusko Jaurlaritzaren Informatika Elkarte
MAC	Media Access Control

1.Introducción

En estos últimos años, el auge del IoT (Internet of Things) ha sido indudablemente superior a otras tecnologías. Dentro de este ámbito del Internet de las Cosas, hay que destacar la tecnología LoRa (Long Range) [20], conocida por su alcance extendido y bajo consumo de energía. Esta ha emergido como una herramienta excepcionalmente eficaz y económica para la interconexión de dispositivos IoT. Estas características la vuelven especialmente apta para aplicaciones que requieren una cobertura amplia y una huella ecológica mínima, destacando su relevancia en la monitorización de recursos críticos.

Una red LoRaWAN (Long Range Wide Area Network) se compone principalmente de cuatro elementos, los cuales en conjunto hacen funcionar la red. Estos elementos serían los dispositivos IoT con módulos LoRa, los concentradores o Gateways, los servidores LoRaWAN o Network Servers y las aplicaciones finales para visualizar o utilizar los datos de los dispositivos IoT.

Es importante diferenciar entre LoRa y LoRaWAN [22], ya que, aunque ambos términos se utilizan en el ámbito de IoT, se refieren a aspectos diferentes pero complementarios. LoRa es la capa física que define la modulación de la señal y la transmisión a larga distancia con bajo consumo energético. En contraste, LoRaWAN es un protocolo de comunicación que se sitúa en la capa superior, gestionando la comunicación entre los dispositivos LoRa y los Gateways, así como el enrutamiento de los datos hacia los servidores de red. Mientras que LoRa proporciona el medio para la transmisión de datos, LoRaWAN se encarga de la gestión y el protocolo de red, facilitando funcionalidades como la seguridad, la gestión de dispositivos y la estructura de red escalable.

En la Ilustración 1 se puede apreciar un esquema de los elementos que componen una red IoT con tecnología LoRa. Entre ellos se encuentran los nodos finales o sensores, los concentradores o Gateways, el servidor de red o *Network Server* y los servidores de aplicación o *Application Server*.

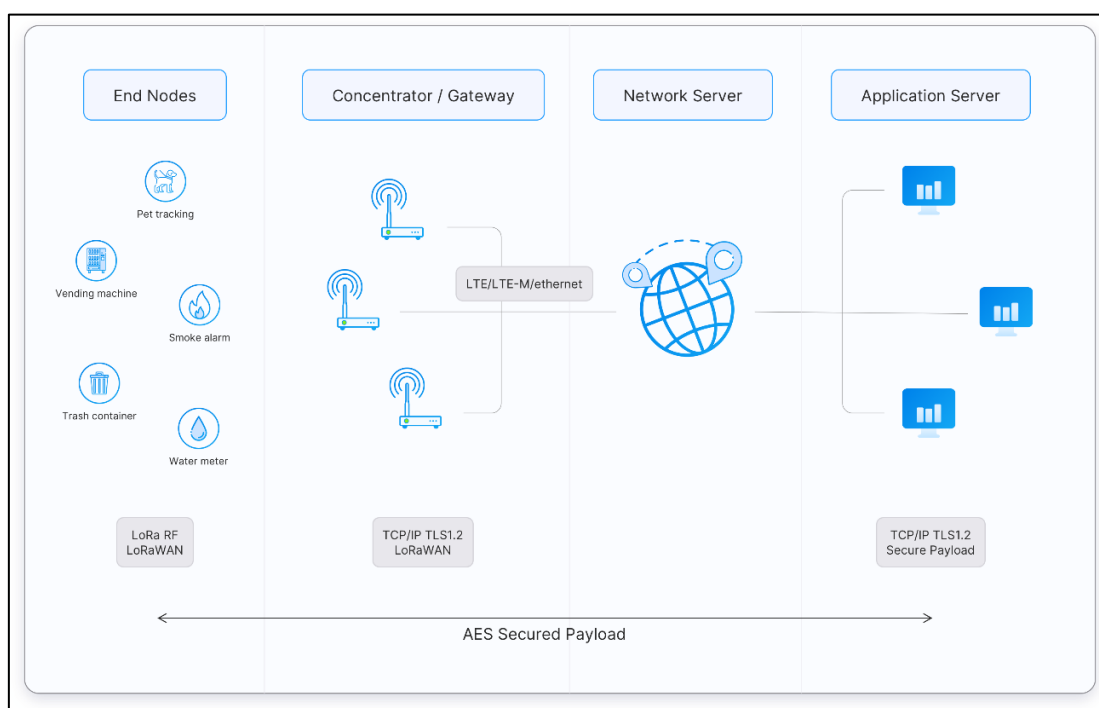


Ilustración 1: Arquitectura de red LoRa [1]

Dentro de esta nueva era del IoT entra en juego Itelazpi [21], una empresa pública vasca con sede en Zamudio, Bizkaia, dirigida a la prestación de servicios portadores de señales de radio y televisión y servicios de comunicaciones públicas. En los últimos cinco años, Itelazpi ha ido desplegado una red LoRa de envergadura en el entorno de la CAV (Comunidad Autónoma Vasca). Con alrededor de 200 Gateways y una asombrosa conectividad de más de 100.000 dispositivos IoT, esta red se ha convertido en un pilar fundamental para la gestión eficiente de recursos en la región.

Cabe destacar, que el mundo del metering [2] ha dado un salto tecnológico significativo gracias al IoT. Uno de los grandes beneficiados gracias al IoT y la tecnología LoRa es la monitorización de los contadores de agua, la cual ha facultado a las autoridades locales con la capacidad de recopilar datos en tiempo real sobre el consumo hídrico. Estos datos poseen un valor inestimable, ya que pueden ser empleados para optimizar la administración del agua, identificar fugas de manera proactiva y reducir el derroche de este recurso precioso.

La monitorización del agua mediante la red LoRa se erige como una herramienta imprescindible para garantizar la disponibilidad y la sostenibilidad de este recurso crítico en la región vasca. Esto logrará que sea más sencillo para las autoridades locales la recolección de datos precisos y en tiempo real sobre el consumo de agua.

En el País Vasco, la red LoRa desplegada por Itelazpi ha sido dedicada principalmente al metering, a los contadores de agua más concretamente. Para entender un poco la arquitectura de esta red, en la Ilustración 2 se pueden observar los elementos que componen esta red.

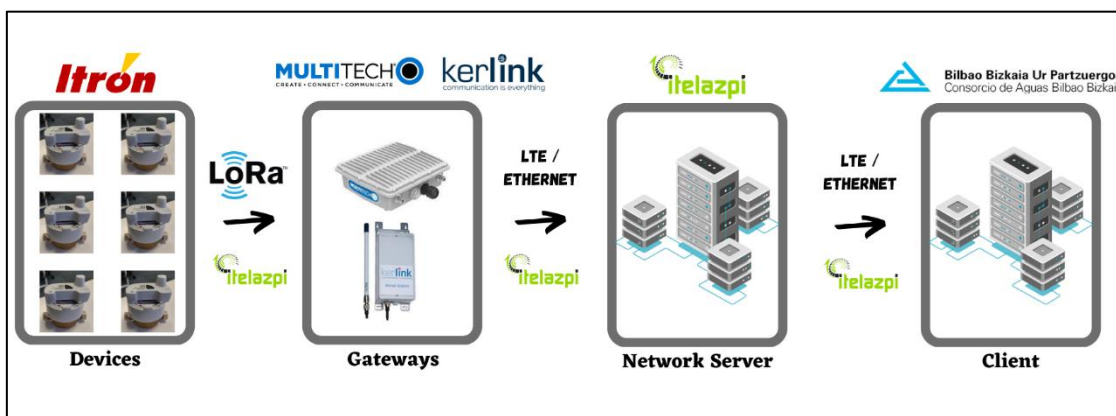


Ilustración 2: Diagrama de la red LoRa de Itelazpi

En la ilustración se pueden observar cuatro partes:

- **Devices:** Estos son los contadores de agua desplegados que transmiten datos a los Gateways mediante la tecnología LoRa.
- **Gateways:** Estos son los que dan cobertura y componen la red. Reciben datos de los devices y los envían al Network Server mediante conexión LTE o ethernet usando la red de transporte de Itelazpi.
- **Network Server:** Este sería el servidor central de la red, el cual se encarga de recibir todos los mensajes que le transmiten los Gateways y hacer la decodificación del protocolo LoRaWAN. Este se encuentra en el CPD de Itelazpi.
- **Client:** Este sería el cliente final, el que quiere recibir los datos de los devices. El Network Server es el encargado de reenviar los datos que le han llegado al cliente final mediante diversos protocolos (MQTTS, web socket, etc.).

2.Contexto

El contexto de este Trabajo de Fin de Master, TFM, se enmarca en unas prácticas de cooperación educativa en la ya mencionada empresa Itelazpi. Estas prácticas se han ido desarrollando a lo largo de más de dos años y medio, dando así una imagen más global de la empresa y facilidad para entender en profundidad lo relacionado a este TFM sin tener que empezar de cero. Aun así, el trabajo desarrollado durante este TFM, se ha realizado a lo largo de algo más de 5 meses. Gracias a este recorrido, se ha podido seguir la evolución de la red LoRa desplegada por Itelazpi desde una etapa muy temprana, adquiriendo progresivamente conocimientos sobre esta.

Actualmente, la red se encuentra en continua expansión por toda la CAV, gracias en parte a todos los dispositivos de metering nuevos que se han empezado a usar por diferentes entidades como pueden ser ayuntamientos y consorcios. Estos dispositivos componen más del 98% de los dispositivos conectados a la red LoRa, lo cual supone que la red está casi dedicada para estos dispositivos, teniendo que planearla y optimizarla para estos mismos.

Dada la importancia de la red LoRa de Itelazpi en la CAV, esta se despliega teniendo en cuenta criterios de eficiencia, sencillez, velocidad, etc. Es por ello que, Itelazpi, siendo una empresa pública con ciertas limitaciones, ha invertido recursos en una buena planificación de la red, así como una optimización posterior a la instalación y puesta en marcha en producción.

La parte de la planificación es algo relativamente sencillo de llevar a cabo, ya que se utilizan simuladores para poder visualizar la cobertura teórica que se conseguiría poniendo los Gateways en los lugares planeados. Esto facilita el trabajo a la hora de tener que añadir nuevos Gateways a la red para dar cobertura a más lugares, ya que se sabe de antemano donde llega y no llega la señal de los demás Gateways.

Hay que añadir que, la instalación de nuevos Gateways no es tan trivial como el añadir Gateways en las sombras de cobertura de los ya instalados. La densificación de la red LoRa de Itelazpi no es tan fácil por tres factores principales, los cuales se pueden ver en la Ilustración 3.

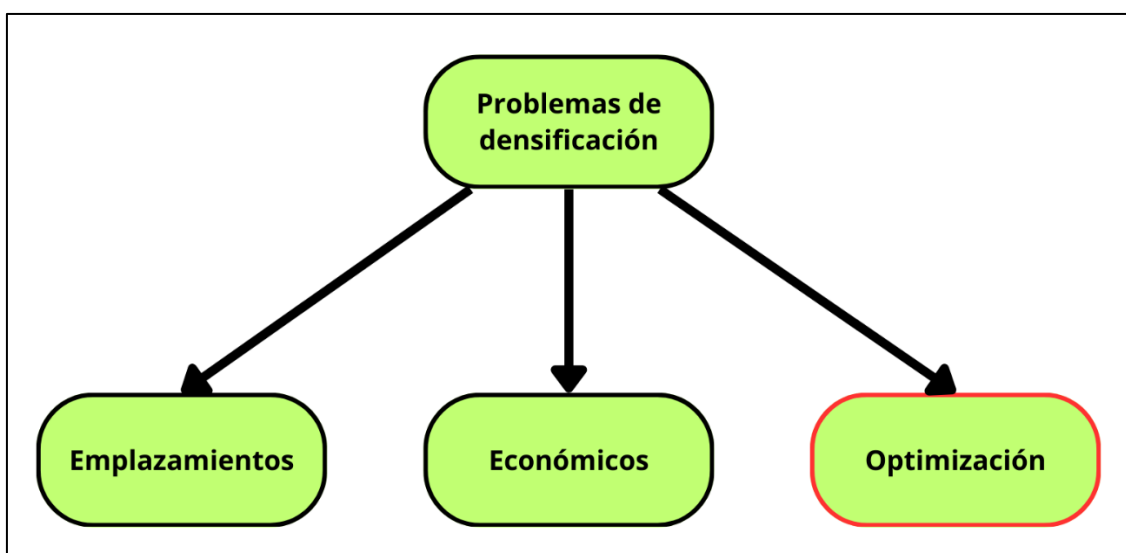


Ilustración 3: Esquema de problemas de densificación de la red LoRa de Itelazpi

El primero son los emplazamientos de los Gateways. Estos emplazamientos son limitados ya que Itelazpi al ser una empresa pública, tiene acceso a los edificios públicos (polideportivos, escuelas,

ayuntamiento...) que haya por la zona que se quiera dar cobertura, pero no cuenta con esas facilidades para los edificios privados (bloques de casas, empresas...).

El segundo factor sería el económico o el coste monetario. Instalar un nuevo Gateway conlleva una serie de gastos como la compra del Gateway y la instalación de este. Pero, por otro lado, también supone un gasto económico el dar de alta el Gateway en la red y poder operar con él en el Network Server, gestionando el tráfico que recibe, etc.

El tercer y último factor importante sería la parte de la optimización de la red ya instalada. Esta parte puede llegar a ser más complicada, ya que, una vez desplegada la red, cada dispositivo conectado se comporta de cierta manera, pudiendo afectar al resto de dispositivos.

Este trabajo pretende encontrar formas de mejorar la calidad de la red, y de los tres factores previamente mencionados, los dos primeros quedarían fuera del alcance de este trabajo, por lo que en la realización de este TFM se pretende optimizar la red LoRa ya desplegada mediante ajustes en los parámetros de QoS (Quality of Service) de la propia red.

Esta necesidad de optimizar la red surge del rápido crecimiento del número de dispositivos. Esto ha implicado que en ciertas zonas haya una alta densidad de dispositivos metering, causando que la calidad de la red baje ya que se producen colisiones en los mensajes en zonas con demasiados dispositivos.

Con el fin de minimizar la pérdida de paquetes en la red, se pretenden hacer ajustes de varios parámetros avanzados del protocolo LoRaWAN para comprobar si es posible mejorar la calidad de la red utilizando únicamente software (parámetros QoS) en vez de tener que utilizar hardware (más Gateways en nuevos emplazamientos).

Ya que la red es tan amplia y se extiende a lo largo de toda la CAV, se pretende acotar la zona de estudio a algún lugar que permita simular el comportamiento general y completo de la red, pero sin tener que trabajar con la cantidad de dispositivos y Gateways totales.

Como el estudio se llevará a cabo en una única zona de la red LoRa, pero la finalidad del proyecto es que sea escalable todas las demás zonas de la red, se creará una aplicación de escritorio capaz de llevar a cabo todo el procedimiento de una forma sencilla y lineal, facilitando así su uso futuro.

2.1. Zona de estudio

El lugar que se va a utilizar como zona de estudio será la localidad de Galdakao, Bizkaia, ubicado en la comarca del Gran Bilbao. Se ha elegido esta zona por varias razones, como pueden ser la localización, la cobertura LoRa y el número de dispositivos y Gateways instalados y funcionales.

La localización de Galdakao es excelente, ya que se ha buscado una zona la cual estuviese cerca de la sede de Itelazpi en Zamudio, por si fuese necesario desplazarse hasta la localización y hacer pruebas de campo.

En cuanto a la cobertura, un gran porcentaje de Galdakao está bajo la cobertura de la red LoRa de Itelazpi, por lo que los problemas de pérdidas de mensajes de dispositivos que se pueden dar es más probable que sean por mala calidad de la red y no por falta de cobertura.

Finalmente, el número de Gateways y dispositivos instalados es lo suficientemente alto como para poder asumir un comportamiento de la red que se expandirá al resto de esta. Galdakao cuenta con 16 Gateways y 13.000 dispositivos IoT (contadores de agua en su totalidad)

repartidos por todo el municipio, lo cual se puede considerar suficiente para hacer los diferentes análisis que se plantearan a lo largo del TFM. En la Ilustración 4 se pueden observar las ubicaciones de los Gateways en la zona de Galdakao. En la imagen se pueden apreciar 16 Gateways, 15 de ellos activos (verde) y uno de ellos apagado (rojo).

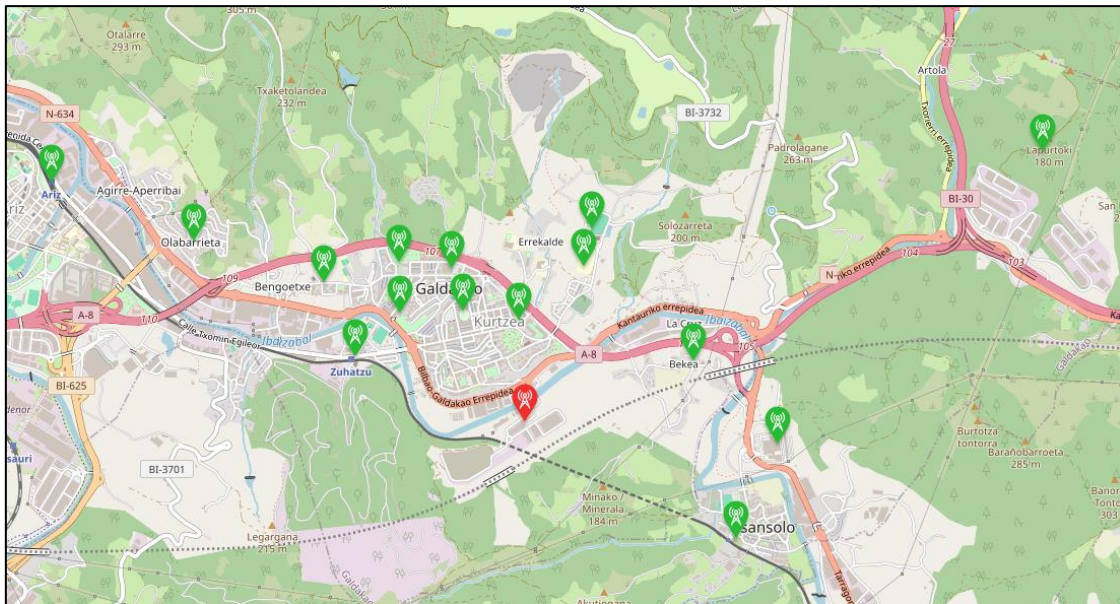


Ilustración 4: Ubicación de los Gateways en Galdakao

En resumen, este TFM va a buscar soluciones de optimización de la red LoRa desplegada por Itelazpi debido a la creciente demanda que se está viviendo por incorporar dispositivos IoT. Durante el proceso, se creará una aplicación de escritorio capaz de guiar al usuario por todo el proceso hasta obtener los resultados finales y que este solo tenga que encargarse de interpretarlos.

Esto se llevará a cabo en Galdakao como proyecto piloto con el fin de encontrar diferentes maneras de aumentar la eficiencia de la red LoRa haciendo uso exclusivo de los parámetros avanzados del protocolo LoRaWAN.

3. Objetivos y alcance del trabajo

El objetivo principal de este trabajo es consiste en la **optimización de la red IoT LoRa desplegada por Itelazpi mediante ajustes realizados en los parámetros avanzados de la red**. Esta optimización será aplicada en un área focalizada de la red (Galdakao) y con los resultados obtenidos y mediante la aplicación de escritorio desarrollada, se podrá expandir al resto de la red IoT.

Para poder llevar a cabo este objetivo, se plantean cinco objetivos secundarios, los cuales ayudaran a llevar a cabo una mejor realización del trabajo final.

- **Definición de parámetros importantes y de resultados a obtener para el análisis:** para realizar un análisis lo más completo posible de la red LoRa de Itelazpi, es indispensable definir los parámetros que serán necesarios para realizar dicho análisis. Además, también es necesario definir cuáles serán los resultados que se van a querer obtener, al igual que el formato en el que se presentarán.
- **Creación de la aplicación de escritorio:** con el fin de poder replicar este análisis en las diferentes zonas de la red de Itelazpi, se creará una aplicación de escritorio con interfaz de usuario para facilitar y simplificar los futuros análisis de la red.
- **Estudio previo a realizar cambios en la red:** antes de realizar ningún cambio en la red, será necesario analizar el estado actual de esta, con el fin de identificar cuáles pueden ser los puntos más débiles y respecto a estos programar una serie de cambios en los parámetros avanzados de la red.
- **Realización de cambios para mejorar el estado de la red:** para poder encontrar un punto óptimo de los parámetros avanzados de la red que se quieren cambiar, será necesario ir cambiándolos y algunos de estos serán prueba y error. Dependerá de cuan rápido se llegue al resultado esperado para saber el número de iteraciones que se realizarán.
- **Obtención de los parámetros óptimos para los resultados deseados:** una vez realizadas las iteraciones de prueba y error para encontrar los parámetros QoS adecuados, se asignarán estos parámetros y se hará un análisis final con el fin de ver si realmente se obtienen los resultados esperados.

4. Beneficios que aporta el trabajo

En este apartado se va a hacer mención a los beneficios económicos, técnicos y sociales que suponen este trabajo de fin de master.

4.1. Beneficios sociales

Este trabajo ayudara a optimizar la red LoRa de Itelazpi. Esta optimización no solo impulsará la eficiencia en la gestión de recursos críticos como puede ser el agua, sino que también redundará en beneficios significativos para las autoridades locales, como los ayuntamientos.

Esto se traducirá directamente en una mejora palpable en la gestión del agua en la Comunidad Autónoma Vasca. Este progreso se reflejará en una mayor disponibilidad de agua, en una reducción sustancial del desperdicio y, en última instancia, en el fortalecimiento de la sostenibilidad de este recurso crítico en la región.

4.2. Beneficios económicos

En cuanto a los beneficios económicos, estos se podrían dividir en dos partes, los beneficios de la empresa gestora de la red LoRa, Itelazpi, y los beneficios económicos que las empresas que gestionan los contadores de agua.

Por parte de Itelazpi, la optimización de la red LoRa supondrá un ahorro de gastos más que un mayor ingreso de capital. Esto se debe a que, optimizando la red, se pretende dar más cobertura y de mejor calidad utilizando los mismos Gateways, por lo que Itelazpi no tendrá que instalar nuevos Gateways, ahorrándose así los costes de compra, instalación, supervisión y mantenimiento de estos.

En el caso de los usuarios de la red, estos también podrán reducir costes con la optimización de la red, ya que los datos de los dispositivos tendrán una tasa de error mucho menor y la cobertura ofrecida será mayor, por lo que el gasto de personal en ir físicamente a los lugares donde están instalados estos dispositivos se puede recortar drásticamente.

4.3. Beneficios técnicos

El principal beneficio técnico de este trabajo es la resolución de los aspectos específicos que causan un peor desempeño de la red, impactando positivamente en la eficiencia local. Esto se debe a que el trabajo se ha realizado para la zona focalizada de Galdakao, por lo que los resultados obtenidos podrán ser visibles en esa zona.

Esto implica que, gracias a haber realizado el análisis en una zona en concreto, se ha obtenido una instantánea precisa y actualizada de la red en Galdakao. Pero, los resultados pueden servir para establecer bases sólidas para la toma de decisiones en futuras expansiones y ajustes de la red LoRa.

En términos generales, este trabajo ayudara a la futura planificación de la red LoRa al igual que a la optimización de la ya desplegada en otros lugares que no sean Galdakao, ya que el trabajo se ha realizado de la manera generalizada posible.

5. Descripción de la solución propuesta

En este apartado del trabajo se van a detallar los pasos seguidos durante el transcurso para la elaboración de este. Estos pasos se explicarán de una forma ordenada tal y como se organizó el TFM desde un principio.

Siguiendo el esquema de planificación (Ilustración 5), este apartado se va a dividir para describir el trabajo, relacionado con los cinco objetivos secundarios. Primero se explicarán las diferentes definiciones que se han hecho, tanto de los parámetros que se van a utilizar como de los resultados que se pretende obtener.

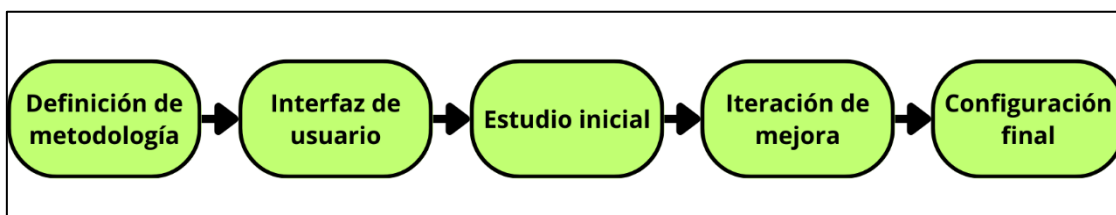


Ilustración 5: Esquema de planificación del proyecto

A continuación, se explicará todo lo relacionado con la aplicación de escritorio que se ha creado, empezando desde los scripts independientes más básicos hasta llegar a la aplicación final la cual aúna todo lo previamente creado.

Después se explicará el estudio inicial que se ha hecho, antes de realizar ningún cambio a la red, para así poder observar el estado de esta y obtener conclusiones respecto a que parámetros podría ser necesario cambiar.

Posteriormente, se explicará la iteración de mejora que se ha realizado con el fin de encontrar los valores óptimos de los parámetros avanzados de la red que se han decidido cambiar. Esta parte es totalmente escalable en cuanto a número de iteraciones y se podría repetir las veces necesarias hasta obtener los resultados esperados. Pero, debido al limitado tiempo del TFM, se ha hecho una única iteración.

Por último, en la configuración final se realizará un análisis con los valores finales de los parámetros avanzados de la red que se han modificado, para asegurar así el correcto funcionamiento de la red y las mejoras respecto al estado inicial de esta.

5.1. Metodología de captura de datos y resultados

Este primer apartado del trabajo es fundamental, ya que en él se definen los aspectos significativos del TFM. Al establecer las bases y los objetivos, se orienta el desarrollo del resto del trabajo.

Primero se han definido los parámetros que pueden resultar importantes para este trabajo, los cuales ayudaran a entender mejor la red y visualizar como esta cambia a través de estos parámetros.

Una vez teniendo definidos que parámetros se necesitan para hacer un buen análisis de la red, se definen los métodos de extracción de estos de los diferentes lugares donde están guardados.

Por último, sabiendo cuales van a ser los parámetros que se van a utilizar para analizar la red, se definen diversos gráficos para poder visualizar estos parámetros de una forma sencilla y clara.

Esto también incluye mezclar diferentes parámetros con el fin de visualizar los datos de una manera más compacta.

En resumen, en este bloque se explicarán en profundidad los siguientes subapartados:

- Definición de parámetros
- Definición de métodos de captura de datos
- Definición de gráficos a obtener

5.1.1. Definición de parámetros

Como primer punto de este trabajo es importante definir cuáles son los parámetros que se van a utilizar para definir la calidad de la red LoRa de Itelazpi. Estos parámetros se pueden obtener del *Network Server* con el cual se gestiona toda la red IoT. El *Network Server* utilizado por Itelazpi es Orbiwise [3], una plataforma pensada para proyectos a gran escala y con muchas funcionalidades útiles para llevar a cabo diferentes tareas.

Orbiwise está dividido en dos partes, una para la gestión de los devices y otra para la gestión de los Gateways, DASS (Data Access Sub-System) y NST (Network Supervision Tool) respectivamente. Como para el completo análisis de la red es necesario obtener datos tanto de los devices como de los Gateways, se hará uso de ambas instancias con el fin de obtener el mayor número de parámetros útiles.

Parámetros de los devices

Haciendo uso de la interfaz gráfica de Orbiwise, se pueden analizar los parámetros disponibles de los dispositivos. Para empezar, en la información básica de los dispositivos se pueden encontrar como parámetros útiles el DevEUI (identificador único de cada dispositivo) y la posición de este (latitud y longitud). Esta información se puede ver en la Ilustración 6, señalados únicamente los parámetros mencionados.

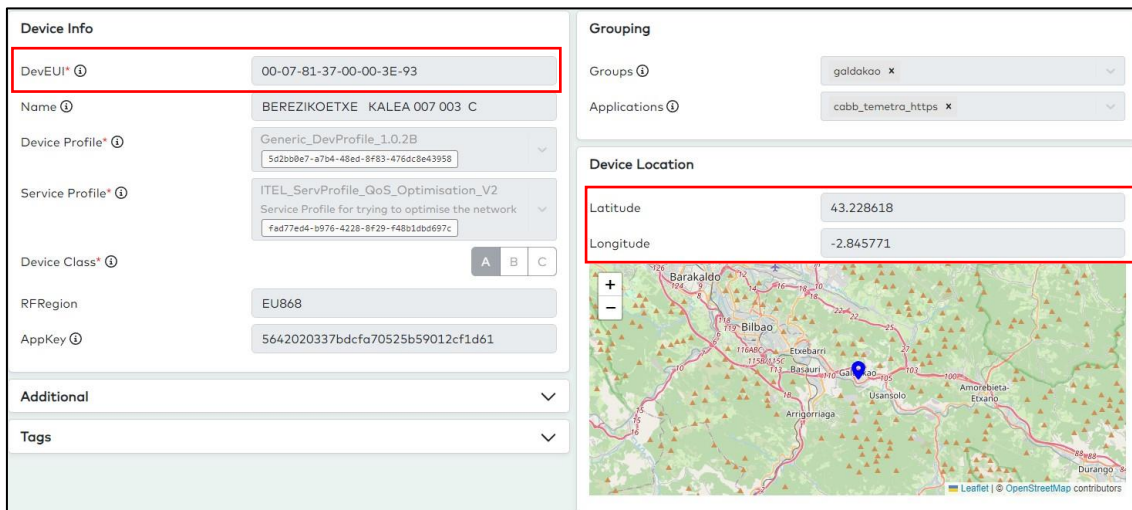


Ilustración 6: Información básica de un device en Orbiwise

Por otro lado, accediendo a información más detallada de los devices, se pueden encontrar parámetros relacionados con la calidad de la red, como pueden ser el RSSI, SNR, etc. De estos parámetros que se pueden observar en la Ilustración 7, se hará uso del RSSI (Received Signal Strength Indicator), del SNR (Signal-to-Noise Ratio), el SF (Spreading Factor), el Main Gateway (Gateway principal) y el número de Gateways al que llegan los mensajes del dispositivo.

DevEUI	Device comment	DevAddr	Tenant	SF	Lst RX	Lst RSSI	Lst SNR	Main Gateway	#Gtwys	Main Gtwy NM
0007813700003e93										
0007813700003e93	BEREZIKOETXE KALEA 007 003 C	113FD053	admin	SF11BW125	23/4/2024 9:11:41	-112	-7	000800fffe4bc872	5	Galdakao ETS ...

Ilustración 7: Información detallada de un device en Orbiwise I

De estos parámetros, el RSSI y el SNR se utilizarán para averiguar con que potencia de señal llegan los mensajes que los devices envían. El SF se utilizará para ver el modo en el que se transmiten los datos. El Main Gateway será útil para ver que Gateways tienen más carga de trabajo, y finalmente, el número de Gateways al que llega cada mensaje será un indicativo de la redundancia de la red.

Continuando con la información detallada de los devices, también es posible seleccionar rangos de tiempo y obtener información única y exclusivamente de ese rango definido, además de medias y porcentajes. En la Ilustración 8 se pueden observar diferentes parámetros los cuales también son realmente útiles para realizar un análisis de la calidad de la red.

DevAddr	DevEUI	UL Frame Cnt	ULD Cnt	UL bytes	UL Duty Cycle [%]	DL Frame Cnt	DL bytes	DL Duty Cycle [%]	Lost Frame Cnt	FER [%]	Avr RSSI[dBm]	Avr SNR[dB]	Used SF	Last Seen
113FD053	0007813700003E93	4	0	239	0.007	0	0	0.000	0	0.000	-112.0	-4.7	11	09:11:41.040 23/04/2024

Ilustración 8: Información detallada de un device en Orbiwise II

Entre ellos se encuentran el número de mensajes recibidos en el *Network Server* del dispositivo (UL Frame Cnt) y las medias en el rango de tiempo seleccionado de los parámetros RSSI y SNR (Avr RSSI[dBm] y Avr SNR[dB]).

Además, partiendo del dato del número de mensajes que han llegado a Orbiwise y analizando cual es el número máximo de mensajes que deberían de haber llegado, se puede obtener el número de mensajes perdidos (Lost Frames).

Finalmente, existe un parámetro que en la interfaz web de Orbiwise no se muestra, pero es igual de importante para medir la calidad de la red como el resto. Este parámetro es el tiempo en el aire (Time on Air, ToA) de cada mensaje, ya que la tecnología LoRa al utilizar una banda no licenciada [4], el tiempo que se puede ocupar la banda es limitado, por lo que es interesante controlar este parámetro.

En la Ilustración 9, se puede observar que el parámetro ToA aparece en los mensajes de la capa MAC (Medium Access Control) que envían los dispositivos.

Message
<pre> major=0, FCtrl:80 [ADR], Len:61, toa:1.479s, v:1.0.2B, [MicOk] rg:EU868 Encrypted: 40 53 D0 3F 11 80 08 0E 0T AA 5C 9C 47 15 0D 6A 75 0E 90 3C 5A F6 32 79 60 64 B6 79 73 9C 78 DC A2 06 F1 52 10 14 3B 6D 84 C4 BC 40 96 00 E8 8C 10 A8 6F 82 34 CD 0B BA F0 A0 5D 3F 6E </pre>

Ilustración 9: Mensaje MAC enviado por un dispositivo

En resumen, los parámetros que es necesario obtener de cada device quedan listados en la Tabla 1, con su nombre para su uso y su definición breve.

Nombre	Definición
deveui	Identificador único del device
latitude	Latitud para su ubicación
longitude	Longitud para su ubicación
average_rssi	Media de RSSI en el tiempo analizado
average_snr	Media de SNR en el tiempo analizado
average_time_on_air_ms	Media de ToA en el tiempo analizado
average_gws_reached	Media de Gateways alcanzados en el tiempo analizado
main_gw	Gateway principal del device
lost_frames	Numero de mensajes no recibidos en Orbiwise
received_frames	Numero de mensajes recibidos en Orbiwise
last_SF_used	SF usado en el último mensaje recibido

Tabla 1: Resumen de parámetros a obtener de los devices

Parámetros de los Gateways

Al igual que con los devices, los parámetros que se pueden ser útiles para hacer un análisis de la red pueden encontrarse en la interfaz web de Orbiwise. Primero de todo, observando la información básica de un Gateway, se pueden encontrar los parámetros de Gateway ID (identificador único de cada Gateway) y Name (Nombre asignado al Gateway para reconocerlo fácilmente), tal y como se puede observar en la Ilustración 10.

Gateway Information

Gateway ID	000800fffe4bc870
Name	Galdakao Olabarrieta S/N: 21713698
Gateway type	Multitech Conduit IP, 863-870MHz
Firmware version	HAL v5.0.1-mts-5 PktFwd v4.0.1-mts-7
Software version	v6.2.0-1
Region	EU868
tags	Multitech x Production x Galdakao x OutdoorGW x Bizkaia x EU868 x

Update Name and Tags

Ilustración 10: Información básica de un Gateway en Orbiwise I

Además de esta información básica, también se pueden encontrar las coordenadas (latitud y longitud) de donde está instalado el Gateway (Ilustración 11), al igual que pasaba con los devices.

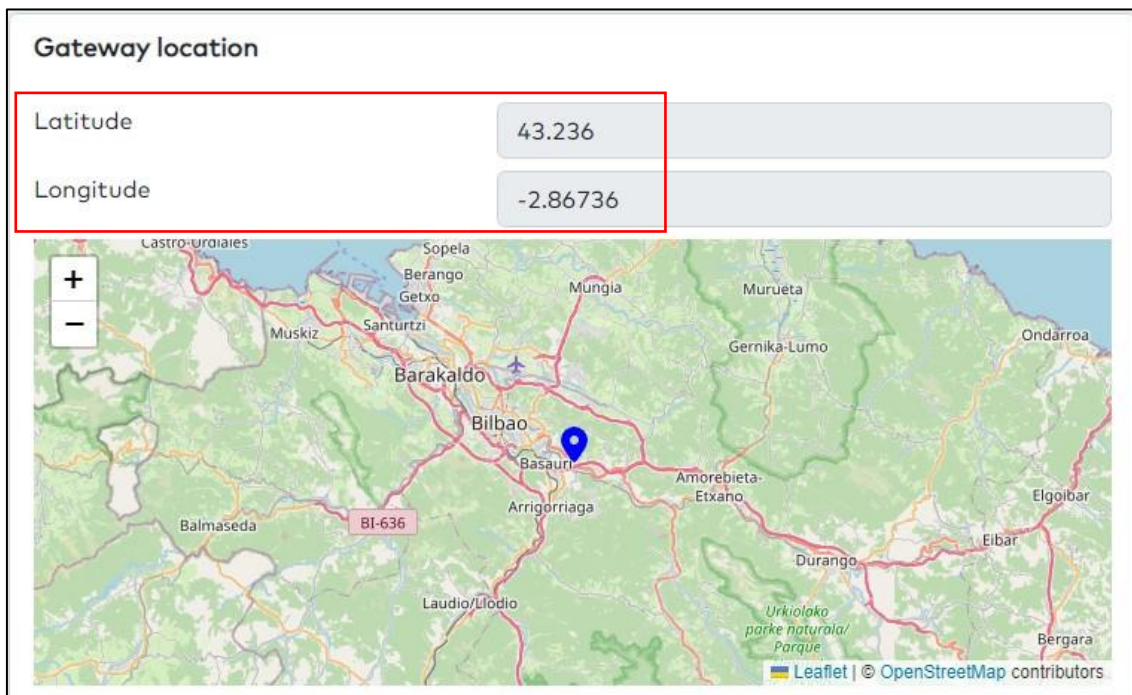


Ilustración 11: Información básica de un Gateway en Orbiwise II

Accediendo a información más detallada de cada Gateway, se pueden encontrar parámetros directamente relacionados con el intercambio de mensajes. Entre ellos destacan UL Frame Cnt (Uplink Frame Count), Lost Frame Cnt (Lost Frame Count), Avr RSSI [dBm] (Average RSSI [dBm]) y Avr SNR [dB] (Average SNR [dB]). Estos se pueden encontrar en la Ilustración 12.

Gateway ID	Gateway Name	UL Frame Cnt	ULD Cnt	UL Bytes	DL Frame Cnt	DL Bytes	Lost Frame Cnt	FER [%]	Avr RSSI [dBm]	Avr SNR [dB]	Last Seen
000800FFFE4BC870	Galdakao Olabarrieta S/N: 21713698	233	0	12545	6	97	0	0.000	-117.2	-6.6	16:00:56.774 24/04/2024

Export Table as CSV Export Table as JSON

Ilustración 12: Información detallada de un Gateway en Orbiwise

De los parámetros mencionados, UL Frame Cnt será útil para saber cuántos mensajes recibe un Gateway y ver si este está saturado respecto a los demás. Lost Frame Cnt se utilizará para averiguar si el Gateway tiene pérdida de paquetes, esto es, que haya recibido el mensaje de un device pero que por alguna razón no lo haya podido retransmitir al *Network Server*. Por otro lado, Avr RSSI y Avr SNR servirán para ver la potencia media de los mensajes recibidos en el Gateway en el rango de tiempo previamente establecido.

Resumiendo, los parámetros que es necesario obtener de cada Gateway quedan listados en la Tabla 2, con su nombre para su uso y su definición breve.

Nombre	Definición
Gateway_ID	Identificador único del Gateway
name	Nombre asignado al Gateway para su fácil reconocimiento
latitude	Latitud para su ubicación
longitude	Longitud para su ubicación
UL_Frame_Cnt	Numero de mensajes recibidos en el tiempo analizado
Lost_Frame_Cnt	Numero de mensajes perdidos en el tiempo analizado
Avr_RSSI_[dBm]	Media de RSSI en el tiempo analizado
Avr_SNR_[dBm]	Media de SNR en el tiempo analizado

Tabla 2: Resumen de parámetros a obtener de los Gateways

Resumen de parámetros

Con el fin de visualizar mejor los parámetros que se van a utilizar para hacer el análisis de la red, se han resumido en la Tabla 3, donde aparece el nombre, la definición y la asociación del parámetro a un device o un Gateway.

Además, se han añadido nuevos parámetros que únicamente se pueden obtener utilizando tanto información de los devices como de los Gateways. Estos parámetros son el Gateway más cercano a cada device y las distancias respectivas entre el device y el Main Gateway y el Gateway más cercano.

Nombre	Definición	Asociado
deveui	Identificador único del device	device
latitude	Latitud para la ubicación del device	device
longitude	Longitud para la ubicación del device	device
average_rssi	Media de RSSI de mensajes del device en el tiempo analizado	device
average_snr	Media de SNR de mensajes del device en el tiempo analizado	device
average_time_on_air_ms	Media de ToA de mensajes del device en el tiempo analizado	device
average_gws_reached	Media de Gateways alcanzados por device en el tiempo analizado	device
lost_frames	Numero de mensajes no recibidos en Orbiwise por device	device
received_frames	Numero de mensajes recibidos en Orbiwise por device	device
last_SF_used	SF usado en el último mensaje enviado por el device	device
main_gw	Gateway principal del device	device
distance_to_main_gw	Distancia entre el device y el Main Gateway	device

nearest_gw	Gateway más cercano al device	device
distance_to_nearest_gw	Distancia entre el device y el Gateway más cercano	device
Gateway_ID	Identificador único del Gateway	Gateway
name	Nombre asignado al Gateway para su fácil reconocimiento	Gateway
latitude	Latitud para la ubicación del Gateway	Gateway
longitude	Longitud para la ubicación del Gateway	Gateway
UL_Frame_Cnt	Numero de mensajes recibidos en el tiempo analizado por Gateway	Gateway
Lost_Frame_Cnt	Numero de mensajes perdidos en el tiempo analizado por Gateway	Gateway
Avr_RSSI_[dBm]	Media de RSSI de mensajes recibidos por Gateway en el tiempo analizado	Gateway
Avr_SNR_[dBm]	Media de SNR de mensajes recibidos por Gateway en el tiempo analizado	Gateway

Tabla 3: Resumen de parámetros que se van a utilizar para analizar la red

5.1.2. Definición de los métodos de captura de parámetros

Con el fin de poder hacer uso de los parámetros ya definidos para realizar el análisis de la red LoRa, será necesario establecer los métodos con los que se van a obtener esos datos, para así poder tener acceso a estos de una manera en la que se pueda automatizar.

Para ello, Orbiwise cuenta con una API (Application Programming Interface) y una extensa documentación la cual será posible consultar con el fin de averiguar cómo obtener los parámetros solicitados.

Al igual que el apartado anterior, en este también se diferenciará la obtención de los parámetros de los devices y de los Gateways.

Parámetros de los devices

Analizando la documentación de la API de Orbiwise, se encuentra un método para obtener una lista de todos los devices pertenecientes a un grupo (Ilustración 13). Itelazpi organiza los dispositivos por grupos, y una de las directrices para asignar grupos es el municipio en el que está instalado el sensor. Por lo que, para el caso de este trabajo, el grupo que se va a utilizar el “galdakao”.

4.4.4. Get List of Devices	
Get list of registered devices with the device info for each device.	
URL: https://host[:port]/rest/nodes[?group=groupid][&application=appid1]	
Method:	GET
Direction:	Application->DASS

Ilustración 13: Endpoint de la API de Orbiwise para obtener todos los devices de un grupo

Con este endpoint se obtienen diversos parámetros de cada dispositivo, pero de los definidos previamente, únicamente 3, los cuales se pueden ver en la Tabla 4. En la tabla, {host} hace

referencia a la dirección de Orbiwise y puede ser tanto la dirección privada como la pública, y {groupid} hace referencia al nombre del grupo que se va a utilizar, en este caso "galdakao".

Parámetro	Endpoint
deveui	https://{host}/rest/nodes?group={groupid}
latitude	https://host/rest/nodes?group=groupid
longitude	https://host/rest/nodes?group=groupid

Tabla 4: Parámetros obtenidos listando los devices

Una vez habiendo obtenido una lista de todos los devices de los que se va a hacer el análisis, usando el siguiente endpoint (Ilustración 14) se pueden obtener los mensajes (payloads) que ha enviado cada dispositivo.

4.5.1. Receive all Pending Uplink Payloads from Device

Receive an array with all received payloads for a device. The payloads are stored in the persistent temporary storage inside OrbiWAN. The application should delete the payload once they have been safely "consumed". Unread payloads will be automatically deleted after an expiry period.

URL: `https://host[:port]/rest/nodes/{deveui}/payloads/ul[?query_options]`

Method: GET

Direction: Application->DASS

Ilustración 14: Endpoint de la API de Orbiwise para obtener los payloads de un device

Además, este endpoint permite añadir diferentes opciones para filtrar los payloads. Entre ellas, las más interesantes son las de filtrar por fecha, ya que el análisis se hará en un rango de tiempo determinado. En la Ilustración 15 se pueden observar algunas de las opciones disponibles, y marcadas en rojo, las que se van a utilizar.

The valid queries that can be used with this API are:

Parameter	values	description
data_format	hex,base64	Data format for all binary data. Default is base64
from_date	ISO8601 date/time	show only payload from after this time/date
to_date	ISO8601 date/time	show only payload from before time/date
get_pages	boolean	see section on paged queries

Ilustración 15: Extracto de las opciones disponibles del endpoint de obtención de payloads

Utilizando este endpoint, se obtendrá como resultado la información de la Ilustración 16 para cada device. De esta información obtenida, se harán cálculos para obtener ciertos parámetros que no se consiguen directamente, sino que es necesario realizar algunas operaciones, como puede ser el caso de obtener las medias de RSSI, SNR, ToA, etc.

```

[
  {
    "dataFrame": "AB==", // raw (encrypted) payload in base64 format
    "confirmed": false, // confirmed(true) or unconfirmed(false) uplink
    "port": 1, // MAC port the message was receive on
    "timestamp": "2015-02-11T10:33:00.578Z", // time of reception in GMT
    "fcnt": 138, // uplink FCNT (needed for decryption)
    "rssi": -111, // RSSI from gateway
    "snr": -6, // SNR from gateway
    "sf_used": 8, // used spreading factor - Depreciated,
    // don't use in new code.
    // please use dr_used field instead.
    "dr_used": "SF8BW128", // used data-rate, SF and BW
    "cr_used": "4/5", // Forward error correction code used
    "device_redundancy": 1, // number of times the device has transmitted
    // the burst
    "time_on_air_ms": 41.216, // Airtime for this burst [ms]
    "id": 278998, // unique identifier (64-bit) of payload.
    // Needed to delete the payload
    "session_id": "0f9aaa82-b16c-4c2d-9b99-344e2f2c8b7d", // session ID under which the
    packet was received
    "decrypted": true, // set true if the DASS decrypted the payload,
    // false if the message is still encrypted.

    "gtw_info": [ // see note below.
      {gtw_id: "0000000012340000", rssi: -100, snr: 5 },
      {gtw_id: "0000000012350000", rssi: -90, snr: 15 },
      ...
    ],

    "push_status": [ // only available with push_status=true option.
      { "app": "app-id1", "app_uuid": "uuid-of-app1", status: 200 }
      { "app": "app-id2", "app_uuid": "uuid-of-app2", status: 404 }
      ...
    ],

    "decoded": { // only available with associated payload decoder
      // user defined object from payload decoder
    }
  },
  ...
]
  
```

Ilustración 16: Respuesta obtenida de una llamada al endpoint de obtención de payloads

En la Tabla 5 se resume los parámetros que se obtienen de este endpoint, además de si estos son obtenidos directamente o es necesario realizar ciertas operaciones. En la tabla {deveui} hace referencia al identificador único de cada device, por lo que por cada device que se vaya a analizar, se realizara una llamada al endpoint. Por otro lado, {from_date} hace referencia a fecha y hora de inicio del análisis, y {to_date} a la del final del análisis.

Parámetro	Endpoint	Tipo
average_rssi	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Indirecto
average_snr	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Indirecto
average_time_on_air_ms	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Indirecto
average_gws_reached	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Indirecto
main_gw	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Indirecto
lost_frames	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Indirecto
received_frames	https://{host}/rest/nodes{deveui}/payloads/ul?{query_options}	Directo
last_SF_used	https://{host}/rest/nodes{deveui}/payloads/ul?{query_options}	Directo

Tabla 5: Parámetros obtenidos utilizando el endpoint de obtención de payloads

Parámetros de los Gateways

En lo que respecta a los Gateways, la API también incluye un endpoint para poder obtener una lista de todos los Gateways asociados a una cuenta, como se puede observar en la Ilustración 17.

4.14.1. Get List with Information about Gateways

Get list with information about all gateways associated to an account.

URL: `https://dass-host[:port] /rest/gateways`

Method: GET

Direction: application->DASS

Ilustración 17: Endpoint para obtener todos los Gateways asociados a un usuario

Itelazpi tiene todos los Gateways asociados a un mismo usuario, ya que, al ser una red entera y no proyectos separados, es imposible separar los Gateways por diferentes clientes. Esto significa que este endpoint devolverá una lista de todos los Gateways de Itelazpi y que habrá que seleccionar los deseados para el análisis de una forma manual.

En cuanto al endpoint para obtener una lista de todos los Gateways, en la Ilustración 18, sacada de la documentación, se puede observar los parámetros que devuelve.

A JSON payload with the following content is returned:

```

[
  {
    "id": "0000000008050313",
    "name": "313 Labo",
    "address": {},
    "latitude": 46.21025,
    "longitude": 2.922363,
    "antenna_gain": 3,
    "position_valid": true,
    "status": "OK",
    "backhaul_type": "WIRED",
    "backhaul_cell_operator": "",
    "backhaul_cell_rssi": "N/A"
  },
  ...
]
```

Ilustración 18: Respuesta obtenida de una llamada al endpoint de listado de Gateways

De todos estos parámetros, en la Tabla 6 se resumen los útiles para el análisis previamente definido.

Parámetro	Endpoint
id	<code>https://{host}/rest/gateways</code>
name	<code>https://{host}/rest/gateways</code>
latitude	<code>https://{host}/rest/gateways</code>
longitude	<code>https://{host}/rest/gateways</code>

Tabla 6: Parámetros obtenidos listando los Gateways

Como ya se ha mencionado antes, Orbiwise está dividido en dos instancias, DASS y NST. Hasta ahora, todos los datos se han obtenido del DASS, pero para la información detallada de los Gateways, es necesario utilizar la API del NST.

Esta API no es tan extensa como la otra, por lo que no cuenta con tanta documentación. Además, esta API sigue en desarrollo al momento de hacer el trabajo, por lo que no está totalmente pulida.

En la Ilustración 19 se puede ver el endpoint que se va a utilizar para conseguir los parámetros faltantes de los Gateways, los cuales no se han podido obtener con la otra API.

Getting aggregated Gateways stats - `getGWStatsAggregated`

Sample Request

```
/rest?tz=-330&from=17%2F12%2F2017%2018%3A33%3A34&to=20%2F12%2F2017%2016%3A33%3A38&getGWStatsAggregated=1
```

Set in the request, set `getGWStatsAggregated = 1`

Ilustración 19: Endpoint de la API NST para obtener parámetros avanzados de un Gateway

Y en la siguiente imagen (Ilustración 20), se pueden analizar los parámetros que devuelve esta llamada al endpoint.

Result format			
Property	Description	Property	Description
gw	gateway	ts	timestamp of the slot- 12:00, 12:30, 1:00 etc on GMT
dl	downlink counts	ul	uplink counts
dlB	downlink bytes	ulB	uplink bytes
aRSSI	average RSSI	aSNR	average SNR
dlToa	downlink time on air	ulToa	uplink time on air
sf	SF used	lf	lost frames

Ilustración 20: Formato de respuesta de una llamada al endpoint de obtención de parámetros avanzados de un Gateway

De todos los parámetros devueltos, en la Tabla 7 se puede ver un resumen de los parámetros que se van a utilizar y el endpoint que se va a usar.

Parámetro	Endpoint
ul	https://{host}/tfw1/rest?getGWStatsAggregated=1&from={from_date}&to={to_date}&gw={gw_id}
lf	https://{host}/tfw1/rest?getGWStatsAggregated=1&from={from_date}&to={to_date}&gw={gw_id}
aRSSI	https://{host}/tfw1/rest?getGWStatsAggregated=1&from={from_date}&to={to_date}&gw={gw_id}
aSNR	https://{host}/tfw1/rest?getGWStatsAggregated=1&from={from_date}&to={to_date}&gw={gw_id}

Tabla 7: Parámetros avanzados obtenidos de la API del NST

Resumen de parámetros

Con el fin de unificar todo lo descrito en este apartado, se ha resumido en la Tabla 8 los parámetros que se van a obtener, con sus respectivos endpoints e información adicional. Además, se han añadido también los parámetros que no se consiguen de un único endpoint, sino que es necesaria la información de ubicación tanto de los devices como de los Gateways.

Parámetro	Endpoint	Gateway/device	Tipo
deveui	https://{host}/rest/nodes?group={groupid}	Device	Directo
latitude	https://{host}/rest/nodes?group={groupid}	Device	Directo
longitude	https://{host}/rest/nodes?group={groupid}	Device	Directo
average_rssi	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Device	Indirecto
average_snr	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Device	Indirecto
average_time_on_air_ms	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Device	Indirecto
average_gws_reached	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Device	Indirecto
main_gw	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Device	Indirecto
distance_to_main_gw	-----	Device	Indirecto
nearest_gw	-----	Device	Indirecto
distance_to_nearest_gw	-----	Device	Indirecto
lost_frames	https://{host}/rest/nodes{deveui}/payloads/ul?from_date={from_date}&to_date={to_date}	Device	Indirecto
received_frames	https://{host}/rest/nodes{deveui}/payloads/ul?{query_options}	Device	Directo
last_SF_used	https://{host}/rest/nodes{deveui}/payloads/ul?{query_options}	Device	Directo
id	https://{host}/rest/gateways	Gateway	Directo
name	https://{host}/rest/gateways	Gateway	Directo
latitude	https://{host}/rest/gateways	Gateway	Directo
longitude	https://{host}/rest/gateways	Gateway	Directo
ul	https://{host}/tfw1/rest?getGWStatsAggregated=1&from={from_date}&to={to_date}&gw={gw_id}	Gateway	Directo
lf	https://{host}/tfw1/rest?getGWStatsAggregated=1&from={from_date}&to={to_date}&gw={gw_id}	Gateway	Directo
aRSSI	https://{host}/tfw1/rest?getGWStatsAggregated=1&from={from_date}&to={to_date}&gw={gw_id}	Gateway	Directo
aSNR	https://{host}/tfw1/rest?getGWStatsAggregated=1&from={from_date}&to={to_date}&gw={gw_id}	Gateway	Directo

Tabla 8: Resumen de métodos de obtención de parámetros

5.1.3. Definición de gráficos a obtener

Una vez habiendo definido los parámetros que se van a obtener y usar para realizar el análisis de la red, y confirmar los métodos de obtención de estos, se definirán los gráficos que se van a crear a partir de estos parámetros.

Esto servirá para que el usuario que está realizando el análisis pueda realizar este de una manera más visual, sin tener que estar leyendo datos numéricos y poder centrarse en las causas y efectos de estos valores.

En cuanto a los gráficos a obtener, se van a dividir en 4 categorías similares y una diferente. Las cuatro categorías similares serán gráficos de barras que en cada categoría cambiara el eje X que se va a usar para medir los parámetros, y dentro de cada una de estas categorías, como eje Y se utilizaran diferentes parámetros.

En cuanto a la categoría diferente, esta no serán gráficos de barras, sino mapas para visualizar la zona analizada a vista de pájaro.

Volviendo a las 4 categorías similares, estas serán las siguientes:

- **Gateways:** el eje X serán los Gateways que se han seleccionado para el análisis.
- **Distribuciones:** esta categoría serán distribuciones de los resultados, por lo que, dependiendo del parámetro que se quiera medir, el eje X cambiará.
- **Distancias al Main Gateway:** en este caso, el eje X será distancia, y se clasificaran los datos dependiendo de la distancia a la que este el device del Main Gateway.
- **Último SF utilizado:** por último, en esta categoría el eje X será el Spreading Factor, por lo que se clasificaran los parámetros dependiendo del último SF que haya usado el device.

A continuación, se describirán los gráficos que se van a obtener dentro de cada categoría, haciendo un total de 24 gráficos entre las 5 categorías ya mencionadas.

- **Mapas**
 - **Gateways y devices:** en este mapa se podrán visualizar todos los devices y Gateways seleccionados para el análisis, con el fin de visualizar la distribución geográfica de la zona.
- **Gateways**
 - **Media de RSSI por Gateway:** en este grafico se podrá observar la media de RSSI de todos los mensajes recibidos por cada Gateway.
 - **Media de SNR por Gateway:** este otro será igual que el anterior, pero visualizando el SNR en vez del RSSI.
 - **Mensajes perdidos por Gateway:** en este grafico se podrán analizar los paquetes perdidos por Gateway.
 - **Mensajes Recibidos por Gateway:** en este otro, igual que el anterior, pero con los paquetes recibidos.
- **Distribuciones**
 - **Media de Gateways alcanzados:** en este grafico se visualizará cual ha sido la media de Gateways alcanzados por los mensajes de cada device en la franja de tiempo seleccionada.

- **Último SF usado:** en este otro gráfico, el eje X serán los diferentes SF utilizados por los devices, y así se mostrará el número de devices que están utilizando cada SF.
- **Mensajes perdidos:** en este, se podrá ver el número de devices que ha perdido X mensajes, donde X puede ir desde 0 hasta el número máximo de mensajes enviados en el intervalo de tiempo seleccionado.
- **Mensajes recibidos:** este grafico será igual que el anterior, pero utilizando los mensajes recibidos en vez de los perdidos.
- **Main Gateways:** este grafico se utilizará para ver cuántos devices tienen cada uno de los Gateways seleccionados para el análisis como Main Gateway.
- **Media de RSSI:** en este gráfico, el eje X será los diferentes valores de RSSI (la media de todos los mensajes recibidos en el periodo de tiempo seleccionado) y el eje Y será el número de dispositivos.
- **Media de SNR:** este otro será igual que el anterior, pero utilizando los valores de SNR en vez de los de RSSI.
- **Media de Time on Air:** este otro también será igual que los anteriores dos, pero utilizando los valores de Time on Air en vez de RSSI o SNR.
- **Distancias al Main Gateway**
 - **Media de mensajes perdidos por distancia al Main Gateway:** en este grafico se analizarán las medias de paquetes perdidos por diferentes distancias al Main Gateway.
 - **Media de mensajes perdidos en porcentaje por distancia al Main Gateway:** este otro utilizara los mismos datos que el anterior, pero se mostrara en porcentaje al número máximo de posibles paquetes perdidos.
 - **Media de mensajes recibidos por distancia al Main Gateway:** en este se mostrarán las medias de paquetes recibidos por diferentes distancias al Main Gateway.
 - **Media de mensajes recibidos en porcentaje por distancia al Main Gateway:** este será igual que el anterior, pero en vez de utilizar el número de paquetes recibidos, se mostrará un porcentaje de paquetes recibidos, en comparación al máximo posible.
 - **Media de RSSI por distancia al Main Gateway:** en este se mostrará la media del RSSI a la que llegan los paquetes por la distancia a la que los devices se encuentran del Main Gateway.
 - **Media de SNR por distancia al Main Gateway:** este será igual que el anterior gráfico, pero mostrando los valores de SNR.
 - **Media de Time on Air por distancia al Main Gateway:** este también será igual que los anteriores dos, pero mostrando los valores de ToA.
- **Último SF utilizado**
 - **Media de mensajes perdidos en porcentaje por último SF utilizado:** en este grafico se clasificarán los devices por el ultimo SF que hayan utilizado y se mostrara la media de paquetes perdidos en porcentaje al máximo posible por cada SF.
 - **Media de mensajes recibidos en porcentaje por último SF utilizado:** este otro será igual que el anterior, pero utilizando los paquetes recibidos en vez de los perdidos.
 - **Media de RSSI por último SF utilizado:** en este se mostrará la media de RSSI de los paquetes recibidos dependiendo del último SF del dispositivo.

- **Media de SNR por último SF utilizado:** en este último, se mostrará lo mismo que en anterior, pero utilizando los valores de SNR en vez de RSSI.

Estos gráficos servirán para analizar los datos obtenidos de cada análisis del estado de la red después de aplicar los cambios. Además, estos también servirán para comparar los resultados de los diferentes estudios, visualizando en un único gráfico el antes y el después.

5.2. Creación de interfaz de usuario

En este segundo apartado, el objetivo es crear una interfaz gráfica para poder realizar tantos análisis de la red LoRa de Itelazpi como se quiera, de cualquier zona seleccionada y englobando todo lo posible.

Para ello, como primer paso se han generado diversos scripts utilizando el lenguaje de programación Python. Estos scripts se pueden dividir en dos secciones, los que sirven para la obtención de los parámetros y los que sirven para tratar los datos obtenidos para generar los gráficos.

Después de haber generado los scripts, se creará una interfaz de usuario funcional para unificar todos estos. Y finalmente, se retocará la interfaz de usuario funcional con el fin de hacerla más amigable para el usuario final, además de crear un instalador para poder instalar la aplicación en cualquier máquina.

En resumen, en este bloque se explicarán en profundidad los siguientes subapartados:

- Generación de scripts para la obtención de datos
- Generación de scripts para la visualización de gráficos
- Creación de interfaz de usuario funcional
- Creación de interfaz de usuario final
- Corrección de errores y casos límites

NOTA: a lo largo de este apartado se hará referencia a diferentes carpetas y/o archivos del proyecto completo subido en GitHub, con el fin de poder seguir mejor la explicación. El proyecto completo se encuentra en este enlace: <https://github.com/isedano005/LoRa-Network-Analyzer>.

5.2.1. Generación de scripts para la obtención de datos

Para la obtención de datos de la API de Orbiwise, es necesario distinguir entre los devices y los Gateways, ya que la información de los parámetros de estos no se extrae del mismo sitio. Por ende, este apartado se dividirá en dos, la que concierne a la obtención de los parámetros de los devices y la que concierne a los Gateways.

NOTA: el código descrito en este subapartado (con algunas modificaciones ya que después se ha adaptado a la interfaz de usuario) se puede encontrar en el módulo de la API (**api_module**) del trabajo final.

Parámetros de los devices

Para obtener los parámetros previamente definidos de los devices, es necesario seguir una serie de pasos hasta llegar a los parámetros requeridos. En la Ilustración 21 se puede observar el diagrama de bloques de lo que concierne a la obtención de estos parámetros de una forma sencilla.

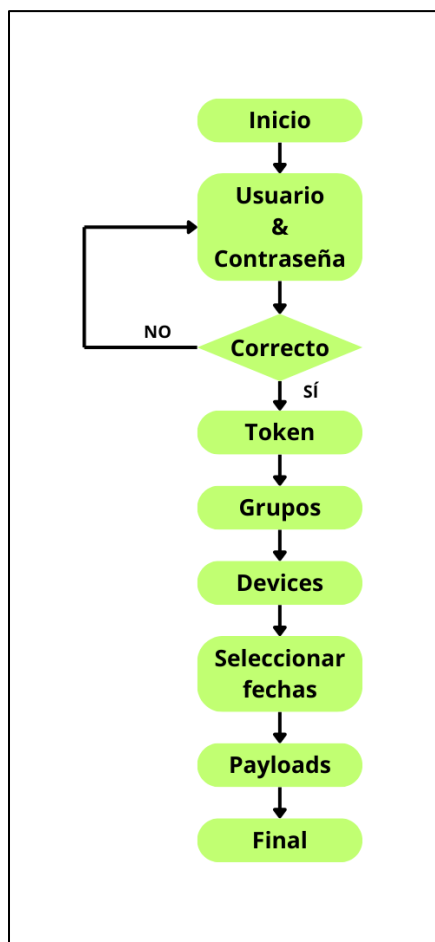


Ilustración 21: Diagrama de bloques de la obtención de parámetros de los dispositivos

El diagrama se podría resumir en los siguientes puntos:

- Pedir las credenciales del usuario con el que se quiere realizar el análisis.
- Obtener un token para la API asociado al usuario para mayor seguridad.
- Obtener los grupos asociados al usuario.
- Seleccionar un grupo y obtener todos los dispositivos asociados a este.
- Seleccionar el rango de fechas en las que se quiere realizar el análisis.
- Obtener todos los payloads de todos los dispositivos en ese rango de tiempo.

Toda esta información obtenida se va almacenando en diferentes archivos JSON para poder tratar estos datos más adelante.

Entrando más en detalle en cada uno de estos puntos, lo primero de todo es autenticarse contra el servidor mediante unas credenciales. Estas credenciales son el usuario y contraseña que se utilizan para la interfaz web de Orbiwise.

Tal y como Itelazpi tiene organizado los diferentes usuarios, las credenciales utilizadas para el caso de estudio de Galdakao solo sirven para un único cliente (CABB) y una única marca de contadores en concreto (iTron).

Como la parte de seguridad es importante ya que se está trabajando sobre una red en producción, se ha decidido utilizar un token para aumentar la seguridad de las llamadas a la API, en vez de estar transmitiendo una y otra vez las credenciales mismas.

Para ello, la API de Orbiwise cuenta con un endpoint que permite obtener un token para un cierto tiempo con el que las llamadas a la API quedarán autenticadas con el usuario y contraseña que previamente se han introducido.

El endpoint en cuestión es el siguiente:

<https://itelorbiwise.itelazpi.eus/rest/oauth2/token>

Y hace falta pasarle los siguientes datos para obtener el token:

- “grant_type”: “password”
- “username”: {usuario}
- “password”: {contraseña}

Una vez obtenido el token, el siguiente paso es obtener todos los grupos dentro del usuario. Como ya se ha mencionado anteriormente, los grupos son una forma de organizar los dispositivos, y entre las formas de organizarlos se encuentra la geográfica, definiendo un grupo por cada municipio y asignado estos grupos a cada device correspondiente.

El endpoint para obtener los grupos es el siguiente:

<https://itelorbiwise.itelazpi.eus/rest/groups>

Y se le pasa el siguiente parámetro:

- “Authorization”: “Bearer {token}”

Con esto se obtiene una lista de todos los grupos existentes en ese usuario. De estos grupos se selecciona uno de ellos, en este caso “galdakao”, y se vuelve a hacer una llamada a la API para obtener todos los devices asociados al grupo. Para ello se llama al siguiente endpoint:

<https://itelorbiwise.itelazpi.eus/rest/groups>

Como el número de devices puede ser elevado (en el caso de Galdakao unos 13.000), la API para evitar sobrecargas devuelve un máximo de 2.000 devices por llamada, pero es posible obtenerlos todos si se le añaden los siguientes parámetros para obtener una lista paginada:

- “Authorization”: “Bearer {token}”
- “group”: “galdakao”
- “limit”: 2000
- “get_pages”: “true”

Con esta llamada, la API devuelve varias referencias a las diferentes páginas para poder realizar varias consultas una detrás de otra para ir obteniendo los devices asociados al grupo en grupos de 2.000 devices por llamada.

Estas referencias vienen en el parámetro “page_state”, y añadiendo este nuevo parámetro a la llamada al endpoint en vez de “get_pages”, se van obteniendo todos los devices asociados al grupo seleccionado.

Finalmente, una vez habiendo obtenido la lista de todos los devices de los que se va a realizar el análisis, se obtendrán todos los payloads de estos, esto es, todos los datos que los devices han enviado.

Para ello, primero será necesario seleccionar un rango de tiempo en el que se va a realizar el análisis. En el caso de Galdakao, los análisis recogen datos de 7 días, tiempo suficiente para tener una cantidad de datos considerable. Se podría escoger un rango más amplio, pero esto supone más cantidad de datos, o lo que es lo mismo, más carga de trabajo para la API y para la aplicación a desarrollar en sí.

Para obtener los payloads de cada device, es necesaria una llamada a la API por cada device.

<https://itelorbiwise.itelazpi.eus/rest/nodes/{deveui}/payloads/ul>

Además, a cada llamada es necesario añadirle los siguientes parámetros:

- "Authorization": "Bearer {token}"
- "data_format": "hex"
- "from_date": {fecha de inicio del análisis}
- "to_date": {fecha de fin del análisis}

Una vez realizadas todas las llamadas a la API necesarias y habiendo obtenido todas las respuestas, todos estos datos se almacenan en archivos JSON para trabajar con ellos más adelante.

Con esto habría concluido la extracción de datos de los devices de la API, por lo que ahora es el turno de los Gateways.

Parámetros de los Gateways

Para la obtención de los parámetros de los Gateways, se seguirá la lógica explicada en el diagrama de bloques de la Ilustración 22.

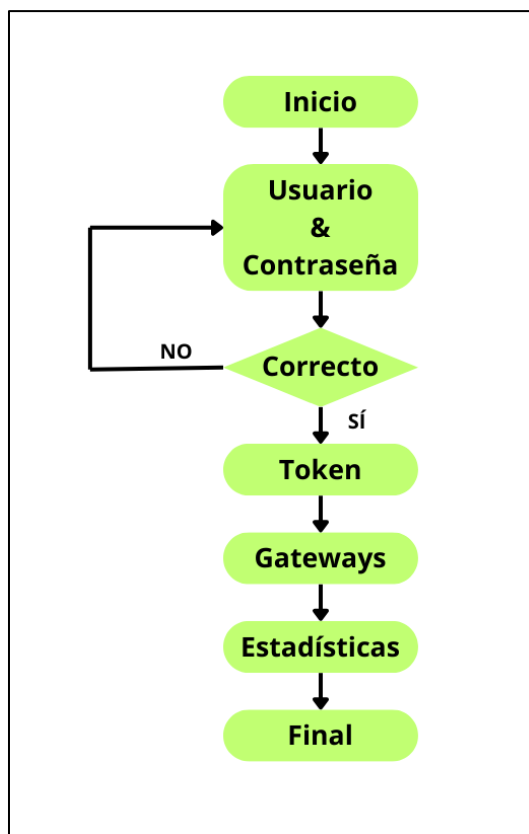


Ilustración 22: Diagrama de bloques de la obtención de parámetros de los Gateways

El diagrama de la imagen es parecido al de los devices, teniendo un inicio prácticamente idéntico. Entrando en más en detalle en el diagrama, los pasos a seguir para obtener los parámetros predefinidos de los Gateways serían los siguientes:

- Pedir las credenciales del usuario administrador de Gateways.
- Obtener un token para la API asociado al usuario para mayor seguridad.
- Obtener la lista de todos los Gateways de la red.
- Seleccionar los Gateways a analizar y obtener los parámetros de cada uno de ellos.

Al igual que se ha hecho con los devices, estos datos se almacenarán en archivos JSON para poder trabajar con ellos más adelante.

Entrando en detalle en cada bloque, se puede observar que hasta la obtención del token el proceso es idéntico al conseguido con los devices. La única diferencia serían las credenciales usadas para obtener el token, ya que, para el caso de los Gateways, existe un usuario que se encarga de almacenar la información de todos los Gateways.

Por eso mismo, el endpoint para obtener el token es el mismo, pero los parámetros utilizados son diferentes:

<https://itelorbiwise.itelazpi.eus/rest/oauth2/token>

Los parámetros en cuestión serían los siguientes:

- “grant_type”: “password”
- “username”: {usuario_gateways}
- “password”: {contraseña_gateways}

Una vez obtenido el token, el siguiente paso es obtener la lista de todos los Gateways que forman la red LoRa. Para ello, se llama al siguiente endpoint:

<https://itelorbiwise.itelazpi.eus/rest/gateways>

Y se le pasa el siguiente parámetro:

- “Authorization”: “Bearer {token}”

Una vez habiendo obtenido el listado de todos los Gateways, será necesario seleccionar cuales se van a analizar. Esto se puede hacer de una forma relativamente sencilla, ya que los Gateways se nombran teniendo en cuenta la ubicación en donde se han instalado, por lo que, para este caso de estudio de Galdakao, todos los Gateways que se encuentran instalados en el municipio, tienen en el nombre “Galdakao”.

Teniendo ya los Gateways que se quieren analizar definidos, se pasa a utilizar la API del NST para obtener los parámetros de los Gateways. Como ya se ha mencionado anteriormente, esta API no funciona tan bien como la de Orbiwise, y para este caso de la obtención de los parámetros de los Gateways, el endpoint utilizados no devuelve un objeto al hacer la llamada, si no que, la única forma de obtener datos del endpoint es haciendo la llamada a través de un navegador, como si fuese una URL.

Esto implica que la automatización del proceso se complica un poco, ya que es necesario el uso de un navegador en vez de llamadas a la API a través de código Python. Para solventar este problema, se utilizará web scraping [5], una técnica para obtener datos de páginas web de una forma automatizada.

La utilización del web scraping conlleva el uso de un navegador en segundo plano (totalmente transparente para el usuario), con lo que, ralentiza el proceso de la obtención de datos comparando a la utilización de únicamente llamadas simples a la API.

Dicho esto, las URLs que se van a utilizar tendrán este formato:

```
https://itelorbiwise.itelazpi.eus/tfw1/rest?getGWStatsAggregated=1&from={from_date}  
&to={to_date}&gw={gw_id}&tz=-60
```

Donde {from_date} y {to_date} serían las mismas fechas seleccionadas en la parte de obtención de los devices y {gw_id} sería el identificador único de cada Gateway, por lo que habría que utilizar una URL por cada Gateway que se quiera analizar.

Todos los datos se irán almacenando en archivos de una forma ordenada para poder utilizarlos posteriormente en el tratamiento de estos datos y la creación de gráficos utilizando estos mismos.

5.2.2. Generación de scripts para la visualización de gráficos

Después de haber generado los scripts para la obtención de los parámetros tanto de los devices como de los Gateways, el siguiente paso consiste en crear otros nuevos scripts para la creación de los gráficos que previamente se habían definido. Para ello, se seguirá utilizando el lenguaje de programación Python.

Antes de empezar a crear los gráficos, es necesario adaptar los datos previamente obtenidos que se han ido guardando en diferentes archivos JSON. Con esto se pretende homogeneizar los datos y obtener un único archivo con los parámetros que se van a graficar para los devices y otro para los Gateways.

NOTA: el código descrito en este subapartado (con algunas modificaciones ya que después se ha adaptado a la interfaz de usuario) se puede encontrar en el módulo de los datos (**data_module**) del trabajo final.

Tratamiento de datos

Para empezar, se han cogido todos los payloads, previamente obtenidos, de todos los devices y se han calculado los valores medios de diferentes parámetros, con los datos obtenidos de la franja de tiempo elegida. Esto se ha hecho ya que la API no devuelve los valores medios, sino que devuelve un valor asociado a cada uno de los payloads.

Los parámetros de los cuales se ha calculado el valor medio son los siguientes:

- RSSI
- SNR
- ToA
- Número de Gateways alcanzados

Por otro lado, se ha calculado también cual ha sido el Main Gateway más común para cada device de todos los payloads. Además, se han contado el número de mensajes (payloads) recibidos (received frames) y se ha calculado, en base al número máximo de mensajes recibidos, el número de mensajes perdidos (lost frames). Finalmente, del último payload recibido se extrae el SF en el cual está emitiendo el device.

Una vez hecho eso, se crea un nuevo archivo JSON con estos datos y también con varios parámetros que se han obtenido directamente de la API al listar los devices. El archivo en cuestión contara con los siguientes parámetros para cada device:

- Deveui
- Latitud
- Longitud
- Media de RSSI
- Media de SNR
- Media de ToA
- Media de número de Gateways alcanzados
- Main Gateway
- Mensajes perdidos
- Mensajes recibidos
- Ultimo SF utilizado

Teniendo todos estos datos, los únicos parámetros restantes para cada device son las distancias al Main Gateway y al Gateway más cercano. Para poder calcular estos parámetros, se utilizarán las coordenadas (latitud y longitud) de los devices y de los Gateways. Entonces, para cada device se calculará la distancia a su Main Gateway utilizando la fórmula de Haversine (Ilustración 23). Además, con el mismo método, se calculará la distancia al resto de Gateways del análisis y se obtendrá el más cercano al device y la distancia a la que se encuentra.

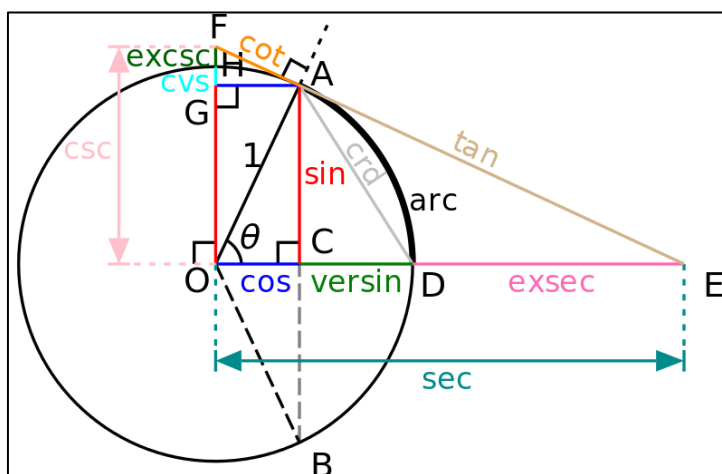


Ilustración 23: Explicación gráfica de la fórmula de Haversine [6]

Una vez habiendo realizado estos cálculos, se añaden los siguientes tres campos a los ya existentes para cada device, obteniendo así toda la información necesaria para cada uno de los dispositivos.

- Distancia al Main Gateway (m)
- Gateway más cercano
- Distancia al Gateway más cercano (m)

Teniendo ya un único archivo JSON con todos los parámetros de los devices, el siguiente paso es realizar algo similar con todos los Gateways.

Para ello, lo primero de todo sería unir los parámetros obtenidos individualmente de cada Gateway en un único archivo. En el caso de los Gateways, la API sí que facilita los valores medios de los parámetros, por lo que no sería necesario realizar cálculos como con los devices.

De todos los parámetros que se obtienen de cada Gateway haciendo web scraping, se guardarán los siguientes:

- Identificador de Gateway
- Numero de mensajes de uplink recibidos
- Numero de paquetes perdidos
- Media de RSSI (de los paquetes recibidos)
- Media de SNR (de los paquetes recibidos)

Y una vez unificados todos esos datos, se combinarán con los obtenidos al haber listado los Gateways, los cuales son los siguientes:

- Nombre del Gateway
- Latitud
- Longitud

Con todo esto, el tratamiento de datos se puede dar por finalizado ya que se ha conseguido unificar los parámetros de los devices en un único archivo JSON, al igual que los parámetros de los Gateways, que se guardan en otro archivo JSON diferente.

Creación de gráficos

Teniendo ya todos los parámetros tanto de los devices como de los Gateways en archivos JSON de un formato conocido, el siguiente paso es generar los scripts para obtener los gráficos previamente definidos a partir de estos datos.

Los gráficos que se van a crear van a ser interactivos, esto es, no una simple imagen sino unos gráficos en los que se van a poder añadir y quitar ciertos parámetros, hacer clic en las barras de los datos y obtener información adicional, etc. Para ello, los gráficos generados tendrán la extensión HTML, en vez de las típicas PNG o JPEG de las imágenes.

El primer gráfico que se va a generar será el mapa que contendrá la localización de los Gateways, con información adicional de cada uno de estos, y la ubicación de cada device. Estos últimos serán representados como puntos, ya que al poder ser tantos (en este caso unos 13.000), el mapa tarda demasiado tiempo en cargar y no es cómodo visualizarlo.

Para crear el mapa se han utilizado 2 librerías de Python:

- **Geopy [7]:** Para obtener las coordenadas centrales del municipio del que se realiza el análisis y así centrar el mapa.
- **Folium [8]:** Para dibujar un mapa y añadir marcadores de los Gateways y los devices de una forma sencilla.

Con la ayuda de estas librerías, se ha generado un script capaz de crear un mapa interactivo que muestre todos los Gateways y devices seleccionados para el análisis y guardar este mapa en un archivo HTML (mapGWsAndDevices.html).

Los gráficos restantes son gráficos de barras, por lo que no necesitaran las librerías geopy y folium, pero si esta otra:

- **Plotly [9]**

Esta librería facilita la creación de gráficos interactivos, entre ellos los de barras, que ha sido el método elegido para visualizar los datos.

Los primeros gráficos de barras que se van a crear son los relacionados con los Gateways, los cuales tienen en el eje X cada uno de los Gateways que se ha seleccionado para el análisis. Todos los datos necesarios para crear estos gráficos se encuentran en el JSON de información de los Gateways.

Una vez teniendo esos datos, para la creación de los gráficos se utilizará como eje X el identificador del Gateway, y para el eje Y, los siguientes parámetros, uno para cada gráfico.

- Numero de mensajes de uplink recibidos
- Numero de paquetes perdidos
- Media de RSSI (de los paquetes recibidos)
- Media de SNR (de los paquetes recibidos)

Y con estos datos se generan los siguientes gráficos interactivos.

- uplinkFrameCounter_x_gateways.html
- lostFrameCounter_x_gateways.html
- averageRSSI_x_gateways.html
- averageSNR_x_gateways.html

Con estos gráficos generados, el siguiente paso sería el crear otros scripts para los gráficos de los devices. Primero, se crearán los gráficos de distribuciones, después los que ordenan los devices por distancias al Main Gateway, y finalmente, los que los ordenan por el ultimo SF utilizado.

Empezando con las distribuciones, lo primero que se ha hecho es organizar y agrupar los datos en bloques para poder mostrar de una forma sencilla, pero a la vez significativa, los datos obtenidos. Los valores se han agrupado de la siguiente manera:

- **Media de RSSI:** se ha redondeado el valor de cada device al entero más cercano, haciendo que en el eje X se muestren solo los valores enteros en saltos de 1 dBm. El eje Y muestra el número de devices que, después de redondear, tienen esa media de RSSI.
- **Media de SNR:** exactamente igual que la media de RSSI, pero con los valores de SNR. El eje X muestra valores enteros de SNR en saltos de 1 dB y el eje Y el número de devices que tienen esa media de SNR, después de haber redondeado el valor.
- **Media de ToA:** el valor de ToA se redondea al entero múltiplo de diez más cercano, dejando así en el eje X saltos de 10 ms.
- **Mensajes perdidos:** el número de mensajes perdidos siempre va a ser un número entero, por lo que se agrupan dependiendo ese número.
- **Mensajes recibidos:** al igual que el número de mensajes perdidos, estos también serán un número entero. Además, este gráfico y el de mensaje perdidos será inversamente proporcional.
- **Media de número de Gateways alcanzados:** el número de media de Gateways alcanzados se redondea al décimo más cercano, dando más precisión al gráfico que si

se redondease al entero más cercano. Por lo que, el eje X muestra saltos de 0.1 Gateways alcanzados.

- **Main Gateways:** en este caso, el eje X serán los identificadores de los Gateways (como con los gráficos de los Gateways) y como cada device cuenta con un Main Gateway, se cuentan el número de ocurrencias de cada uno de estos y se muestran en el gráfico, siendo el eje Y el número de devices que tienen a dicho Gateway asociado como el Main Gateway.
- **Último SF utilizado:** parecido al gráfico de Main Gateways, pero en este caso se utilizado el parámetro del último SF utilizado por cada device como eje X. Como eje Y, el número de devices que han utilizado ese SF.

Siguiendo el mismo orden que la lista anterior, los gráficos que se obtienen se guardan con los siguientes nombres:

- RSSI_distribution.html
- SNR_distribution.html
- timeOnAir_distribution.html
- lostFrames_distribution.html
- receivedFrames_distribution.html
- averageGatewaysReached_distribution.html
- mainGateways_distribution.html
- lastSFUsed_distribution.html

Una vez generado los gráficos de distribuciones, los siguientes gráficos son los relacionados con las distancias de los devices al Main Gateway.

Para ello, se agrupan los devices a partir de la distancia al Main Gateway y redondeando al centenar más cercano, dejando así en el eje X la distancia al Main Gateway en saltos de 100 metros.

Una vez teniendo los devices agrupados de esta manera, se calcularán las medias de diferentes parámetros dentro de cada rango de distancia, para así poder graficar los valores de los devices de una forma relativamente precisa sin sobrecargar mucho el grafico de datos.

Las medias calculadas por cada agrupación de distancia han sido de los siguientes parámetros:

- **Media de RSSI:** se calcula la media de todas las medias de RSSI de los devices que estén dentro del rango de distancia. Además de graficar las barras, una para distancia, también se grafica una línea continua para poder observar la media acumulada de todos los valores de RSSI de los devices que se encuentran a menos de X distancia del Main Gateway, donde X es 100, 200, 300, etc.
- **Media de SNR:** exactamente igual que con los valores de RSSI, pero con los valores de SNR para este gráfico. Además de la media acumulada.
- **Media de ToA:** igual que con RSSI y SNR, pero esta vez con los valores de Time on Air en milisegundos. También se grafica la media acumulada para este caso.
- **Media de mensajes perdidos:** para este grafico se calcula la media de mensajes perdidos por device en el rango de distancia al Main Gateway. Además, se calcula también la media acumulada de mensajes perdidos a lo largo de las distancias.
- **Media de mensajes perdidos en porcentaje:** se utilizan los mismos datos que para el anterior gráfico, pero en vez de utilizar el valor absoluto de mensajes perdidos, se

calcula el porcentaje de ese valor en comparación al número máximo de mensajes que podría haber recibido el device en ese rango de tiempo.

- **Media de mensajes recibidos:** al igual que con los mensajes perdidos, se calcula la media de mensajes recibidos en los diferentes rangos de distancia al Main Gateway, además calcularse la media acumulada.
- **Media de mensajes recibidos en porcentaje:** para este gráfico se utilizan los mismos datos que para el anterior gráfico, y al igual que con los mensajes perdidos, se calcula la media de mensajes recibidos en porcentaje al número máximo de mensajes recibidos posibles.

Los gráficos obtenidos con las agrupaciones que se acaban de describir se guardan en archivos HTML. En el mismo orden que la anterior lista, estos son los nombres utilizados para almacenar los datos:

- averageRSSI_x_distanceToMainGateway.html
- averageSNR_x_distanceToMainGateway.html
- averageTimeOnAir_x_distanceToMainGateway.html
- averageLostFrames_x_distanceToMainGateway.html
- averageLostFramesPercentage_x_distanceToMainGateway.html
- averageReceivedFrames_x_distanceToMainGateway.html
- averageReceivedFramesPercentage_x_distanceToMainGateway.html

Después de haber creado el script para generar los gráficos de los devices dependiendo la distancia de estos al Main Gateway, el ultimo script a crear sería el de los gráficos de los devices dependiendo el ultimo SF que estos han utilizado.

Para ello, se agrupan los devices dependiendo el ultimo SF que han utilizado, y como se sabe que solo puede haber 5 posibles SF (SF7 – SF12), no será necesario redondear estos valores. Por lo que, en el eje X de estos gráficos aparecerán los 5 SFs posibles, y en el eje Y el parámetro que se pretenda analizar. Estos parámetros serán los siguientes:

- **Media de mensajes recibidos en porcentaje:** se hace la media de todos los mensajes recibidos de los devices que han transmitido en cada SF, y después se calcula el porcentaje de ese valor en comparación al máximo de mensajes recibidos posibles.
- **Media de mensajes perdidos en porcentaje:** exactamente igual que con los mensajes recibidos, pero con los mensajes perdidos.
- **Media de RSSI:** se calcula la media de los valores de RSSI medios de todos los devices que haya utilizado cada SF.
- **Media de SNR:** igual que el anterior gráfico, pero utilizando los valores de SNR en vez de RSSI.

A todos estos scripts de gráficos también hay que añadirles la opción de poder generar los gráficos con hasta 3 fuentes de datos diferentes y poder visualizar en el mismo gráfico los diferentes resultados. Esto es, utilizando los datos de los diferentes análisis realizados, generar estos gráficos combinados.

Con todo esto, todos los scripts necesarios para realizar un análisis de la red y visualizar gráficos con los datos obtenidos han sido creados.

5.2.3. Creación de interfaz de usuario funcional

Con el fin de facilitar el uso de los scripts para cualquier usuario, se creará una interfaz gráfica para así poder realizar un análisis sin tener que tener conocimientos de como ejecutar scripts de Python o en qué orden es necesario ejecutarlos.

Con esta interfaz se pretende simplificar el proceso del análisis y que el usuario tenga que interactuar lo mínimo posible, solo teniendo que facilitar datos clave en cada análisis, como las credenciales de los usuarios, el municipio elegido, los Gateways a analizar, etc.

Para la programación de esta interfaz gráfica, y siguiendo con el lenguaje de programación Python, se ha utilizado la siguiente librería:

- Tkinter [10]

Esta librería permite crear ventanas de una manera sencilla y la posibilidad de añadir diferentes componentes (texto, imágenes, botones, etc.) para completar el diseño.

La funcionalidad de la interfaz será bastante sencilla y lineal, pero se puede dividir en dos aspectos principales, la ejecución de un análisis completo (desde la obtención de los datos hasta la creación de los gráficos) y la comparación entre diferentes análisis, para lo cual los datos ya han sido obtenidos y únicamente se trabajará con ellos para generar los gráficos combinados.

En la Ilustración 24 se puede observar con más detalle el diagrama de bloques del funcionamiento de la aplicación.

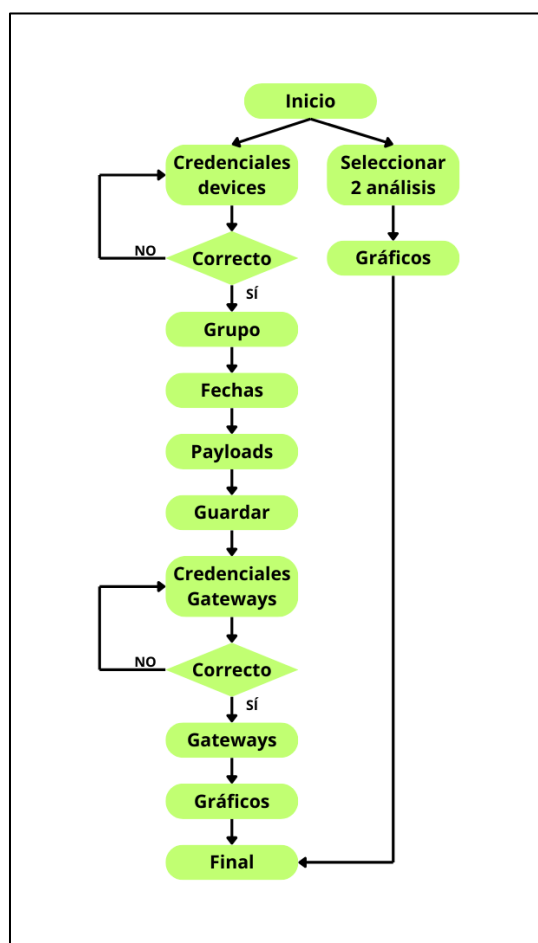


Ilustración 24: Diagrama de bloques de la GUI funcional

Nada más ejecutar la aplicación, la pantalla principal (Ilustración 25) dará la opción de ingresar las credenciales del usuario de los devices para iniciar el análisis o de empezar una comparación entre dos análisis previamente realizados.

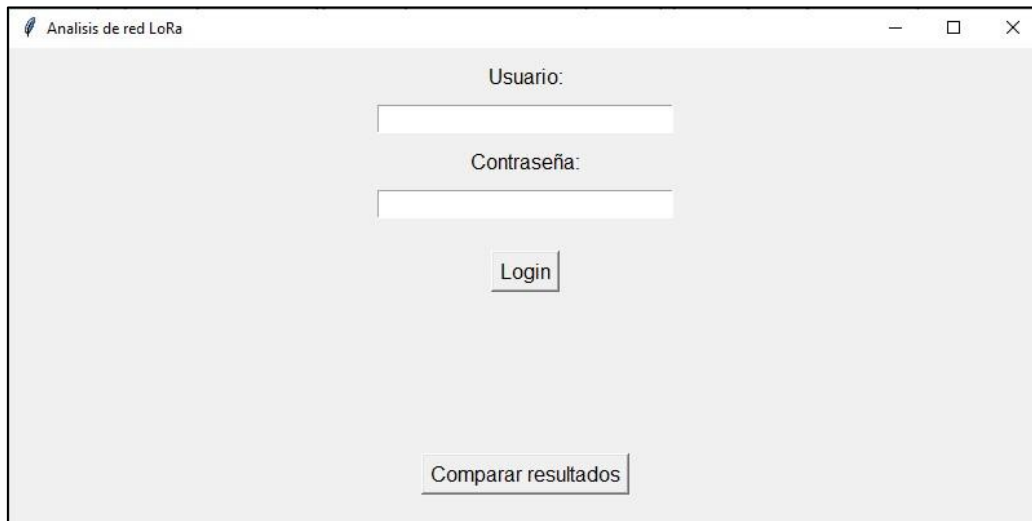


Ilustración 25: Pantalla principal (GUI funcional)

Si se introducen las credenciales del usuario donde están dados de alta los devices de los que se quiere hacer un análisis, se mostrará la siguiente pantalla (Ilustración 26). Para ello, en segundo plano se ejecutarán los scripts de obtención de token y grupos previamente creado y modificados para integrarlos con la interfaz. En caso de haber introducido incorrectamente las credenciales, se volverán a pedir hasta que sean correctas.

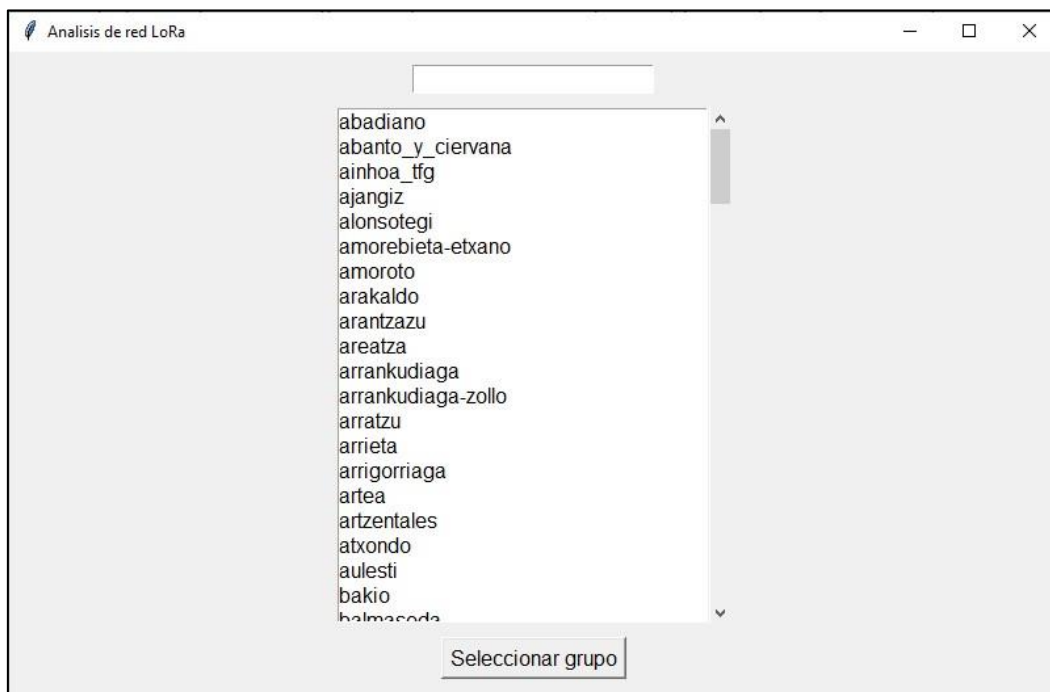


Ilustración 26: Pantalla de selección de grupo (GUI funcional)

En esta pantalla se podrá filtrar por nombre o buscar directamente entre todos los grupos que se encuentran en ese usuario.

Una vez encontrado el grupo (municipio) en donde se quiere hacer el análisis, se obtendrán todos los devices dentro de ese grupo (dependiendo de la cantidad el programa tardará más o menos) y una vez obtenidos todos los devices, se pasará a la siguiente pantalla (Ilustración 27).

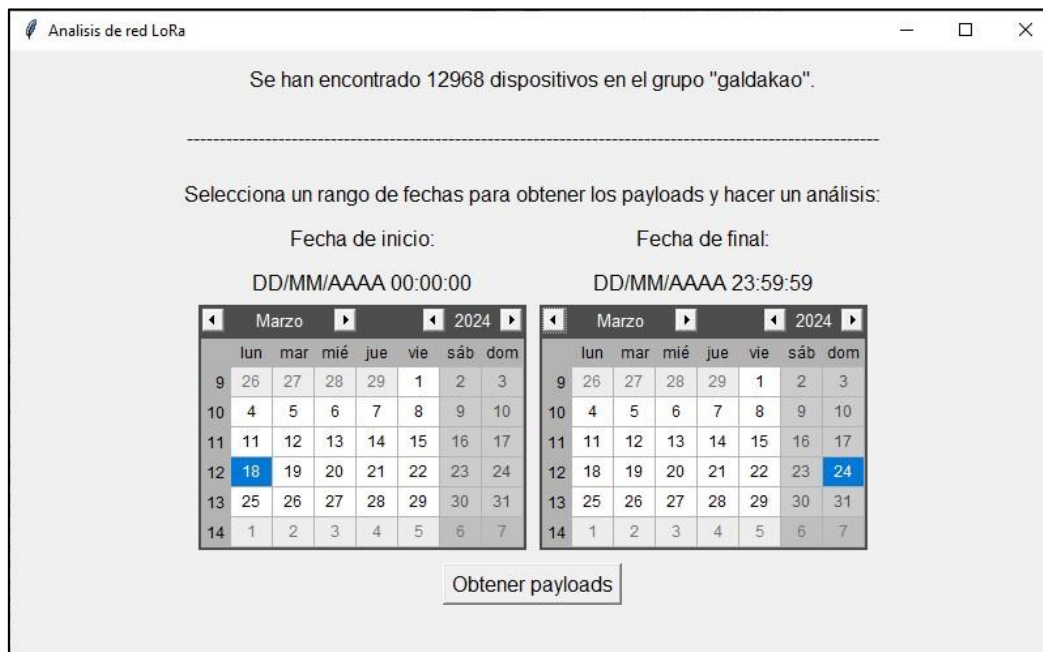


Ilustración 27: Pantalla de selección de tiempo (GUI funcional)

En esta nueva pantalla se podrá observar el número de devices asociados al grupo seleccionado. Además, se mostrarán dos calendarios para poder seleccionar con precisión la fecha de inicio y final del análisis.

Para crear los calendarios y poder trabajar con ellos, se ha utilizado el siguiente modulo:

- **Tkcalendar [11]**

Cabe mencionar que, para las fechas del análisis, se han establecido por defecto que la hora de inicio serán las 00:00:00 del día de inicio seleccionado, y que la hora de finalización serán las 23:59:59 del día de finalización del análisis seleccionado. Estos valores no se podrán cambiar a través de la interfaz gráfica por simplicidad.

Una vez seleccionadas las fechas de inicio y fin del análisis, se pasará a la siguiente pantalla (Ilustración 28), donde se podrá ver el progreso de la obtención de datos en forma de barra de carga y un porcentaje.

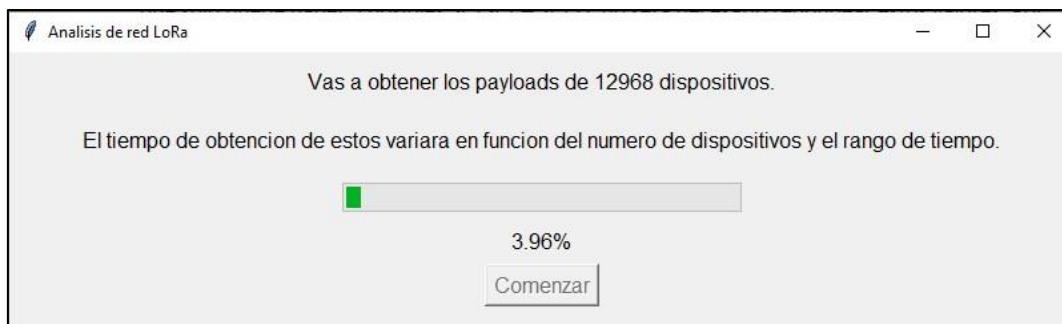


Ilustración 28: Pantalla de obtención de payloads (GUI funcional)

Una vez finalizada la obtención de los datos, el programa pasará a la siguiente pantalla (Ilustración 29) donde se pedirá al usuario elegir la ruta de la carpeta donde se guardarán los datos.

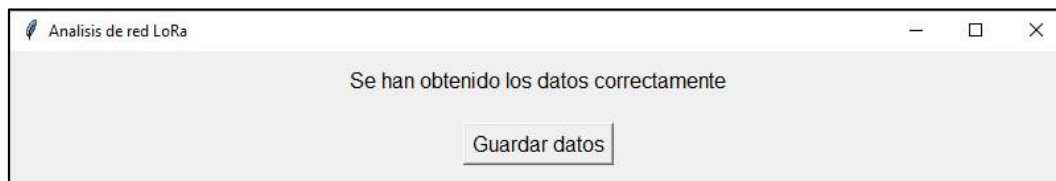


Ilustración 29: Pantalla de guardado (GUI funcional)

El usuario tendrá que seleccionar el directorio raíz de donde se quiere que se guarden los datos, y la aplicación se encargará de crear los subdirectorios correspondientes.

La jerarquía de directorios creados es la siguiente:

- **Directorio raíz:** El elegido por el usuario
 - **Grupo_1:** Nombre del grupo seleccionado para el análisis
 - **FromXXXXXX_ToYYYYYY:** Carpeta para saber en qué fechas ha sido realizado el análisis donde XXXXXX es la fecha de inicio e YYYYYY la fecha de finalización.
 - **Data:** Aquí se almacenarán los archivos JSON de datos en crudo.
 - **NST:** Carpeta para almacenar los datos obtenidos del web scraping.
 - File1.json
 - ...
 - File1.json
 - ...
 - **Graphics:** Carpeta para almacenar los gráficos obtenidos.
 - **DistancesToMainGateways:** Carpeta para guardar los gráficos creados en función de la distancia al Main Gateway.
 - Graphic1.html
 - ...
 - **Distributions:** Carpeta para guardar los gráficos de distribuciones.
 - Graphic1.html
 - ...
 - **Gateways:** Carpeta para guardar los gráficos creados de los Gateways.
 - Graphic1.html
 - ...
 - **LastSFused:** Carpeta para guardar los gráficos creados en función del último SF utilizado.
 - Graphic1.html
 - ...
 - **Maps:** Carpeta para guardar los mapas.
 - Map1.html
 - ...

- **FromAAAAAA_ToBBBBBB**
- ...
- **Grupo_2**
- ...

Por lo que, una vez elegido el directorio raíz, en la pantalla (Ilustración 30) se va a poder observar donde se han almacenado los datos.

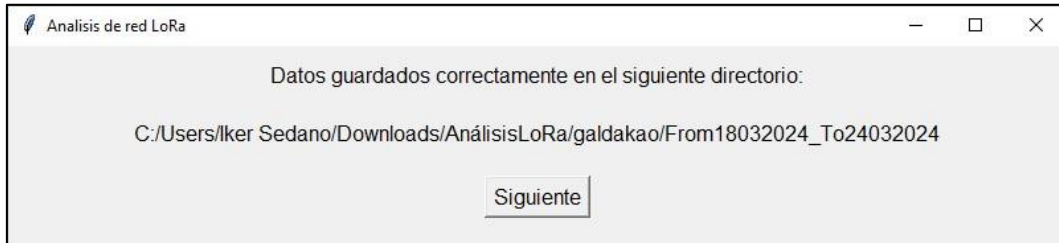


Ilustración 30: Pantalla de datos guardados (GUI funcional)

Con esto, la obtención de datos de los devices ya habría concluido, por lo que el siguiente paso sería la obtención de los datos de los Gateways. Para ello, en la siguiente pantalla (Ilustración 31), se le pedirán al usuario las credenciales del usuario administrador de los Gateways.

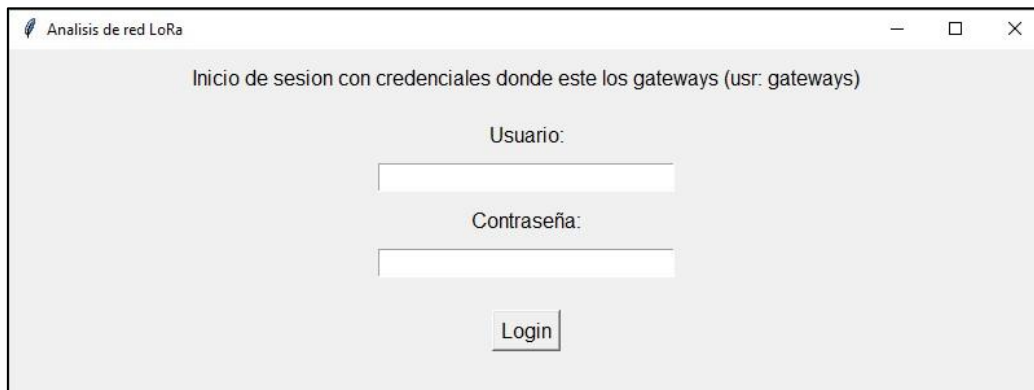


Ilustración 31: Pantalla de ingresar credenciales para los Gateways (GUI funcional)

Si las credenciales son correctas, en la siguiente pantalla (Ilustración 32) se podrá observar una lista de todos los Gateways administrados por Itelazpi.

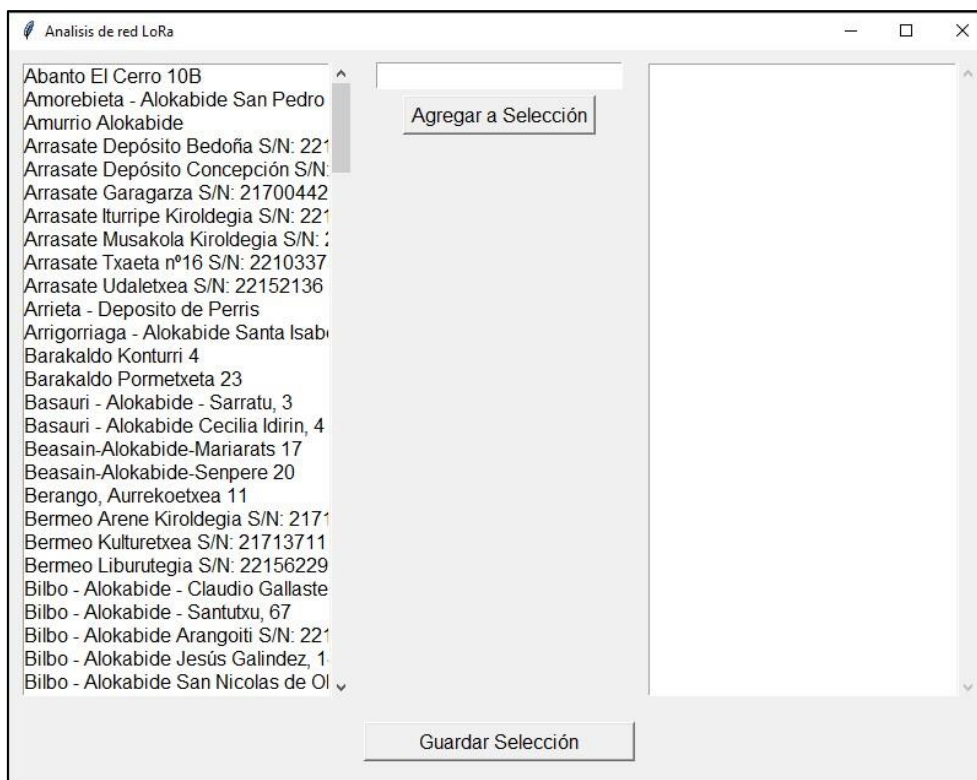


Ilustración 32: Pantalla con la lista de todos los Gateways (GUI funcional)

En esta pantalla, se podrán filtrar por nombre y seleccionar los Gateways que se deseen incluir en el análisis. En la Ilustración 33 se puede observar que para este caso práctico se han seleccionado los Gateways de la zona de Galdakao, los cuales todos ellos tienen en el nombre la palabra “Galdakao”.

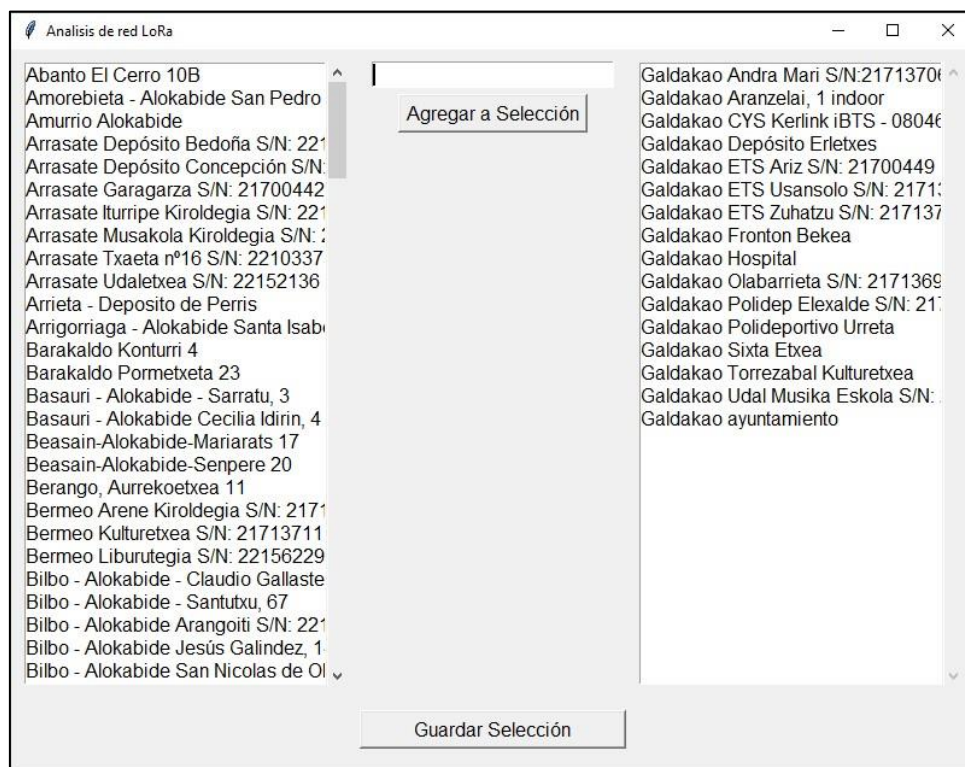


Ilustración 33: Pantalla con Gateways seleccionados (GUI funcional)

Una vez seleccionados los Gateways deseados, se pasará a la siguiente pantalla (Ilustración 34), donde de nuevo se podrá ver una barra de carga mientras se van obteniendo los parámetros de los Gateways seleccionados.

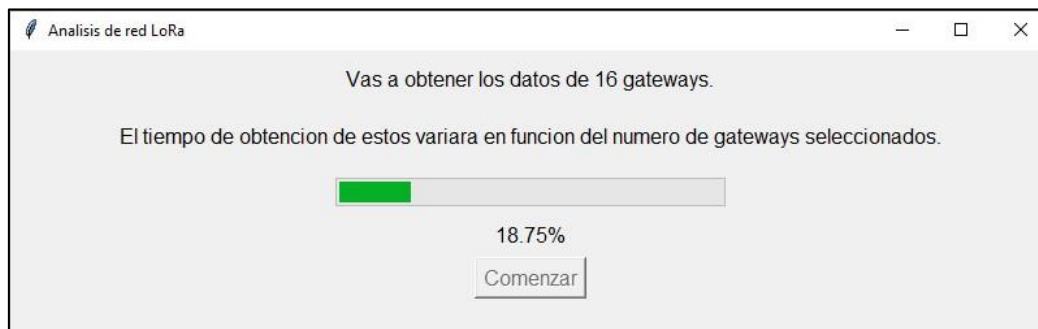


Ilustración 34: Pantalla de obtención de parámetros de Gateways (GUI funcional)

Al finalizar esta obtención de datos, se pasará a la siguiente pantalla (Ilustración 35) donde se iniciará la creación de los gráficos a partir de los parámetros que se han ido obteniendo hasta el momento.

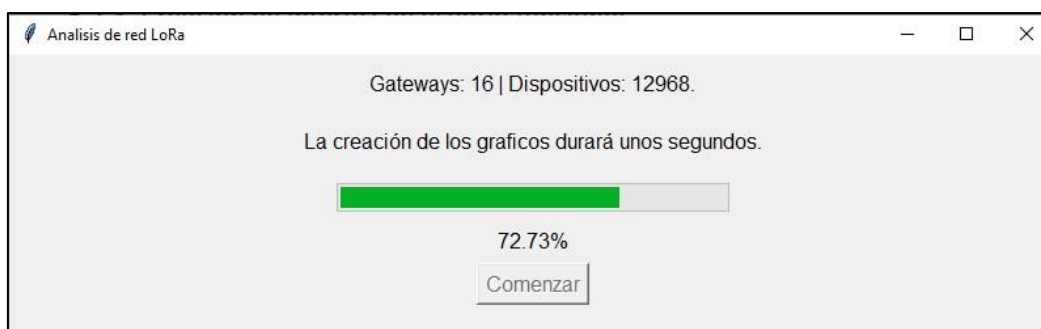


Ilustración 35: Pantalla de creación de gráficos (GUI funcional)

Al finalizar, se mostrará una nueva pantalla simple (Ilustración 36) con la cual se informa al usuario de que el proceso ha finalizado.

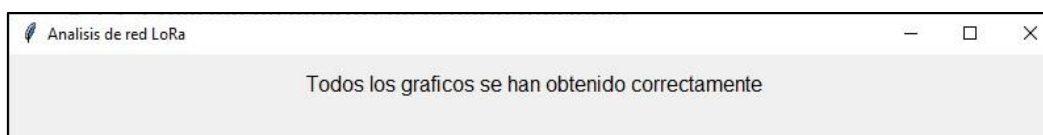


Ilustración 36: Pantalla de finalización (GUI funcional)

Con esto, la parte de realizar un análisis de la red estaría cubierta. Pero, en la pantalla de inicio (Ilustración 25) existe la posibilidad de realizar una comparativa entre dos análisis previamente realizados, para poder obtener los gráficos de ambos análisis unificados y poder analizar mejor la información obtenida.

Una vez en esta pantalla (Ilustración 37), se le dará al usuario la opción de elegir dos carpetas (FromXXXXXX_ToYYYYYY) que contienen los análisis previos.

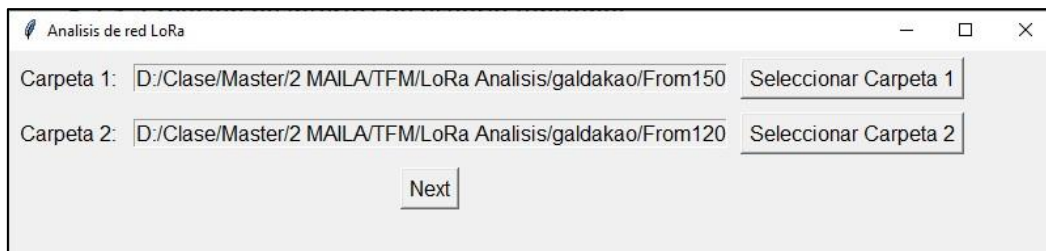


Ilustración 37: Pantalla de selección de carpetas (GUI funcional)

Habiendo seleccionado las carpetas, se generarán los gráficos correspondientes en un nuevo directorio que sigue la siguiente jerarquía:

- Grupo
 - FromXXXXXX_ToYYYYYY
 - FromAAAAAA_ToBBBBBB
 - **Compare_XXXXYYYY_to_AAAABBBB**
 - distancesToMainGateway
 - distributions
 - gateways
 - lastSFused

Donde se creará la carpeta “Compare_XXXXYYYY_to_AAAABBBB” y ahí se guardarán los gráficos creados a partir de los análisis que se habían realizado en las carpetas “FromXXXXXX_ToYYYYYY” y “FromAAAAAA_ToBBBBBB”.

Mientras se generan los gráficos, se volverá a visualizar una barra de progreso (Ilustración 38) para indicar al usuario el porcentaje de completado del proceso.

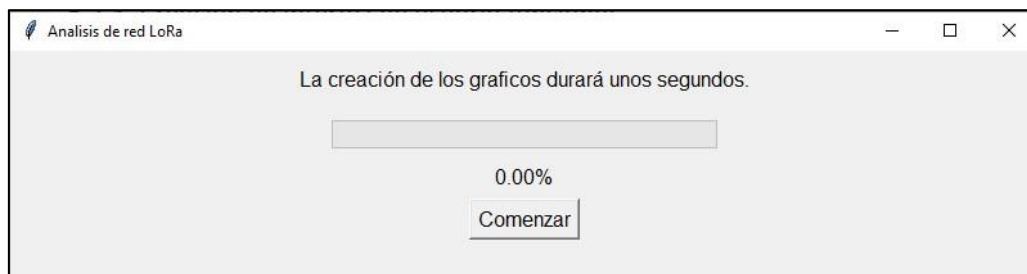


Ilustración 38: Pantalla de creación de gráficos combinados (GUI funcional)

Finalmente, cuando termina el proceso se puede ver la pantalla de finalización (Ilustración 36) al igual que se podía ver cuando se terminaba un análisis.

Con esto, se puede dar por finalizada la interfaz gráfica funcional, ya que cubre la mayoría de aspectos necesarios para realizar un análisis y poder realizar comparativas entre los diferentes análisis realizados.

5.2.4. Creación de interfaz de usuario final

Con el fin de crear una aplicación más profesional y amigable para el usuario final, se decide mejorar la interfaz gráfica añadiéndole algo de diseño y alguna nueva funcionalidad entre otras cosas. Además, la interfaz de usuario funcional es únicamente código Python, ejecutado desde la línea de comandos, pero para crear una aplicación funcional en cualquier ordenador de

Itelazpi, será necesario crear un instalador, ya que no todos los ordenadores, ni usuarios, tienen por qué poder o saber ejecutar Python.

Por ello, en este apartado primero se describirán los cambios de diseño realizados a la aplicación funcional y finalmente se explicará cómo se ha pasado de tener código Python únicamente ejecutable desde la línea de comandos a un instalador capaz de instalar el programa, y todas las dependencias, en cualquier ordenador.

Nuevo diseño

En el nuevo diseño, la funcionalidad de la aplicación es prácticamente idéntica, pero con alguna diferencia en la parte de comparación de análisis y en la finalización de obtención de datos. En la Ilustración 39 se puede observar el diagrama de bloques de esta nueva interfaz, la cual varía ligeramente con el mostrado en la interfaz funcional.

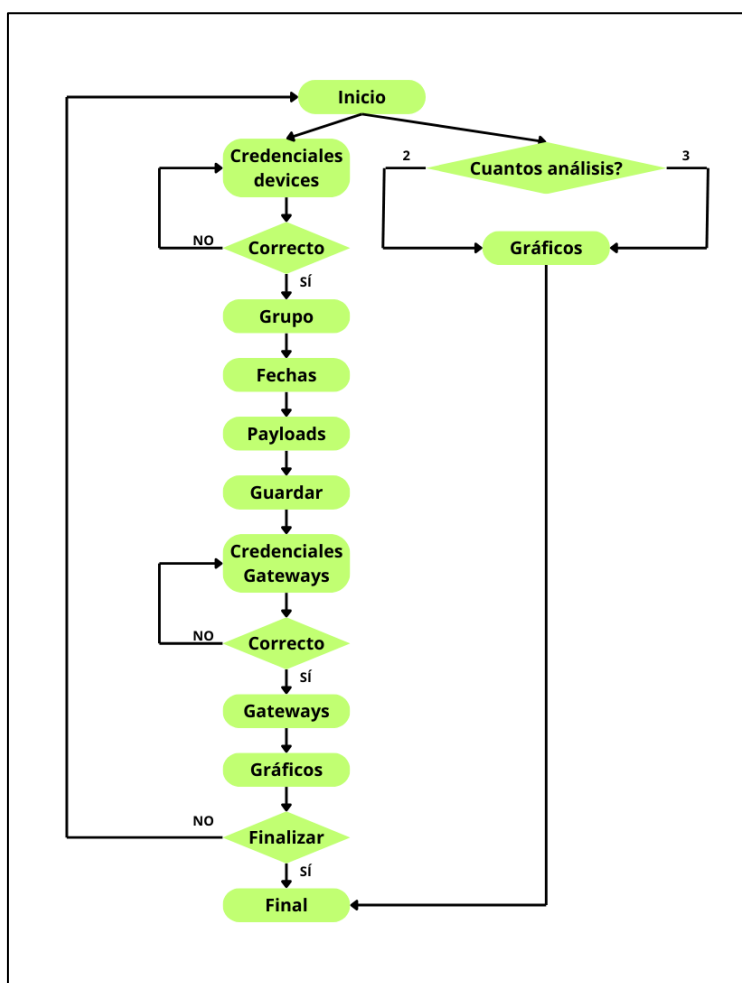


Ilustración 39: Diagrama de bloques de la GUI final

Además de estos cambios, esta nueva interfaz se ha diseñado con la intención de ser más amigable para el usuario, además de más atractiva visualmente. Para esto último, se ha utilizado la web “Figma” [12], la cual ofrece una herramienta de diseño de aplicaciones a través de un editor de arrastrar y soltar (Drag & Drop).

Por otro lado, para exportar los diseños creados con el editor e importarlos en la aplicación diseñada en forma de código Python, se ha hecho uso de la herramienta "Tkinter-Designer" [13], la cual permite transformar los diseños creados con el editor en código Python.

Para ello, a la herramienta el usuario únicamente tiene que pasarle 3 parámetros.

- **Token ID:** Token generado dentro de la propia cuenta de Figma para identificar al usuario.
- **File URL:** URL del diseño creado por el usuario.
- **Output Path:** Carpeta de destino en el ordenador para guardar el código y los demás recursos.

En la Ilustración 40 se puede observar el diagrama de funcionamiento de esta herramienta, de una forma clara.



Ilustración 40: Diagrama de funcionamiento de "Tkinter-Designer"

Haciendo uso de estas herramientas, se ha mejorado la interfaz gráfica con el fin de hacerla más agradable y atractiva para el usuario final, añadiéndole colores, una mejor organización, diseño, etc.

NOTA: el código descrito en este subapartado se puede encontrar en el módulo de la interfaz de usuario (**gui_module**) del trabajo final.

Dejando esto claro, al ejecutarse la aplicación final se podrá ver una nueva pantalla de inicio (Ilustración 41), con las mismas funcionalidades que tenía la interfaz funcional, pero con un diseño renovado.

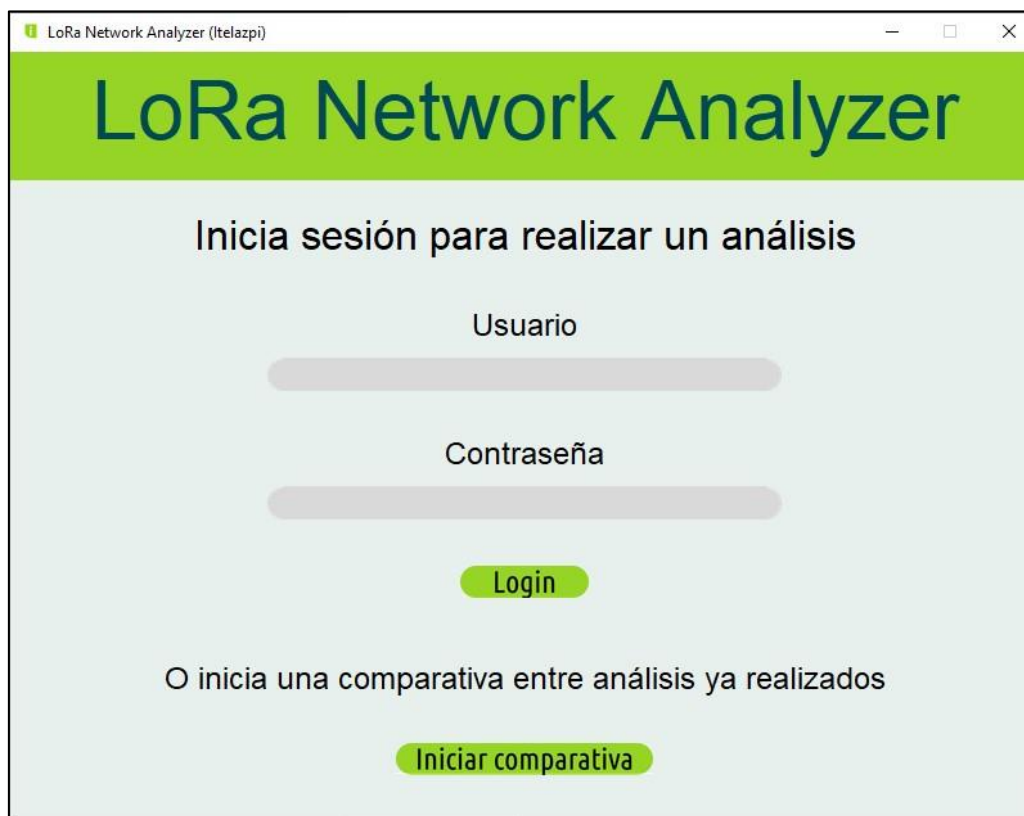


Ilustración 41: Pantalla principal (GUI final)

Comparando con la interfaz funcional, en esta pantalla se ha añadido el nombre de la aplicación en modo de portada, texto explicativo para cada acción y diseño en los botones y entradas de texto.

Además, a la ventana como tal, se le ha añadido el logo de Itelazpi (arriba a la izquierda) y se le ha puesto un nombre significativo a la ventana.

En cuanto a la funcionalidad, esta no ha cambiado. Se pueden introducir las credenciales del usuario donde se encuentran los dispositivos para realizar el análisis o pulsar el botón “Iniciar comparativa” para comparar diferentes análisis ya realizados.

Si se introducen las credenciales (unas correctas), se mostrará la siguiente pantalla (Ilustración 42), donde se podrán ver los grupos que existen en este usuario.



Ilustración 42: Pantalla de selección de grupo (GUI final)

En esta pantalla se ha ajustado un poco el tamaño de los elementos y se ha añadido texto explicativo, además de cambiar el diseño del botón.

Aquí se podrá filtrar y buscar el grupo deseado y una vez seleccionado se pasará a la siguiente pantalla (Ilustración 43), donde se seleccionarán las fechas de inicio y final del análisis.



Ilustración 43: Pantalla de selección de tiempo (GUI final)

En esta nueva pantalla se han centrado los elementos para dar una mejor imagen a esta, además de exponer la información obtenida (número de devices) de una forma clara y concisa.

Una vez se seleccionan las fechas para el análisis, en la siguiente pantalla (Ilustración 44) se podrá ver el progreso de obtención de los payloads de esas fechas de los devices del grupo elegido.

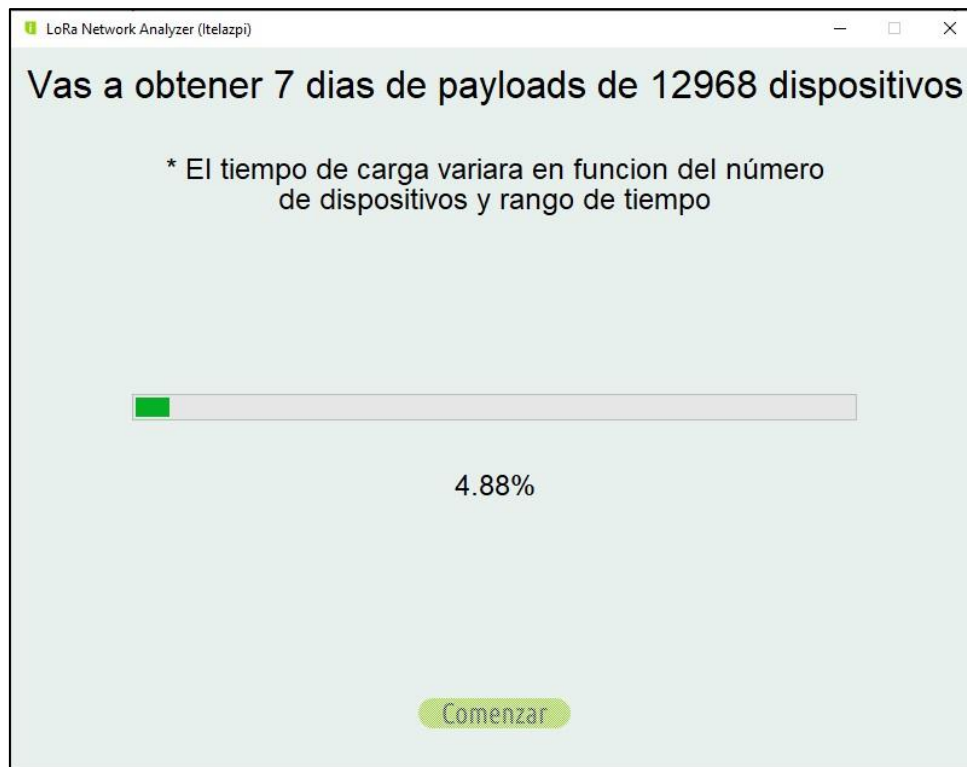


Ilustración 44: Pantalla de obtención de payloads (GUI final)

En esta pantalla se muestra ahora el tiempo seleccionado para el análisis, en días, y el número de dispositivos de los cuales se va a extraer información. Además, también se añade texto explicativo.

Una vez finalizada la obtención de los datos, se pasará a la siguiente pantalla (Ilustración 45), donde se le indicará al usuario que debe elegir un directorio para guardar los datos.

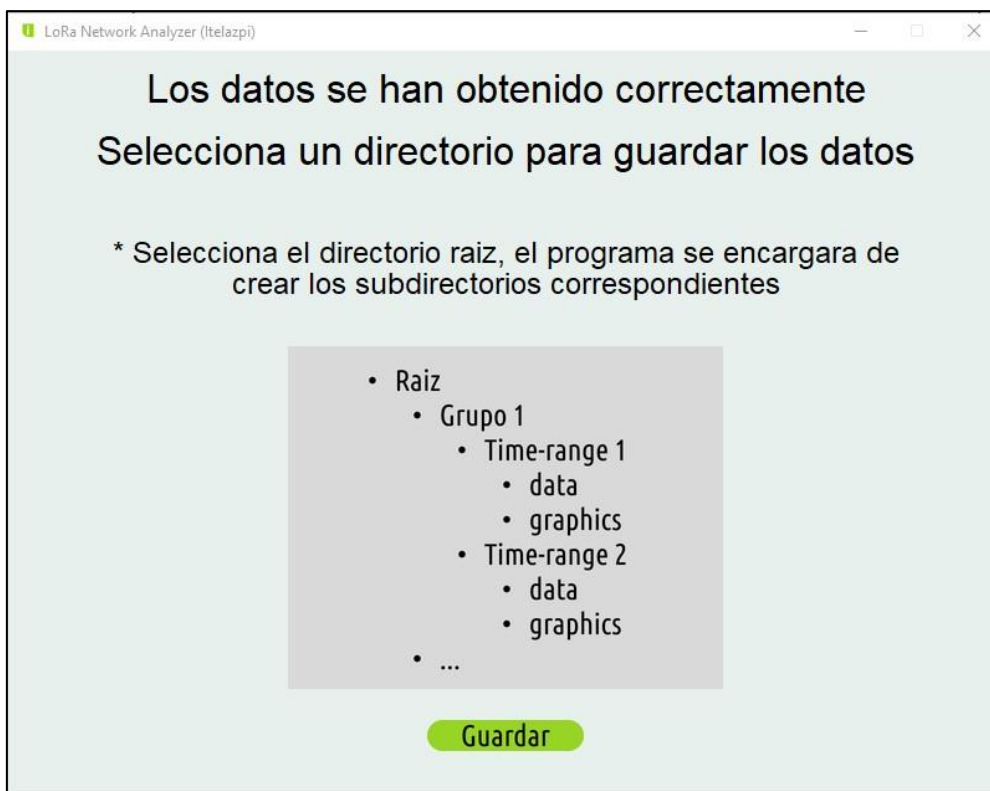


Ilustración 45: Pantalla de guardado (GUI final)

En esta nueva pantalla se ha añadido bastante información sobre cómo se van a generar los subdirectorios de los datos, ya que en la interfaz funcional esta información no se explicaba por ningún lado.

Por la parte de funcionalidad, es exactamente igual que lo que era en la interfaz funcional, por lo que, seleccionando un directorio raíz, se crearan los diferentes subdirectorios, separando, grupos, rangos de tiempo, datos y gráficos.

Una vez elegido el directorio raíz, se mostrará la siguiente pantalla (Ilustración 46) con la información del directorio seleccionado.

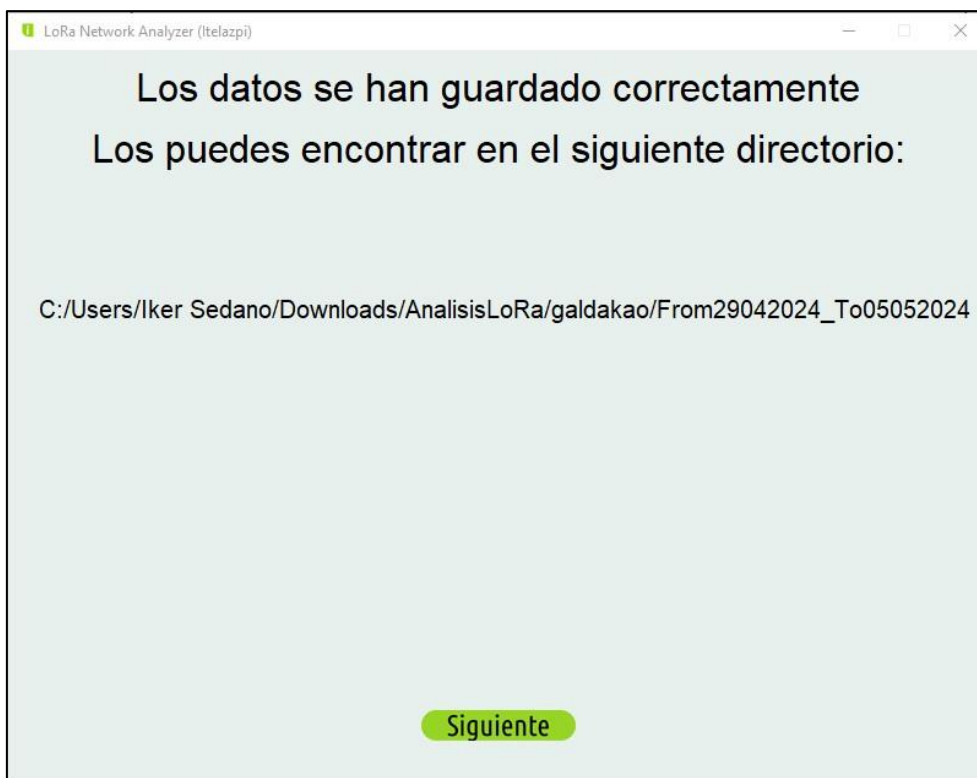


Ilustración 46: Pantalla de datos guardados (GUI final)

Al presionar el botón “Siguiete”, se pasará a la siguiente pantalla (Ilustración 47), donde se pedirán las credenciales del usuario administrador de los Gateways.

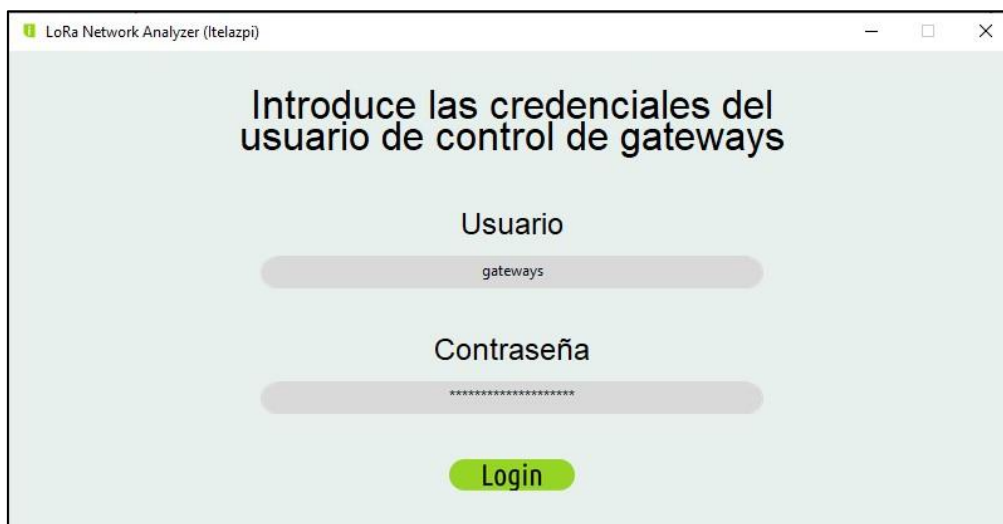


Ilustración 47: Pantalla de ingresar credenciales para los Gateways (GUI final)

Una vez introducidas las credenciales, y si son correctas, se mostrará una lista (Ilustración 48) con todos los Gateways administrados por ese usuario. De esa lista, se tendrán que elegir los Gateways deseados para realizar el análisis.



Ilustración 48: Pantalla con la lista de todos los Gateways (GUI final)

En esta pantalla únicamente se han hecho modificaciones de diseño y se ha añadido un texto explicativo comparado con la interfaz funcional.

Para elegir los Gateways deseados, se puede filtrar por nombre e ir seleccionando uno a uno que Gateways se quieren analizar, hasta haber añadido todos los deseados (Ilustración 49).



Ilustración 49: Pantalla con Gateways seleccionados (GUI final)

Una vez seleccionados los Gateways, se pasará a la siguiente pantalla (Ilustración 50), donde se extraerán los parámetros de los Gateways mientras en la interfaz se puede ver una barra de carga mostrando el progreso de este proceso.

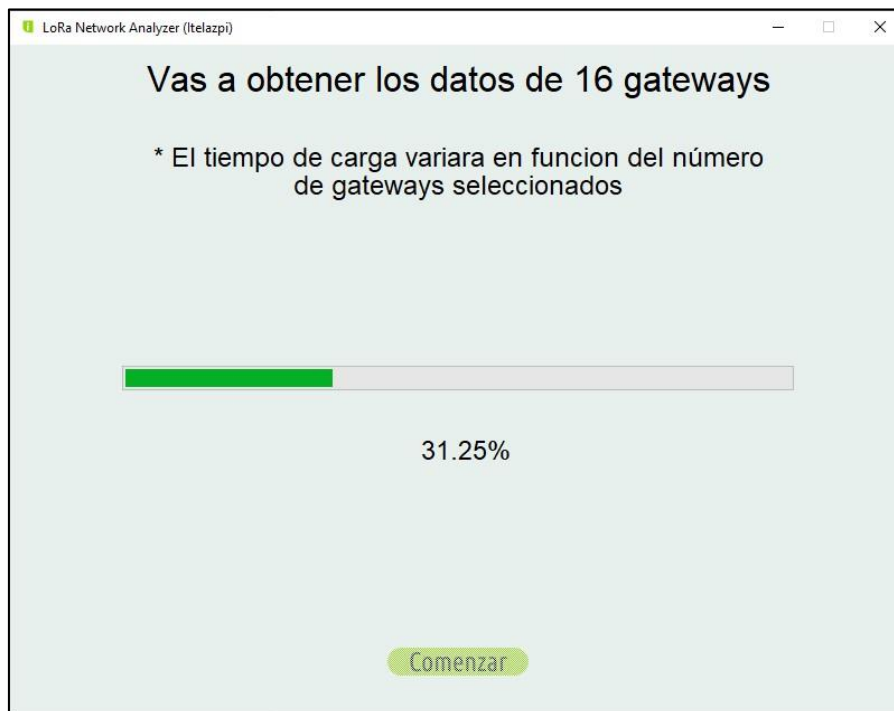


Ilustración 50: Pantalla de obtención de parámetros de Gateways (GUI final)

Al finalizar la obtención de estos datos, se pasará a la siguiente pantalla (Ilustración 51) donde se tratarán los datos y se generarán los gráficos, todo esto mientras el usuario vuelve a ver una barra de progreso.

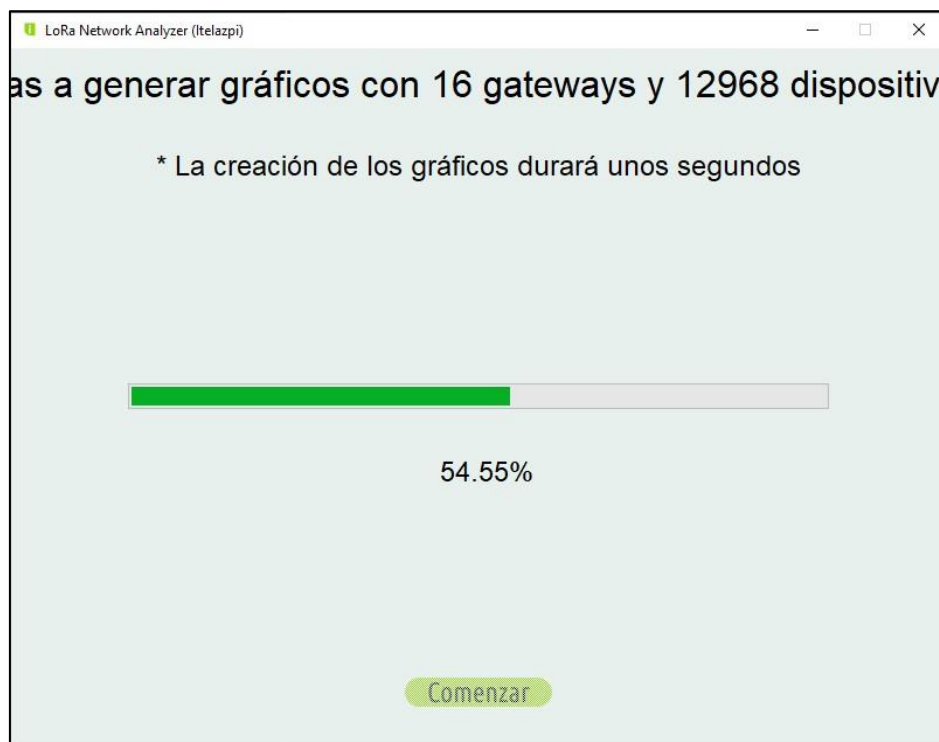


Ilustración 51: Pantalla de creación de gráficos (GUI final)

Al finalizar este proceso, se mostrará la pantalla final (Ilustración 52), donde se le dará al usuario la oportunidad de volver al inicio o de salir de la aplicación, al contrario que en la interfaz funcional, donde la pantalla final era meramente indicativa y no daba la opción de realizar ninguna acción.

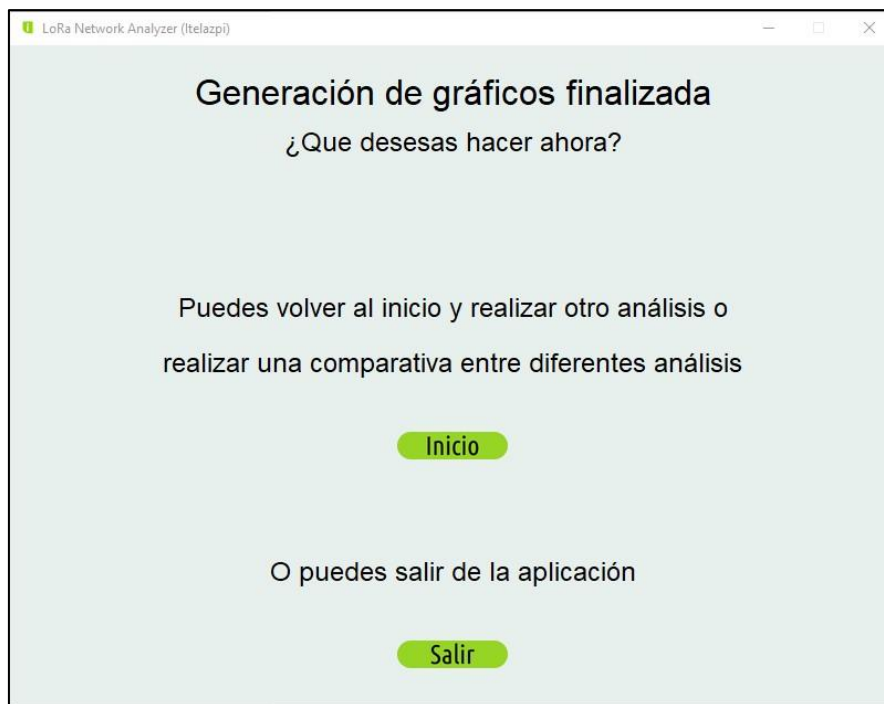


Ilustración 52: Pantalla de finalización (GUI final)

Una vez llegado a esta pantalla, el análisis de la red ya estaría finalizado. Pero, la aplicación da la opción de realizar la comparativa entre 2 o 3 análisis previamente realizados. Para ello, en la pantalla de inicio (Ilustración 41), se puede pulsar el botón "Iniciar comparativa", y esto llevara a la aplicación a la siguiente pantalla (Ilustración 53).

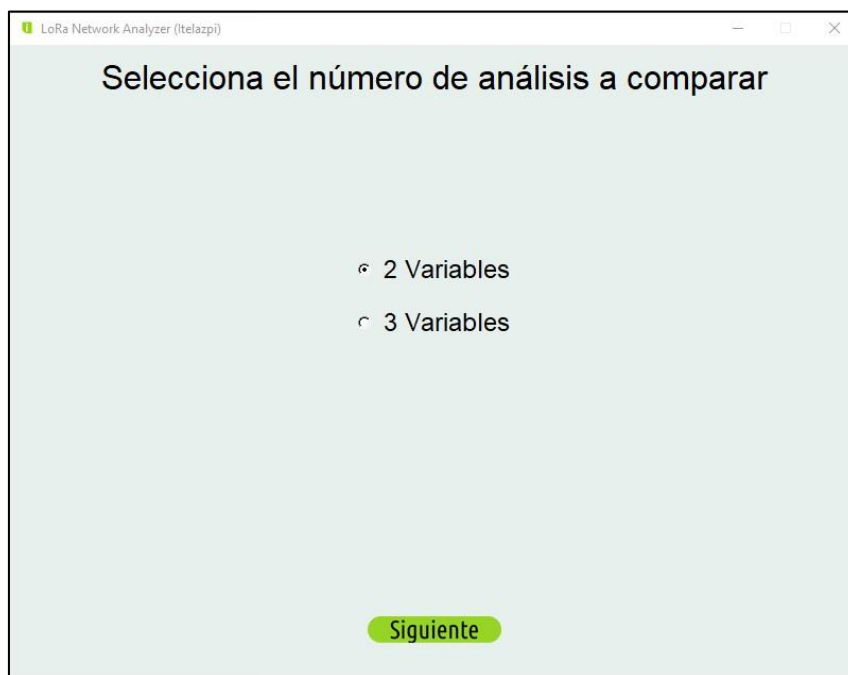


Ilustración 53: Pantalla de elección de numero de análisis (GUI final)

En esta pantalla adicional que no se encontraba en la interfaz funcional, se da la oportunidad al usuario de seleccionar si quiere hacer una comparación entre 2 o 3 análisis previamente realizados.

La única diferencia entre comparar 2 o 3 análisis es el número de rutas que ha de seleccionar el usuario para que la aplicación encuentre los datos de los análisis. El resto es exactamente igual para los dos casos, exceptuando el nombre de la carpeta donde se almacenarán los datos de la comparativa.

Para el caso de 2 variables, el nombre de la carpeta será el siguiente:

- **Compare_XXXXXXXXXXXX_to_AAAABBBB**

Y para el caso de las 3 variables, el nombre será ligeramente diferente:

- **Compare_XXXXXXXXXXXX_to_AAAABBBB_to_CCCDDDD**

Pero dentro de estas carpetas, la jerarquía de las subcarpetas donde se guardarán los gráficos será la misma.

Si se selecciona la opción de 3 variables, al usuario se le mostrará la siguiente pantalla (Ilustración 54).

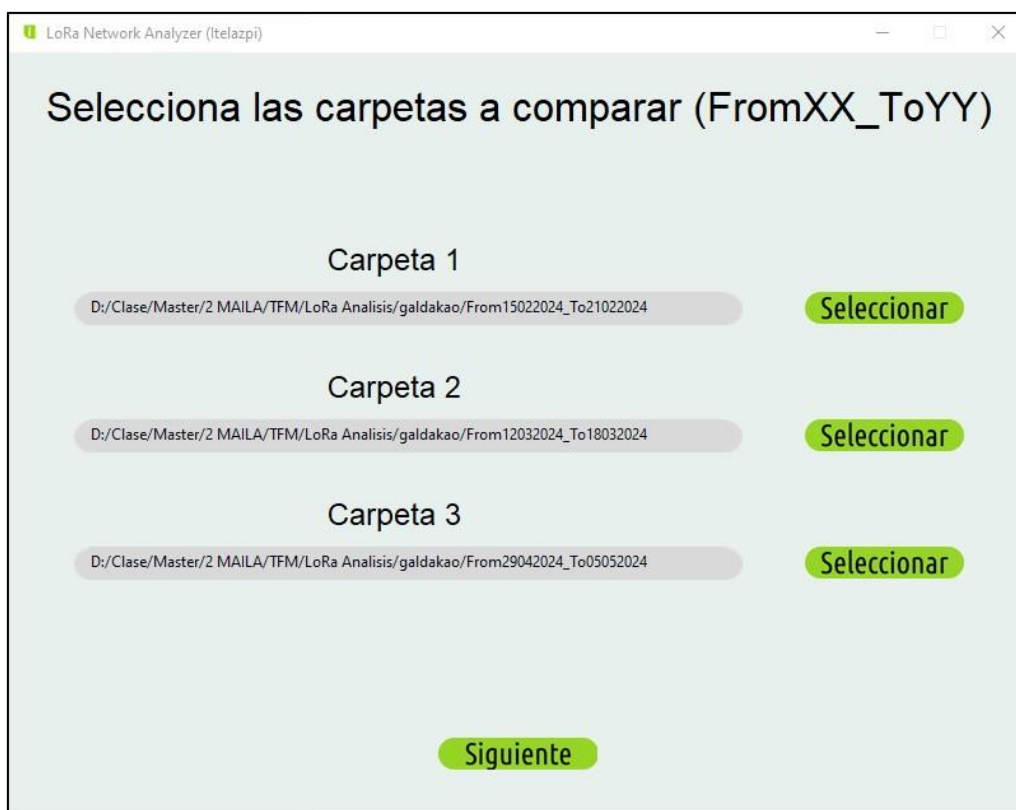


Ilustración 54: Pantalla de selección de carpetas (GUI final)

En esta pantalla inicialmente las rutas estarán vacías y será el propio usuario quien tendrá que seleccionar estas rutas a los diferentes análisis.

Una vez seleccionadas las 3 rutas y dándole a botón de “Siguiente”, la aplicación generará los gráficos unificando los datos de los diferentes análisis en uno solo para cada tipo de gráfico.

Mientras esto se lleva a cabo, el usuario podrá ver una barra de progreso (Ilustración 55) para poder ver el porcentaje de completado del proceso.

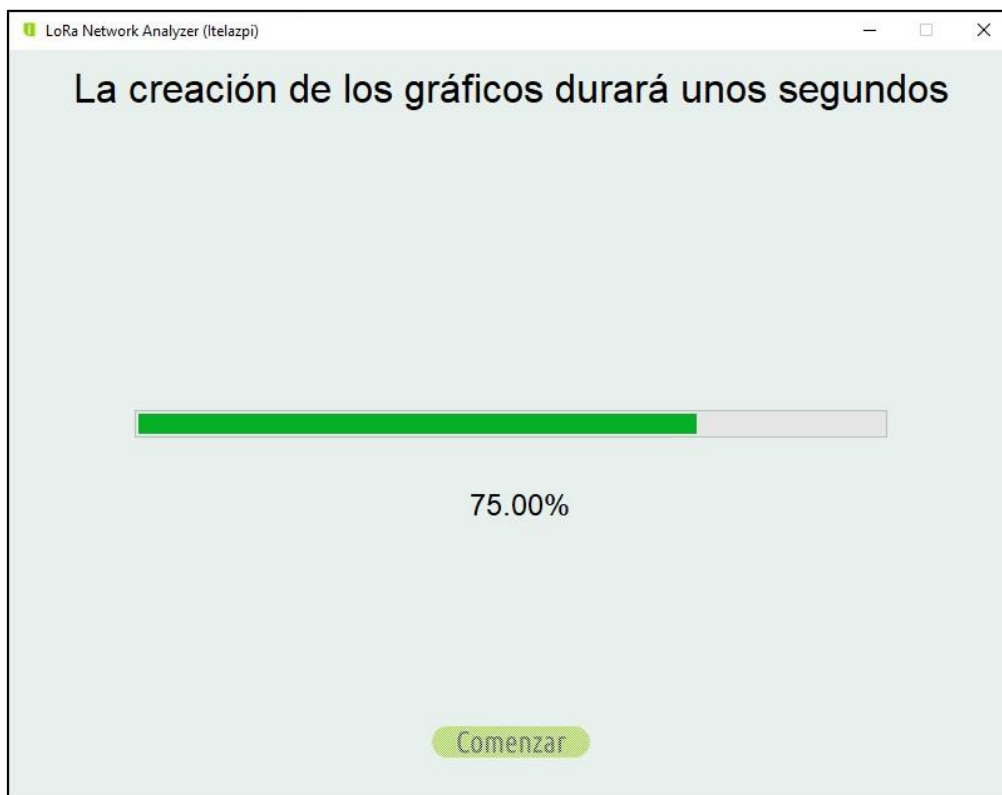


Ilustración 55: Pantalla de creación de gráficos combinados (GUI final)

Finalmente, cuando este proceso ha finalizado, se volverá a mostrar al usuario la pantalla de finalización (Ilustración 52), para que el usuario elija si quiere volver al inicio o salir de la aplicación.

Con esto, la parte del diseño de la interfaz gráfica final ya estaría finalizada, pero la aplicación sigue siendo puro código Python en varios scripts organizados y para poder ejecutarlo es necesario tener varias dependencias instaladas.

Para poder ejecutar esta aplicación en cualquier ordenador, se creará un instalador con el cual se instalará el programa y todas las dependencias necesarias en el ordenador de destino, dejándole al usuario un acceso directo a la aplicación en el cual con hacer clic se le ejecutará la aplicación completamente funcional.

Empaquetado de la aplicación

En cuanto al empaquetado de la aplicación, el primer paso es generar un ejecutable de todo el código Python que se ha ido creando, para que con hacer un único clic se pueda ejecutar toda la aplicación.

NOTA: los archivos necesarios para crear los elementos descritos en este subapartado se pueden encontrar en la carpeta de elementos de instalación (**Setup**) del trabajo final.

Para ello, lo primero de todo es instalar el programa Git Bash [14], ya que este permite ejecutar comandos Linux sobre el Windows donde se está trabajando. Es necesario instalar esta nueva terminal ya que, para crear un ejecutable capaz de realizar web scraping, es necesario ejecutar

ciertos comandos extra, los cuales no serían necesario con una aplicación Python que no hiciese web scraping.

Para empezar con el empaquetado, el primer paso será crear un entorno virtual de Python donde se instalarán todos los módulos necesarios para el funcionamiento de la aplicación. El comando para crear el entorno virtual es el siguiente:

```
python3 -m venv {nombre_entorno_virtual}
```

Una vez creado, se entra dentro del entorno y se activa el entorno con un script ya predefinido:

```
. Scripts/activate
```

Después de activar y poner en marcha el entorno virtual, se moverán todos los scripts y de Python creados a este entorno, y se instalarán los módulos necesarios para su funcionamiento (excepto el módulo de web scraping) con el siguiente comando:

```
pip install {modulo}
```

Para el módulo de web scraping, playwright [15], antes de instalarlo, es necesario cambiar una variable de entorno para que el ejecutable integre el navegador y así no depender de los navegadores del equipo de destino. El comando a ejecutar es el siguiente:

```
PLAYWRIGHT_BROWSERS_PATH=0 playwright install chromium
```

Con esto, ya se podría crear el ejecutable, pero para poder hacerlo, es necesario el módulo pyinstaller [16]. Con este módulo se ejecutará el siguiente comando para crear el ejecutable:

```
pyinstaller -F --onefile --hidden-import babel.numbers --windowed --name 'LoRa Network Analyzer' --icon=icon.ico main.py
```

Si todo sale correctamente, se podrá ver lo siguiente en la línea de comandos (Ilustración 56):

```
182549 INFO: Copying icon to EXE
182560 INFO: Copying 0 resources to EXE
182560 INFO: Embedding manifest in EXE
182605 INFO: Appending PKG archive to EXE
182718 INFO: Fixing EXE headers
183625 INFO: Building EXE from EXE-00.toc completed successfully.
```

Ilustración 56: Resultado de crear el ejecutable

Entrando en detalle en lo que hace cada parámetro:

- “-F”: Indica que se cree un único archivo ejecutable en lugar de un directorio con múltiples archivos.
- “--onefile”: Similar a “-F”. También indica que se cree un único archivo.
- “--hidden-import babel.numbers”: Le indica que instale babel.numbers de forma oculta, ya que hay dependencias que no están siendo detectadas automáticamente. Este módulo es necesario para que los calendarios de selección de fechas funcionen.
- “--windowed”: Esta opción le indica que se desea que la aplicación se ejecute en modo ventana, es decir, sin abrir una ventana de línea de comandos junto con la aplicación.
- “--name ‘LoRa Network Analyzer’”: Le especifica el nombre del archivo ejecutable resultante.
- “--icon=icon.ico”: Le establece un icono al archivo ejecutable en vez del que le pone por defecto.
- “main.py”: El nombre del script de Python principal que se convertirá en ejecutable.

Una vez la ejecución del comando se ha completado, se habrá creado un ejecutable (LoRa Network Analyzer.exe, Ilustración 57), con el icono que se había predefinido.

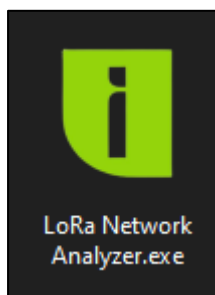


Ilustración 57: Ejecutable de la aplicación

Este archivo ejecutable únicamente funciona bien en la máquina en la que se ha creado, ya que, para su funcionamiento, está obteniendo recursos de otros directorios. Para que el ejecutable funcione en cualquier otro ordenador, habría que mover a esos ordenadores tanto el ejecutable como el directorio de recursos.

Esto, por mucho que sea funcional, no es demasiado profesional, ya que, cualquier programa que se instala en el ordenador se suele hacer a través de un archivo “setup”, con el cual se instalan todos los elementos necesarios.

Con el fin de imitar esta práctica y crear una aplicación más profesional, se utiliza el programa “Inno Setup” [17], el cual permite crear un único archivo de setup para poder instalar la aplicación y todos los recursos necesarios.

Para poder crear el archivo setup, son necesarios los siguientes elementos:

- **Nombre de la aplicación:** el nombre con el que se instalará.
- **Archivo ejecutable:** el de la aplicación que se ha desarrollado.
- **Recursos extra:** recursos necesarios para el correcto funcionamiento de la aplicación.
- **Licencia de la aplicación creada:** documento legal del uso que se le puede dar a la aplicación.
- **Nombre del setup:** el nombre del archivo setup.
- **Icono del archivo setup:** icono con el que se creará el archivo de setup.

Con estos elementos y siguiendo los pasos, se obtendrá el archivo setup que se podrá mover a cualquier ordenador y al ejecutarlo y seguir los pasos indicados, se instala la aplicación que ha sido desarrollada para realizar análisis de la red LoRa de Itelazpi.

5.2.5. Corrección de errores y casos límites

A lo largo del tiempo que se iba creando la interfaz funcional y la aplicación final, se han ido encontrando varios errores o mejoras posibles para que el funcionamiento de la aplicación sea el óptimo.

Las correcciones más significativas se pueden agrupar en 4 grupos:

- Obtención del **endpoint correcto** dependiendo la red en la que se encuentre el usuario.
- Manejo de **excepciones** de fallos de **autenticación**.
- Aplicación **multihilo**.
- Manejo de **casos límites** con **falta de datos**.

En los siguientes subapartados se explicará más detalladamente cada una de estas correcciones.

Obtención de endpoint

En un principio, la aplicación ha sido desarrollada en un ordenador, pero esta aplicación tiene que ser ejecutable en cualquier ordenador de Itelazpi, se encuentre donde se encuentre.

Teniendo esto en cuenta, se han detectado diferentes posibilidades en donde el ordenador puede encontrarse en diferentes redes, por lo que es necesario garantizar la conectividad desde esas redes a los endpoints de las APIs utilizadas.

Las diferentes redes donde podría encontrarse el ordenador son las siguientes:

- Red cableada de Itelazpi (salida a internet por EJIE)
- Red WiFi de Itelazpi (salida por la IP pública de Itelazpi)
- VPN de Itelazpi
- Resto de casos

Estos 4 casos se pueden agrupar en dos grupos. Por una parte, las redes conocidas de Itelazpi (red de EJIE, WiFi de Itelazpi y VPN de Itelazpi), y por otra, las redes desconocidas.

En el primer grupo, al ser estas redes conocidas, se puede configurar la aplicación para detectar el prefijo de las direcciones IP de esas redes y utilizar un endpoint para este caso. El endpoint en cuestión será el accesible únicamente desde la IP privada, ya que estas redes tienen acceso al direccionamiento privado de los servidores donde está instalado el Network Server de Orbiwise.

En el segundo grupo, como en este se engloban el resto de opciones, se opta por asignarle el direccionamiento público, ya que este será accesible desde la gran mayoría de redes.

Excepciones de fallo de autenticación

Al estar desarrollando la aplicación en un entorno controlado y cerrado, el fallo de autenticación por la introducción errónea de credenciales por parte del usuario no se tuvo en cuenta. Esto hizo que la aplicación no supiese manejar esas excepciones y que al introducir unas credenciales incorrectas la aplicación se quedase congelada, teniendo que cerrarla y volver a abrir para poder volver a trabajar con ella.

Esto sumado a que no se guarda el estado del análisis que se está llevando a cabo, hacia que, si se erraba en la inserción de credenciales del usuario administrador de los Gateways, se perdía todo el progreso realizado hasta el momento.

Para evitar que esto pase, se ha tenido en cuenta que puede ser que las credenciales pueden no ser correctas, y para indicarle al usuario que debe volver a introducir de nuevo las credenciales, de forma correcta esta vez, la aplicación lanza una ventana emergente (Ilustración 58) con esta información.

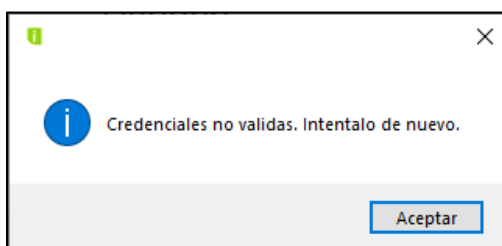


Ilustración 58: Ventana emergente al introducir credenciales erróneas

Programación multihilo

La aplicación desarrollada tiene como objetivo realizar diferentes análisis de la red LoRa de Itelazpi, y para ello, necesita una gran cantidad de datos para obtener resultados.

La obtención de estos datos es manejada por la API de Orbiwise, pero al ser tal la cantidad (para el análisis de Galdakao, 7 días de datos de 13.000 dispositivos), el tiempo de obtención de estos es muy significativo (entre 30 y 40 minutos para Galdakao).

Por ello, en la aplicación se implementan las barras de carga, para que el usuario pueda observar el progreso realizado hasta el momento. Pero, al implementar esto, la aplicación tiene que ser capaz de manejar la interfaz gráfica y las llamadas a la API de forma simultánea, ya que, si no, la aplicación entraría en modo “No responde” (ya que la interfaz gráfica se queda congelada) por mucho que las llamadas a la API se sigan ejecutando.

Para el usuario final, si no sabe el funcionamiento de la aplicación, le da la sensación de que la aplicación se ha colgado, por mucho que no sea así. Es por eso que, para evitar esta situación, en los casos que la aplicación debe hacer llamadas a la API, se hacen desde un hilo diferente, para que la interfaz gráfica siga funcionando y el usuario pueda observar que la aplicación está trabajando y mostrando el progreso.

Casos límite de falta de datos

Al igual que se ha explicado con las excepciones de autenticación, esta aplicación se ha desarrollado en un entorno cerrado y siempre utilizando los mismos datos, los necesarios para un análisis en Galdakao. Pero, función de la aplicación debe ser aplicable a todos los escenarios posibles, es decir a los diferentes municipios en los que Itelazpi cuenta con cobertura LoRa y sensores.

Por ello, es posible que, en los diferentes casos, falten ciertos datos que hacen que la aplicación deje de funcionar correctamente y que no se generen los resultados deseados, o ningún resultado en el peor de los casos.

Tras realizar diferentes análisis y pruebas a la aplicación, los casos donde pueden faltar datos que han sido corregidos son los siguientes:

- Sin grupos en el usuario con el que se ha hecho login.
- Grupo no seleccionado para realizar el análisis.
- Sin datos de los devices en las fechas seleccionadas.
- Sin Gateways con el usuario administrador de Gateways (login con otro usuario).
- Gateways no seleccionados para realizar el análisis.
- Sin datos de los Gateways en las fechas seleccionadas.
- Parámetros faltantes para realizar algún gráfico.
- Archivos JSON no encontrados para realizar comparativas.

Al realizar estas correcciones, se pretende maximizar la eficacia de la aplicación y hacer que el usuario pueda utilizarla de una manera sencilla, sin miedo a equivocarse y tener que volver a empezar desde el principio.

5.3. Estudio inicial

En este tercer apartado se analizará la red tal y como se configura por defecto, con el fin de ver el estado base de la red y poder determinar los cambios que podrían ser beneficiosos para aumentar la calidad de esta.

Para el análisis, se obtendrán los parámetros tanto de los Gateways como de los devices, haciendo uso de los scripts que se han preparado con anterioridad. Una vez obtenidos esos datos, se les aplicará un procesado y se generarán los gráficos predefinidos, estos también, con los scripts que se habían preparado.

Finalmente, se hará una interpretación de estado actual de la red analizando los gráficos, y se valorarán cuáles han sido los resultados obtenidos y cuál es la calidad de la red tras este primer análisis.

En resumen, en este bloque se explicarán en profundidad los siguientes subapartados:

- Obtención de los datos de los Gateways
- Obtención de los datos de los devices
- Procesado de datos
- Generación de gráficos
- Interpretación del estado de la red
- Valoración de los resultados iniciales

5.3.1. Obtención de los datos de los Gateways

Con el fin de realizar un análisis completo de la red, se obtendrán los parámetros de los Gateways seleccionados, es decir, los que se encuentran en Galdakao, ya que el análisis y las optimizaciones se harán en ese municipio.

En la red LoRa de Itelazpi, existen 16 Gateways que se encuentran en Galdakao, pero solo 15 de ellos se encontraban activos en el rango de fechas seleccionadas para la obtención de parámetros, por lo que, el análisis contará únicamente con 15 Gateways, ya que el decimosexto devolverá datos vacíos.

Habiendo definido los Gateways de los que se quiere obtener la información, haciendo uso de los scripts, se creará un archivo JSON con la información estática de todos los Gateways seleccionados, la que se puede ver en la Ilustración 59.

```
[  
  {  
    "altitude": 0,  
    "id": "000800fffe4bc4bb",  
    "latitude": 43.2346701487839,  
    "longitude": -2.85004080332071,  
    "name": "Galdakao Aranzelai, 1 indoor",  
    "status": "OK"  
  },  
]
```

Ilustración 59: Información estática de un Gateway

El rango de tiempo seleccionado para el análisis ha sido de 7 días, ya que se ha considerado tiempo suficiente para realizar medias y cálculos estadísticos de los parámetros obtenidos, sin llegar a sobrecargar a la aplicación de datos, ya que esto la ralentizaría.

Para la obtención de los datos, se utilizarán los scripts previamente creados, y con ellos se obtendrá un archivo JSON con la información de cada Gateway (Ilustración 60).

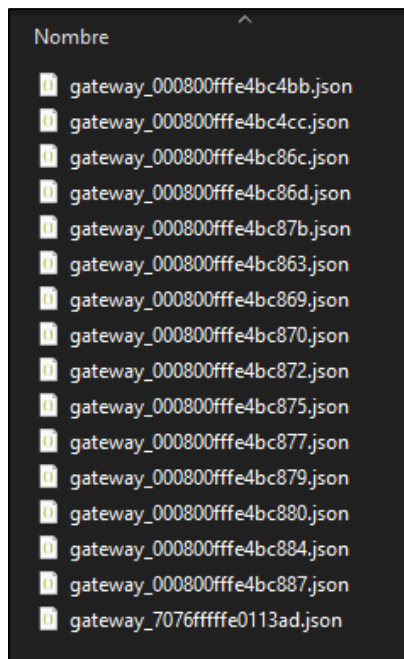


Ilustración 60: Archivos JSON de los parámetros de cada Gateway

Dentro de cada uno de estos archivos, se puede encontrar la información necesaria para el posterior análisis de la red, como se puede observar en la Ilustración 61.

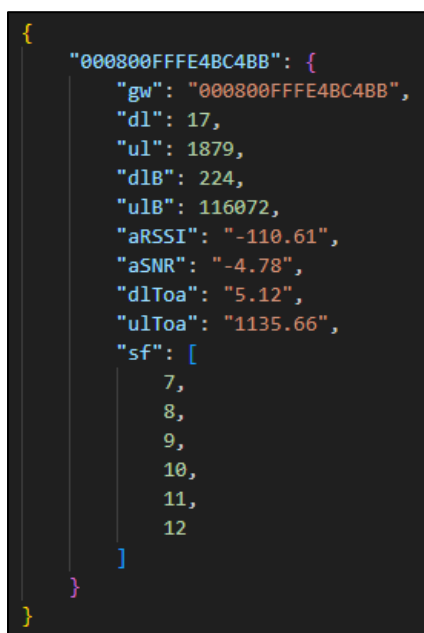


Ilustración 61: Ejemplo de datos de calidad de señal obtenidos de un Gateway

Con estos datos, se puede dar por concluida la obtención de los parámetros de los Gateways en la fase inicial.

5.3.2. Obtención de los datos de los devices

A la vez que se realiza la obtención de los datos de los Gateways, también se obtienen los parámetros de los devices, ya que se pretende usar la misma franja de tiempo (los mismos 7 días) para analizar los parámetros y hacer los cálculos estadísticos pertinentes.

Para obtener los datos necesarios para el análisis, se utilizarán los scripts previamente creados, con los cuales se obtendrán 2 archivos JSON. El primero de ellos contará con la información estática de los devices, la que se puede observar en la Ilustración 62.

```
[
  {
    "deveui": "0007813700003e93",
    "devaddr": "289394771",
    "appeui": "000781370000001",
    "joineui": null,
    "comment": "BEREZIKOETXE KALEA 007 003 C",
    "latitude": 43.228618,
    "longitude": -2.845771,
    "altitude": null,
    "device_status": 3,
    "dl_fcmt": 160,
    "lora_device_class": 0,
    "registration_status": 1,
    "last_reception": "2024-02-22T12:27:24.493Z",
    "groups": "galdakao",
    "applications": "cabb_temetra_https",
    "tags": {
      "connection_ref": "1702676973",
      "meter_serial": "I20IA318844"
    },
    "creation_time": "2022-04-12T06:41:49.706Z",
    "service_profile_uuid": "5a8a51db-3013-4374-8975-b129c1c68719",
    "device_profile_uuid": "5d2bb0e7-a7b4-48ed-8f83-476dc8e43958",
    "activated": true,
    "suspended": false,
    "battery_level": 254,
    "device_snr": 20,
    "battery_level_update_time": "2023-12-29T03:38:56.415Z",
    "RFRegion": "EU868",
    "traffic_status": "NOMINAL",
    "traffic_rate_ul_per_day": 3.6
  },
]
```

Ilustración 62: Información estática de un device

El segundo archivo contiene los parámetros de calidad de los mensajes transmitidos durante los 7 días seleccionados (Ilustración 63).

```
[
  {
    "deveui": "000781370003e93",
    "data": [
      {
        "dr_used": "SF9BW125",
        "fcnt": 3255,
        "rssi": -112,
        "snr": 0.5,
        "time_on_air_ms": 370,
        "main_gw": "000800fffe4bc87b",
        "gws_reached": 5
      },
      {
        "dr_used": "SF9BW125",
        "fcnt": 3256,
        "rssi": -113,
        "snr": -1,
        "time_on_air_ms": 370,
        "main_gw": "000800fffe4bc872",
        "gws_reached": 1
      },
      {
        "dr_used": "SF9BW125",
        "fcnt": 3257,
        "rssi": -103,
        "snr": -6.5,
        "time_on_air_ms": 370,
        "main_gw": "000800fffe4bc872",
        "gws_reached": 1
      }
    ]
  }
]
```

Ilustración 63: Información de cada mensaje transmitido de un device

Con todos estos datos conseguidos de los devices, se puede dar por finalizada la obtención de los parámetros, ya que, con estos dos archivos JSON creados, es posible realizar un análisis muy completo de la calidad de la red.

5.3.3. Procesado de datos

Una vez obtenidos los archivos con los datos “en crudo” tanto de los Gateways como de los devices, es necesario hacerles un tratamiento para seleccionar los datos realmente importantes para el análisis, así como obtener ciertos parámetros derivados de los originales, como pueden ser medias, máximos, etc.

Por la parte de los Gateways, se unifican los datos estáticos con la información de la calidad de señal que han recibido durante los 7 días y se genera un único archivo con toda la información necesaria para la generación de gráficos relacionados a los Gateways (Ilustración 64).

```
[  
  {  
    "Gateway_ID": "000800fffe4bc4bb",  
    "name": "Galdakao Aranzelai, 1 indoor",  
    "status": "OK",  
    "altitude": 0,  
    "latitude": 43.2346701487839,  
    "longitude": -2.85004080332071,  
    "UL_Frame_Cnt": "1135.66",  
    "Lost_Frame_Cnt": 0,  
    "Avr_RSSI_[dBm]": -110.61,  
    "Avr_SNR_[dB]": -4.78  
  },  
]
```

Ilustración 64: Ejemplo de datos unificados de un Gateway

Por la parte de los devices, el procedimiento seguido es similar, creando un único archivo JSON (Ilustración 65) que contiene tanto la información estática de cada device como las estadísticas obtenidas a través de los datos de calidad de señal durante los datos de los 7 días obtenidos.

```
[  
  {  
    "deveui": "0007813700003e93",  
    "latitude": 43.228618,  
    "longitude": -2.845771,  
    "average_rssi": -108.79166666666667,  
    "average_snr": 1.8208333333333335,  
    "average_time_on_air_ms": 370.79166666666667,  
    "average_gws_reached": 4.625,  
    "main_gw": "000800fffe4bc872",  
    "lost_frames": 8,  
    "received_frames": 24,  
    "last_SF_used": "SF9",  
    "gateway_id_mas_cercano": "000800fffe4bc87b",  
    "distancia_entre_dispositivo_y_gateway_mas_cercano": 351.51972535675793,  
    "main_gw_id": "000800fffe4bc872",  
    "distancia_entre_dispositivo_y_main_gw": 644.5853176955999  
  },  
]
```

Ilustración 65: Ejemplo de datos unificados de un device

Con estos dos archivos JSON creados, es posible generar los gráficos que se había definido al inicio del proyecto, para así poder analizar la calidad de la red durante los 7 días en los que se ha realizado el análisis.

5.3.4. Generación de gráficos

Habiendo tratado ya los datos y generado los archivos JSON con la información necesaria tanto de los Gateways como de los devices, haciendo uso de los scripts ya preparados, se empiezan a generar los gráficos predefinidos.

Empezando por el mapa, en la Ilustración 66 se puede observar como este ha sido creado añadiendo los Gateways como etiquetas y los devices como simples figuras circulares (con el fin de no sobrecargar el propio mapa y que la experiencia de usuario sea buena). Además, el mapa se ha centrado automáticamente en la zona de Galdakao, como se había predefinido.

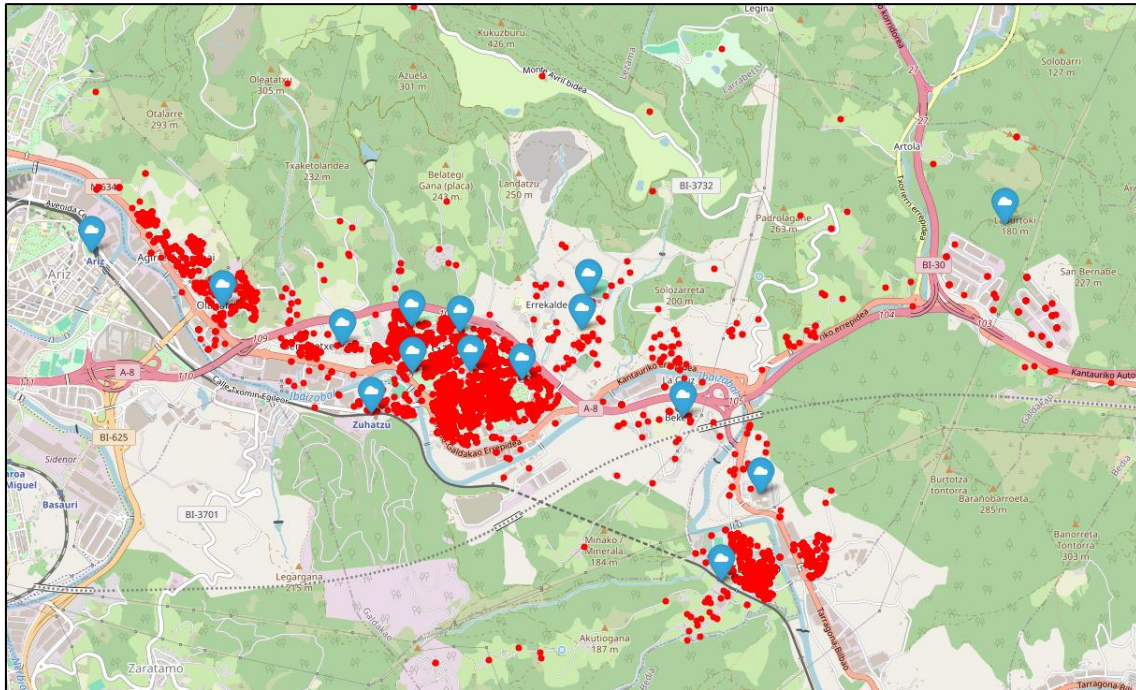


Ilustración 66: Mapa de Gateways y devices de Galdakao

Si se hace zoom en la imagen, se puede observar cómo están repartidos los devices a lo largo de los edificios. Además, haciendo clic en las etiquetas de los Gateways, se puede obtener información extra (nombre del Gateway) con el fin de dar facilidades a la hora de identificarlos. Todo esto se puede ver en la Ilustración 67.

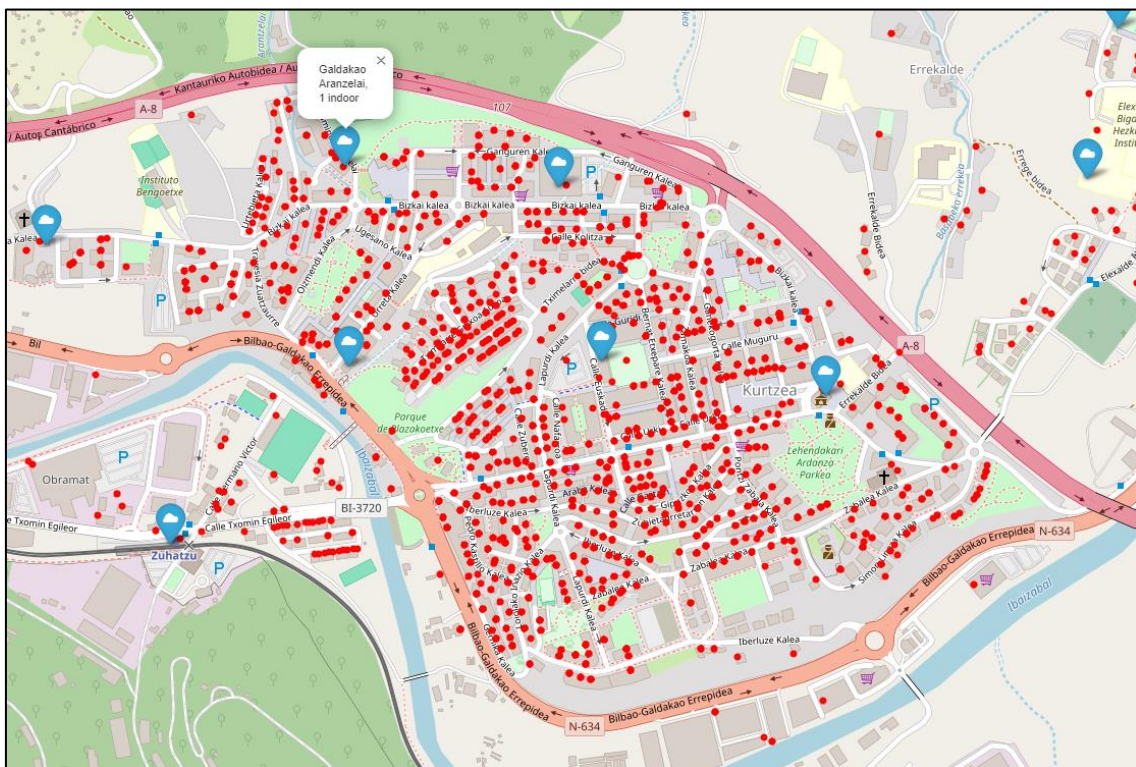


Ilustración 67: Zoom del mapa de Gateways y devices de Galdakao

Este mapa puede servir para hacer una inspección visual de donde están situados los devices y Gateways, y así hacer una aproximación de por dónde puede faltar cobertura LoRa o donde puede ser necesario añadir otro Gateway para darle más capacidad a la red.

En cuanto a los gráficos asociados a los Gateways, se han generado como se esperaba utilizando los scripts previamente creados. Como ejemplo, en la Ilustración 68 se puede observar cuantos mensajes de uplink ha recibido cada Gateway durante el transcurso de los 7 días, para así poder identificar cuáles son los Gateways con más tráfico.

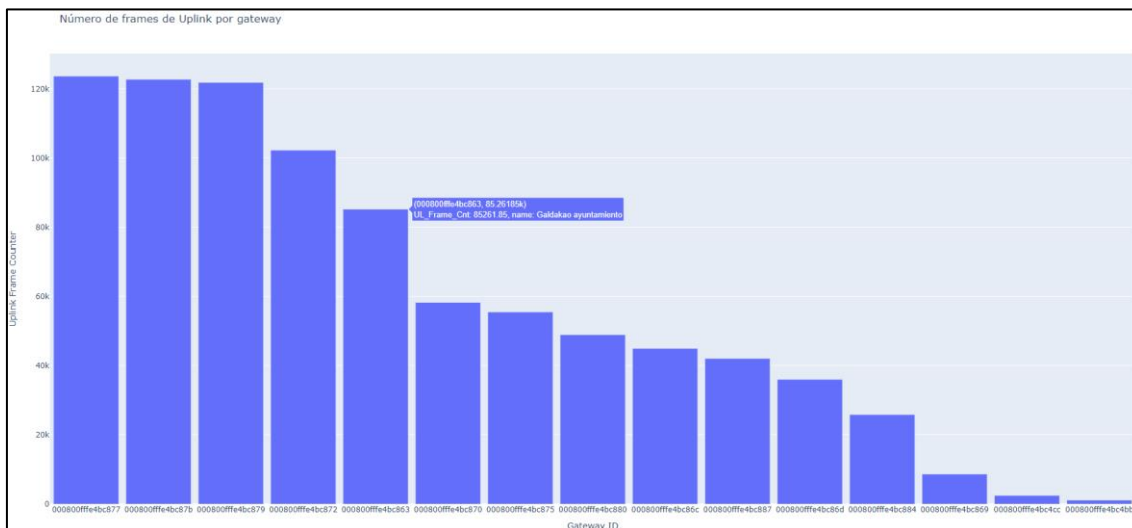


Ilustración 68: Gráfico de número de frames de uplink por Gateway (estudio inicial)

Además, para facilitar la comprensión del gráfico, se ha añadido información extra a cada barra, a la cual se puede acceder fácilmente pasando el ratón por encima de la barra deseada. En esta información adicional se ha añadido el nombre del Gateway (para identificarlo más fácilmente que con el identificador único de Gateway) y el número exacto de frames de uplink.

El resto de gráficos se han obtenido de la misma manera, ejecutando los scripts que se habían creado anteriormente y alimentándolos con los datos ya tratados tanto de los Gateways como de los devices.

Los subgrupos de gráficos que también se han generado son las distribuciones de los parámetros de los devices (Ilustración 69), los que ordenan los datos dependiendo la distancia al Main Gateway y los que los ordenan dependiendo cual ha sido el último SF que han utilizado.

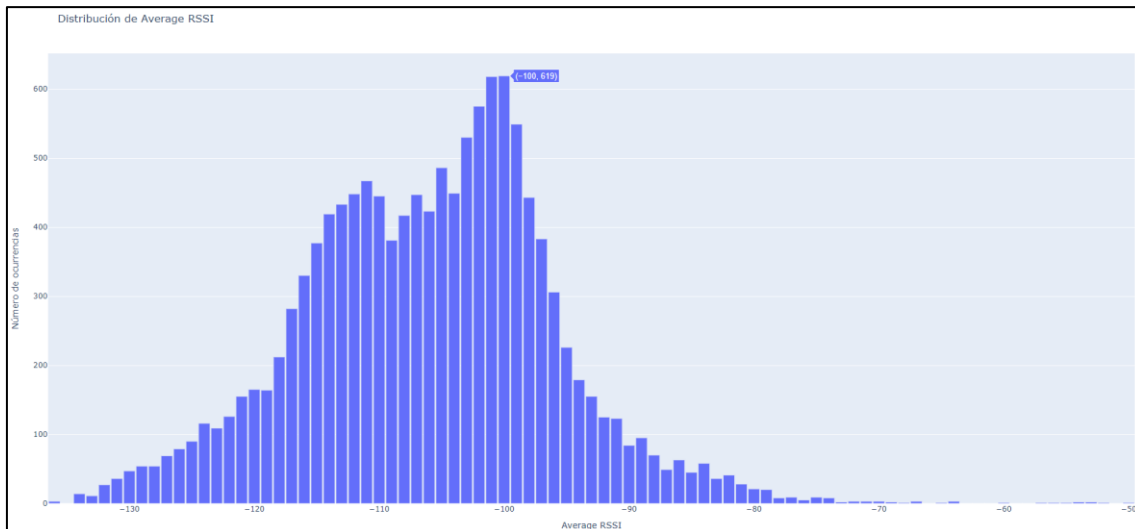


Ilustración 69: Grafico de la distribución de la media de RSSI (estudio inicial)

5.3.5. Interpretación del estado de la red

En este punto del proyecto, se ha conseguido obtener todos los gráficos que se habían predefinido para realizar el estudio preliminar de la red LoRa de Itelazpi. Analizando cada uno de estos gráficos y también en conjunto, se puede obtener una imagen bastante precisa del estado de la red.

Del análisis realizado, no todos los gráficos contienen información relevante, o por lo menos no en este primer análisis, por lo que, en este apartado, se expondrán los gráficos más significativos para el análisis preliminar.

Empezando por las diferentes distribuciones de datos, si se observa el gráfico de a cuantos Gateways llega cada device de media en los 7 días del análisis (Ilustración 70), se puede una gran parte de devices (947) solo llegan a un único Gateway, lo que, en caso de que un mensaje no llegue a ese Gateway, supondría la pérdida total de esos datos.

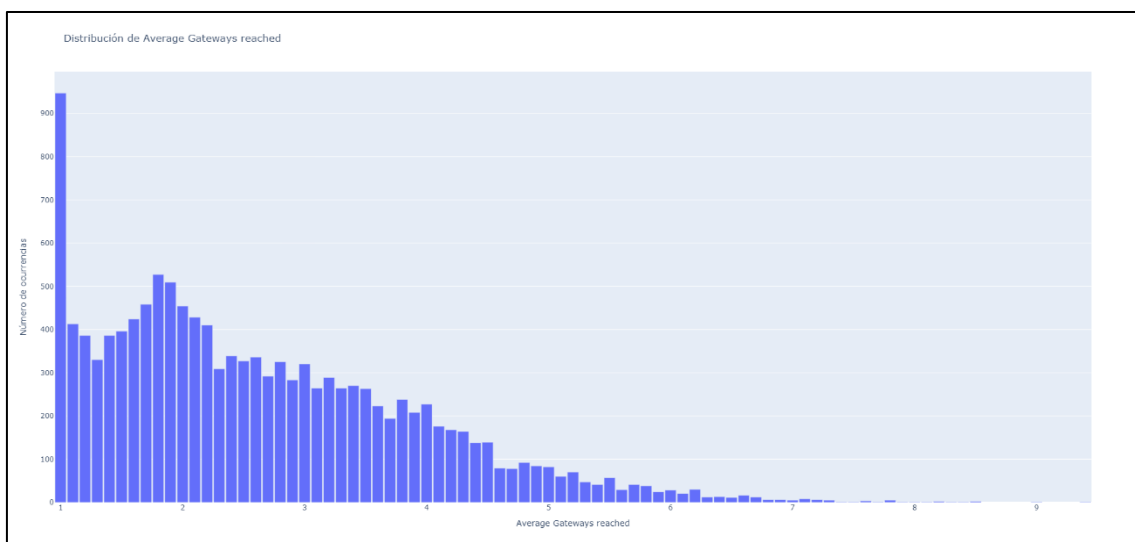


Ilustración 70: Distribución de "Average Gateways reached" (estudio inicial)

Por otro lado, si se observa el gráfico que muestra en que SF están enviando mensajes los devices (Ilustración 71), se puede observar que aproximadamente la mitad (6386 de 12968, un 49,1%

de los devices) utilizan SF7, es decir el modo más eficiente ya que significa que el device está cerca de un Gateway y puede transmitir a menor potencia, ahorrando batería, y además reducir el tiempo en el aire (ToA) de cada mensaje, reduciendo las posibilidades de colisiones.

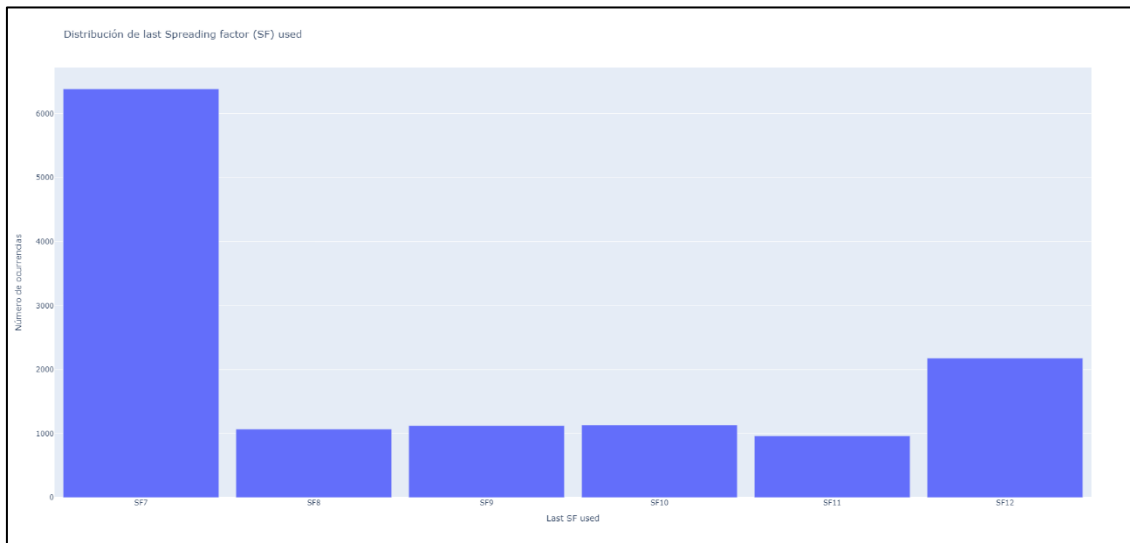


Ilustración 71: Distribución de "Last SF used" (estudio inicial)

Continuando con las distribuciones, si se analizan las gráficas de mensajes perdidos y recibidos, se puede observar que son inversamente proporcionales, ya que el mismo número de devices que ha perdido 0 mensajes, ha recibido el máximo de ellos, en este caso 32.

En estos gráficos se puede observar que cuanto mayor es el número de mensajes perdidos, menor es el número de devices, siendo 10 mensajes perdidos (22 recibidos) un punto en el que el resto de devices que han perdido más mensajes que estos se puede considerar mínimo y normal, por faltas de cobertura.

Además, dependiendo el formato en el que cada fabricante envíe los datos, es posible que la pérdida de tantos mensajes no sea fatal, ya que, si los mensajes recibidos y perdidos se han distribuido alternativamente, el fabricante puede haber tomado medidas para paliar esa situación y enviar datos redundantes en cada mensaje. Aun así, esto está dentro de la capa de aplicación de cada fabricante.

A continuación, se pueden ver los gráficos de las distribuciones de mensajes perdidos (Ilustración 72) y mensajes recibidos (Ilustración 73) respectivamente.

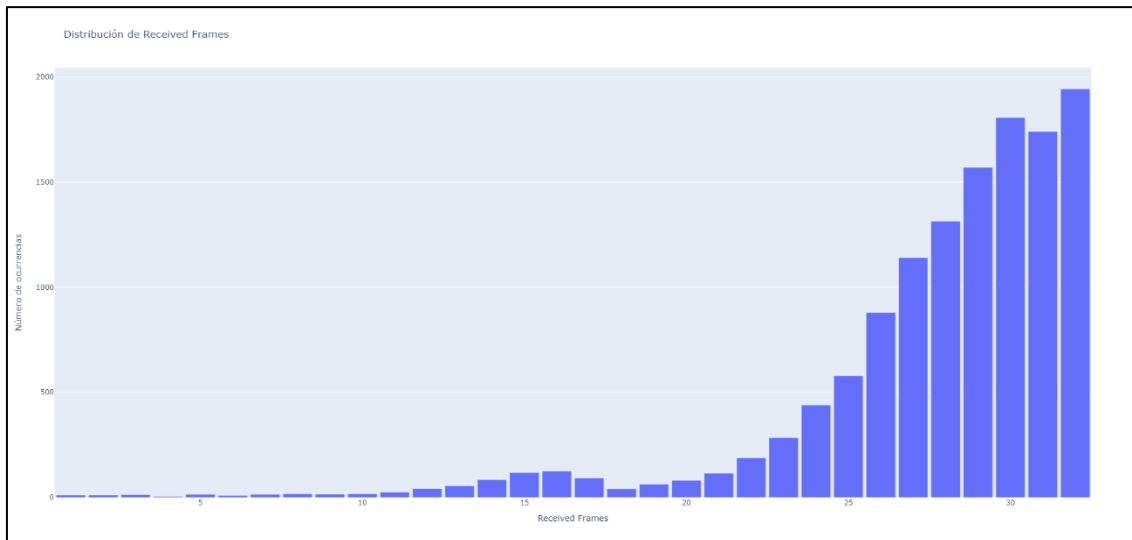


Ilustración 72: Distribución de "Lost frames" (estudio inicial)

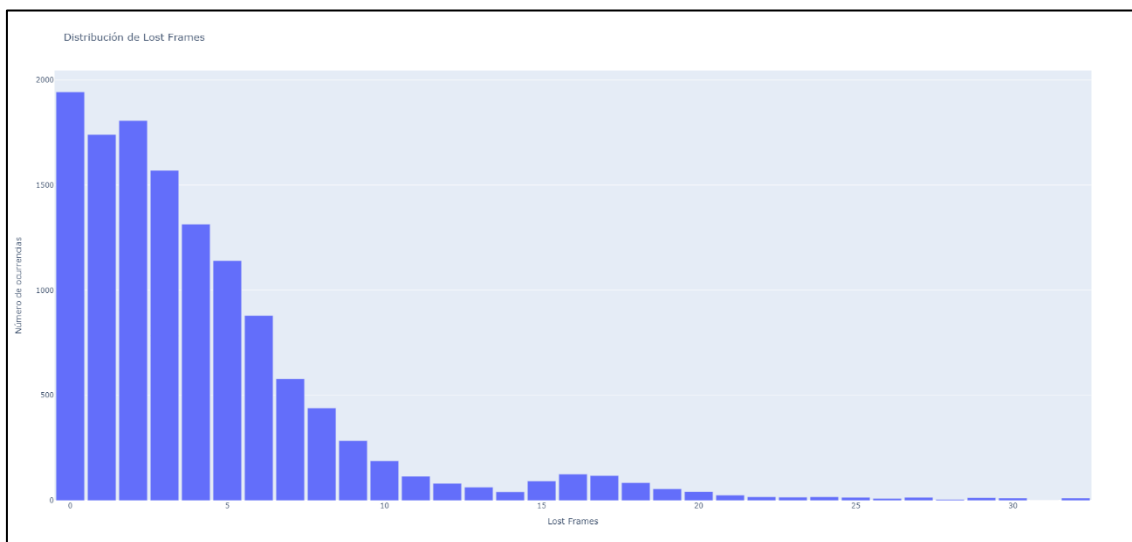


Ilustración 73: Distribución de "Received frames" (estudio inicial)

Observando los gráficos de las distribuciones de RSSI y SNR, se puede apreciar el nivel de señal en el que se están recibiendo los mensajes. Por la parte de RSSI, la mayoría de los mensajes de los devices (10.265 devices) llegan con una potencia de señal de entre los -118 dBm y -95 dBm, como se puede ver en la Ilustración 74.

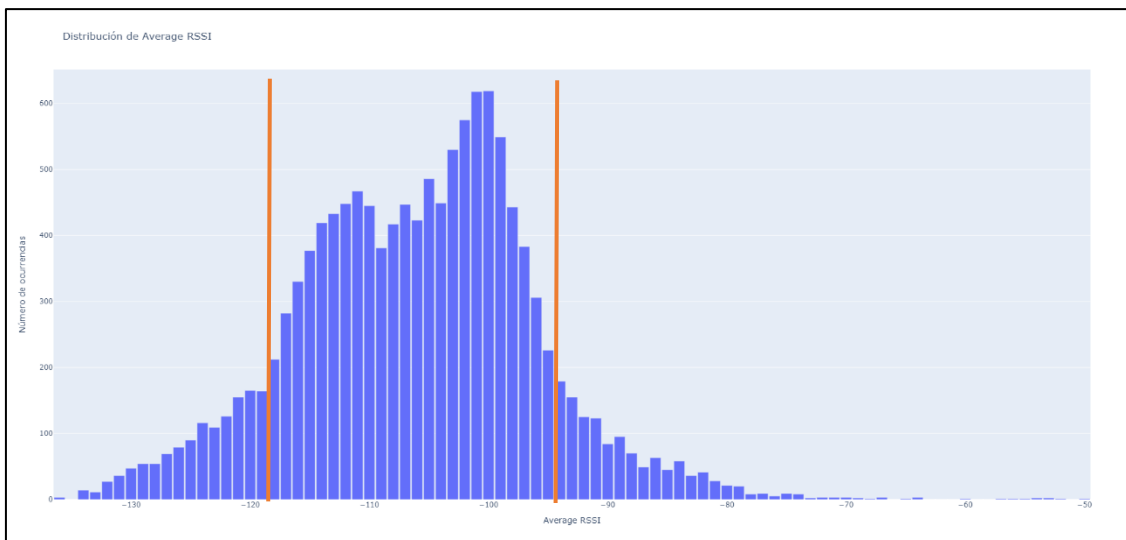


Ilustración 74: Márgenes de potencia de los mensajes (estudio inicial)

Observando la distribución del SNR, la gran mayoría de mensajes (9.285 devices) llegan con un SNR positivo, es decir, llega más señal que ruido, centrándose entre los 3 dB y 9 dB (7.329 devices), como se puede ver en la Ilustración 75. Es necesario destacar también que una parte importante de devices (927) llegan con 0 dB de SNR, siendo un numero demasiado alto comparado con los valores vecinos y la forma en la que están distribuidos los devices.

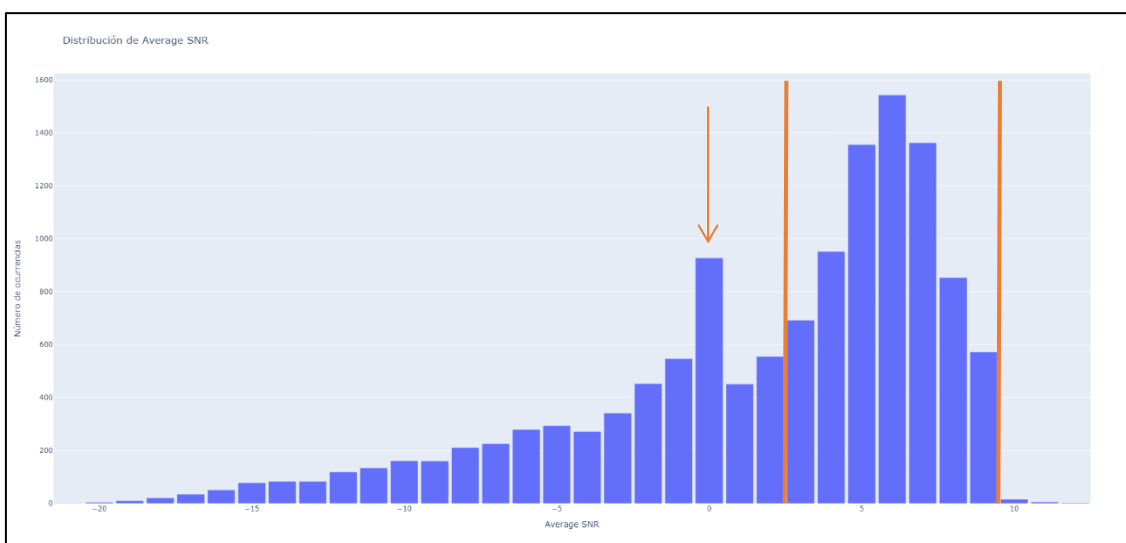


Ilustración 75: Distribución de "Average SNR" (estudio inicial)

Teniendo esto último en cuenta, si se observan los gráficos de la media de RSSI y SNR por el ultimo SF utilizado, se puede ver que cuanto mayor es el SF, menor es la potencia de señal recibida (RSSI) y aparece más ruido (SNR).

Esto se puede ver en los siguientes gráficos, donde en el primero (Ilustración 76) se refleja la media de RSSI por cada SF, y en el segundo (Ilustración 77), la media de SNR por cada SF. Además, se ve claramente que a partir de SF10, este incluido, SNR es negativo, por lo que empieza a llegar más ruido que señal a los Gateways, aunque esto no impide la obtención de los datos transmitidos, ya que el protocolo LoRaWAN está diseñado para trabajar con valores de SNR negativos.

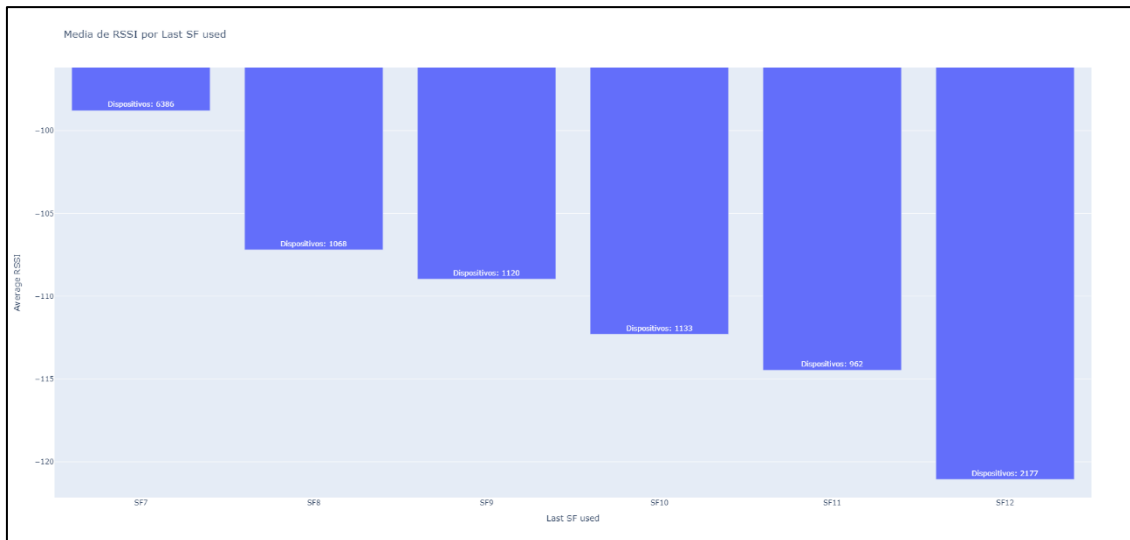


Ilustración 76: Media de RSSI por cada SF (estudio inicial)

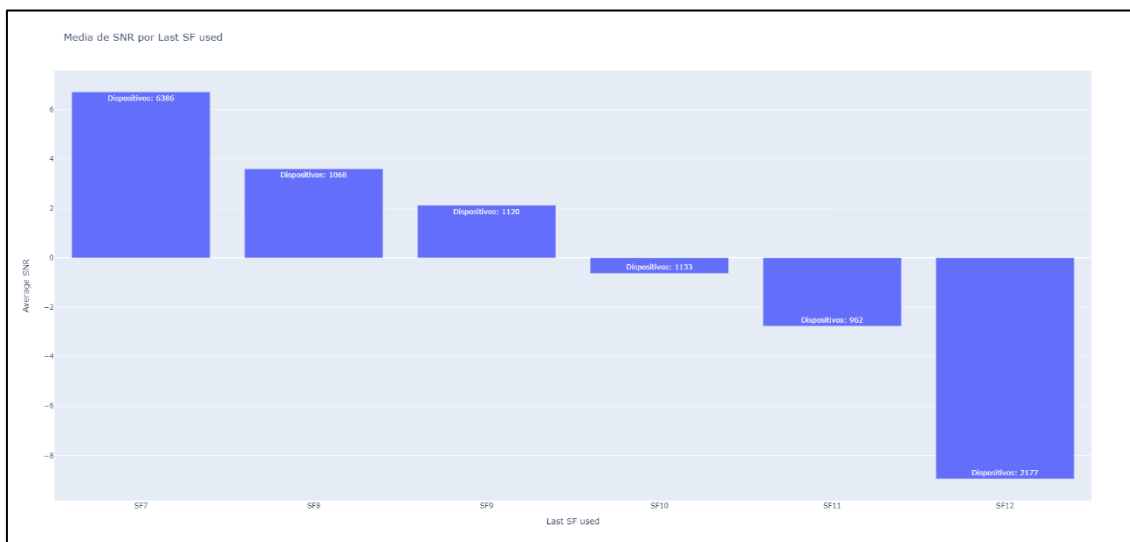


Ilustración 77: Media de SNR por cada SF (estudio inicial)

Analizando los gráficos con información referente a la distancia de los devices a su Main Gateway, se puede observar que, a medida que los devices se encuentran más lejos del Main Gateway, mayor es el porcentaje de mensajes perdidos (y de recibidos).

Analizando los gráficos, se puede observar también que, devices que estén más lejos que 1.500 metros de su Main Gateway son muy pocos, 43 exactamente (un 0,33 %), por lo que el análisis se centrara en el primer kilómetro y medio únicamente.

En la Ilustración 78 se puede ver el grafico del porcentaje de mensajes perdidos dependiendo de la distancia a la que se encuentra el Main Gateway. Además de los valores en barras representando los valores medios, también se puede observar la media acumulada de mensajes perdidos (representada con una línea continua roja). Esta línea se ve que se estabiliza sobre los 1.000 metros, que es hasta donde se encuentran la gran mayoría de devices. Se puede observar también que la media total de mensajes perdidos es de un 12,5%, y que únicamente los devices que se encuentran en un rango de 100 metros del Main Gateway tienen un porcentaje de perdida de mensajes menor al 10%.

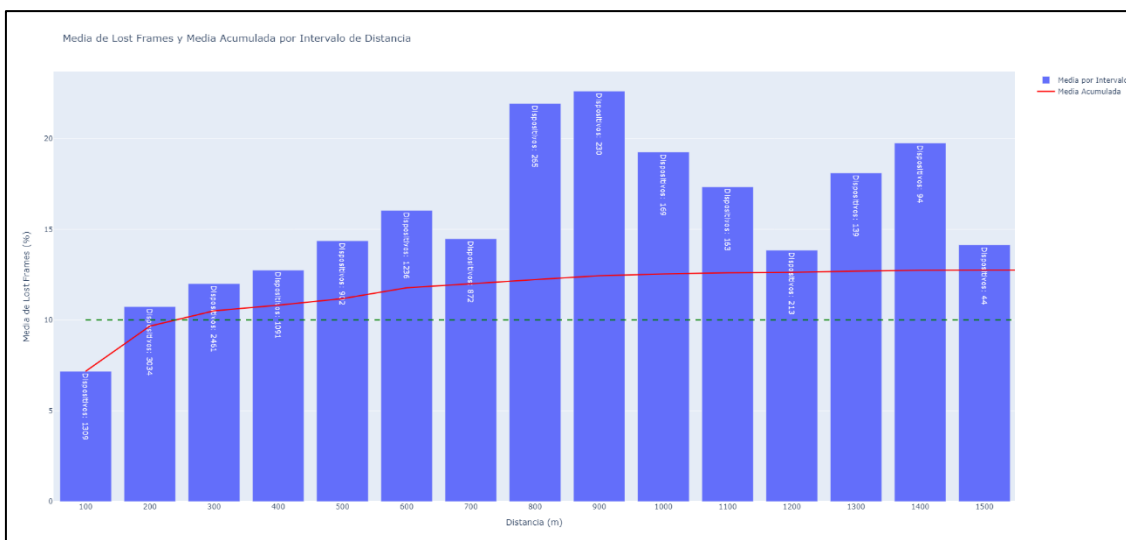


Ilustración 78: Media de mensajes perdidos por distancia al Main gateway (estudio inicial)

Analizando también los mensajes perdidos por los diferentes SF (Ilustración 79), se puede observar que únicamente los devices utilizando SF7 tienen una pérdida de paquetes menor al 10%. Además, los devices trabajando en SF9 y SF12 tienen la mayor pérdida de paquetes, los dos superiores al 20%.

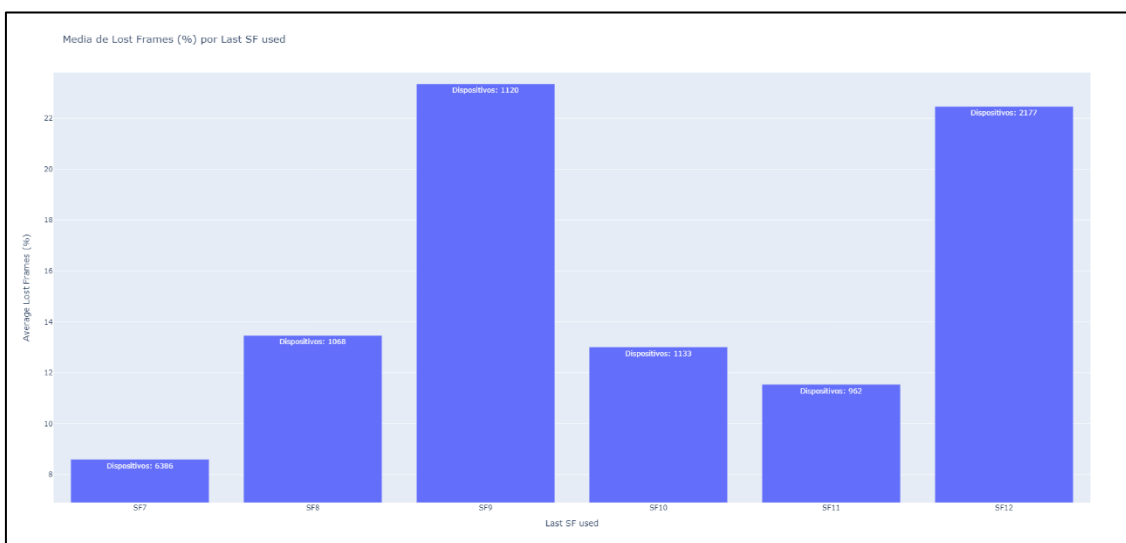


Ilustración 79: Media de mensajes perdidos por SF (estudio inicial)

5.3.6. Valoración de los resultados iniciales

Una vez analizados los gráficos que se han generado para este análisis inicial, se hará una valoración general sobre el estado de la red en el momento en el que se obtuvieron los datos. Con esta valoración se pretende encontrar parámetros de calidad de servicio (QoS) que se puedan modificar para así aumentar el rendimiento de la red.

Viendo los resultados obtenidos, se llega a la conclusión de que existe una gran cantidad de devices que solamente transmite para un único Gateway, haciendo la pérdida de paquetes más probable. Esto puede deberse a la falta de cobertura en la zona por falta de Gateways cercanos o por la ubicación del device (al ser un contador es probable que este bajo tierra) o también puede deberse a la falta de potencia en la señal transmitida.

Por otro lado, en cuanto a la pérdida de paquetes, lo ideal es que sea cero, pero eso es prácticamente inalcanzable por diferentes motivos ajenos a la propia red o de escalabilidad de esta. Para tener un objetivo realista, lo ideal es que no haya más de un 10% de pérdida de paquetes en los dispositivos analizados.

Finalmente, los niveles de potencia de señal, aun estando dentro de lo normal para el protocolo LoRaWAN, se podría intentar aumentar estos valores con el fin de que los mensajes de los dispositivos más lejanos también acaben llegando a los Gateways, llegando a su vez al *Network Server*.

En resumen, el porcentaje de mensajes perdidos supera el esperado, por lo que será necesario realizar acciones para reducir este número. Además, el número de dispositivos que solo llegan a un único Gateway es algo alarmante, ya que esto puede suponer en ciertos escenarios un aumento significativo del porcentaje de paquetes perdidos. Por último, la potencia de señal recibida, aun estando dentro de las franjas que marca el propio protocolo LoRaWAN, no son del todo buenas, por lo que se intentara mejorar estos niveles.

5.4. Iteraciones de mejora

A lo largo de este cuarto apartado, se partirá de las conclusiones obtenidas en el estudio inicial sobre el estado de la red LoRa y que cambios pueden ser aplicables para mejorar la calidad de servicio de esta.

Primero se plantearán los parámetros avanzados de la red que se deben modificar con el fin de obtener las modificaciones en la red que se habían propuesto. Tras plantear estos parámetros, se aplicarán a la red, en este caso, a Galdakao únicamente, y se esperará un tiempo prudencial para que los cambios surtan efecto.

El resto del análisis será similar al estudio inicial, ya que se obtendrán los parámetros tanto de los Gateways como de los dispositivos, se procesarán los datos obtenidos y se crearán los gráficos de este nuevo análisis. Como diferencia con el anterior estudio, en este se utilizará la interfaz gráfica funcional para llevar a cabo el proceso de la obtención de los datos, el tratamiento y la generación de los gráficos. Además, no solo se generarán los gráficos de este análisis, sino que también se crearán gráficos comparativos entre este y el anterior.

Finalmente, se hará una interpretación del estado de la red a través de los gráficos generados y se valorará el nuevo estado tras haber aplicado los cambios pertinentes.

En resumen, en este bloque se explicarán en profundidad los siguientes subapartados:

- Replanteamiento de nuevos parámetros QoS
- Obtención de los datos de los Gateways
- Obtención de los datos de los dispositivos
- Procesado de datos
- Generación de gráficos
- Interpretación del estado de la red
- Valoración de los resultados tras los cambios

5.4.1. Replanteamiento de nuevos parámetros QoS

Con el fin de cambiar la calidad de la red LoRa, se ha planteado cambiar diferentes parámetros QoS. El *Network Server* de Orbiwise permite al administrador de la red establecer perfiles de

calidad de servicio, pudiendo estos asignarse a usuarios enteros, grupos o incluso devices individuales.

Los parámetros que se pueden modificar aparecen en la siguiente tabla (Tabla 9), además de una pequeña descripción de lo que significa cada parámetro y el valor que asigna Orbiwise en el perfil de QoS que viene por defecto en el propio *Network Server*.

Nombre	Descripción	Valor
LoRaWAN region	Región de trabajo de LoRaWAN	EU868
Min data-rate	Data rate mínimo posible para uplink	0
Max data-rate	Data rate máximo posible para uplink	5
Min TX power	Potencia de transmisión mínima posible [dBm] en uplink	2
Max TX power	Potencia de transmisión máxima posible [dBm] en uplink	16
SNR margin in dB	Margen sobre el SNR promedio requerido para la demodulación	12
RSSI margin in dB	Margen sobre el RSSI promedio requerido para la demodulación	12
Target redundancy	Número de retransmisiones uplink, espaciales + temporales	0
Min unconfirmed UL repetitions	Mínimo de veces que un mensaje de uplink tiene que ser recibido en el Network Server	1
Max unconfirmed UL repetitions	Máximo de veces que un mensaje de uplink tiene que ser recibido en el Network Server	5
Averaging window length	Longitud de la ventana de promediado	15
Additional SNR margin on join	Margen adicional en el join cuando las estadísticas SNR no son buenas	10
Max SF steps for ADR on join	Número máximo de cambios de SF en el join	5
Max SF steps for ADR	Número máximo de cambios de SF en modo regular	2
Min number of gateways to receive device	Número mínimo de Gateways que tiene que recibir las tramas	1
Delay between two TxPower control steps	Retraso (mensajes) entre dos mensajes de control de potencia de transmisión	6
Gateway aggregation time [ms]	Tiempo que el Network Server esperará para recopilar los mensajes de uplink	0

Tabla 9: Parámetros QoS configurables en el Network Server

Con las valoraciones del estudio inicial y viendo que parámetros QoS son modificables en el *Network Server*, se pueden seleccionar varios parámetros de estos para intentar modificar la calidad de la red LoRa e intentar obtener los resultados esperados.

De todos los parámetros, se han seleccionado 4 de ellos para modificarlos, y así ver los cambios que estos producen en la red.

- **Min Tx power:** aumentando este parámetro se pretende obligar a los devices a transmitir a mayor potencia, dando mejor señal a los Gateways a cambio de mayor consumo de batería.
- **RSSI margin in dB:** aumentando este parámetro, se obliga a los devices a transmitir con mayor potencia también, pero esta vez porque el Gateway necesita un margen de RSSI más alto para recibir y demodular el mensaje.
- **Target redundancy:** aumentando este otro parámetro, se obliga al device a tener que transmitir el mensaje a un mínimo de dos Gateways, pero en el caso de que no llegue a dos Gateways diferentes, tendrá que retransmitir el mensaje de nuevo para el mismo Gateway.
- **Min number of gateways to receive device:** aumentando este parámetro, se obliga al device a llegar a dos Gateways o más, únicamente siendo válida esta opción y no la de retransmisión de mensaje al mismo Gateway.

Habiendo definido los parámetros QoS que se van a modificar y su porque, en la Tabla 10 se puede observar el valor inicial de este parámetro y su valor después de haberle aplicado el cambio.

Parámetro	Valor original	Nuevo valor
Min TX power	2	16
RSSI margin in dB	12	18
Target redundancy	0	1
Min number of gateways to receive device	1	2

Tabla 10: Nuevos valores de los parámetros QoS modificados (iteración)

Con estos nuevos valores se pretende obligar a los devices a transmitir a una mayor potencia y a un número mayor de Gateways, con el fin de perder el mínimo número de mensajes.

5.4.2. Obtención de los datos de los Gateways

Para la obtención de los datos de los Gateways, es necesario esperar cierto tiempo para que los nuevos parámetros QoS se apliquen correctamente. Estos parámetros son exclusivos de los devices, pero, los parámetros que se obtienen de los Gateways son información sobre los mensajes que han recibido de los devices, por lo que es necesario esperar hasta que los devices transmitan con normalidad con los nuevos parámetros cambiados.

El tiempo que se debe esperar para obtener los datos es algo relativo, ya que depende cuanto tiempo tarden los devices en aplicar los cambios. Como para el análisis se están utilizando datos de 7 días, lo mínimo que se debe esperar es ese tiempo.

Pero, los cambios aplicados a los devices no son instantáneos, ya que, para que el *Network Server* le comunique a un device cierta información, este debe hacerlo durante las ventanas de recepción de datos de los devices, tal y como indica el protocolo LoRaWAN.

Los devices cuentan con dos ventanas de recepción que se abren después de enviar un mensaje, normalmente tras 1 y 5 segundos cada una, en los devices de clase A [18]. Los contadores de agua utilizados para este análisis únicamente envían entre 4 y 5 mensajes al día, por lo que la comunicación entre device y *Network Server* puede llegar a ser algo lenta.

Para evitar que parte de los contadores que se van a analizar no hayan aplicado los cambios, se ha optado por dejar 2 semanas de margen para que los devices puedan hacer los cambios que necesiten.

Por ello, sumando las 2 semanas para aplicar los cambios y los 7 días de recolección de datos, se deben esperar, al menos, 3 semanas para realizar el nuevo análisis.

La obtención de los datos de los Gateways en si es similar a la realizada en el estudio inicial, pero esta vez se utilizará la interfaz gráfica funcional para realizar estas tareas.

Resumiendo, estos son los pasos que se seguirán para la obtención de los datos de los Gateways:

- Seleccionar un rango de tiempo para el análisis.
- Seleccionar los Gateways que se quieren analizar.
- Obtención de información estática de cada Gateway (API Orbiwise).
- Obtención de parámetros relacionados con la calidad de los mensajes de cada Gateway (web scraping).

Una vez realizado estos pasos, se habrán conseguido los archivos JSON necesarios para su posterior tratamiento.

5.4.3. Obtención de los datos de los devices

Para la obtención de los datos de los Gateways, también es necesario esperar las 3 semanas que se habían definido, por las mismas razones que para la obtención de los parámetros de los Gateways, explicadas en el apartado anterior.

En la ilustración 80 se puede observar un mensaje de configuración que envía el *Network Server* a un device, con el fin de que este cambie el SF que está utilizando.

#	Timestamp	Event	Gateway	DevEUI	DevAddr	ChID	FCnt	FHnt	MacCmd	RSSI	SNR	Freq	SF
12	13:23:15.526 26/02/2024	DL	000800FF48C877 Galdakao Andra Mari S/N:21713706	0007813700D13E3A	00A2F16	UnConfDev	191		major=0, FCtrl:05 [] Fops: 0300ffff01, Len:17, toa:659.456s, v:1.0.2B rg:EU868 MacCmds: LinkADDRReq(DataRate:0 [SF12], TXPower:0 [MaxEIRP], Mask:ffff, ChMaskCtrl:0, NbRep:1), DL Status: [1] Message was transmitted on slot 1 Encrypted: 60 16 EF 0A 10 05 BE 00 03 00 FF FF 01 CC 97 38 2D			867.1	SF10B/12S 4/5

Ilustración 80: Mensaje de configuración del Network Server a un device

En este caso, el device estaba emitiendo el SF10, y el mensaje de configuración le obliga a cambiar a SF12, tal y como se observa en el comando MAC que ha sido enviado (ilustración 81).

major=0, FCtrl:05 [] Fops: 0300ffff01, Len:17, toa:659.456s, v:1.0.2B rg:EU868 MacCmds: LinkADDRReq(DataRate:0 [SF12], TXPower:0 [MaxEIRP], Mask:ffff, ChMaskCtrl:0, NbRep:1), DL Status: [1] Message was transmitted on slot 1 Encrypted: 60 16 EF 0A 10 05 BE 00 03 00 FF FF 01 CC 97 38 2D

Ilustración 81: Comando MAC enviado en el mensaje de configuración

Una vez esperado el tiempo predefinido, se utiliza la interfaz de usuario funcional para realizar el nuevo análisis, en vez de tener que estar ejecutando diferentes scripts uno a uno en un orden concreto.

Utilizando la GUI funcional, se deberán seguir los mismos pasos que con los scripts, pero de una forma mucho más guiada y con probabilidades mucho menores de que ocurra algún error durante el proceso. Los pasos a seguir en cuestión serían los siguientes:

- Seleccionar usuario para el análisis.
- Elegir un grupo (municipio) dentro de ese usuario.
- Obtención de la información estática de todos los devices de ese grupo.

- Seleccionar un rango de fechas para el análisis.
- Obtención de los parámetros relacionados con la calidad de los mensajes enviados por los devices durante el rango de tiempo seleccionado, para cada uno de los devices.

Siguiendo estos pasos se habrán conseguido los archivos JSON necesarios para continuar con el análisis.

5.4.4. *Procesado de datos*

Al igual que en el estudio inicial, tanto los datos obtenidos de los Gateways como de los devices tienen que ser procesados al formato JSON predefinido, para posteriormente poder generar los gráficos que se desean.

Con la interfaz gráfica funcional, este proceso se realiza automáticamente una vez finalizada la obtención de los parámetros de los Gateways y de los devices, pero antes de que se generen los gráficos, como es lógico.

El proceso de la generación de los archivos JSON procesados queda relegado a un segundo plano, dejando al usuario una pantalla de carga mientras se ejecuta este proceso.

El formato de los datos procesados es exactamente igual a los que se han obtenido en el estudio inicial, tanto para los devices como para los Gateways.

5.4.5. *Generación de gráficos*

La generación de gráficos durante este análisis también se realiza automáticamente, y en segundo plano, gracias a la interfaz de usuario funcional desarrollada.

Los gráficos generados son los mismos que en el estudio inicial, pero con los datos de entrada de este nuevo análisis. Al igual que los gráficos generados para el estudio inicial, estos también se pueden separar en 5 categorías diferentes:

- Mapas
- Ordenado por Gateway
- Distribuciones
- Ordenado por distancias al Main Gateway
- Ordenado por el último SF utilizado

Pero, además de todos estos gráficos, como este análisis trata de comparar los resultados con el estudio inicial, también se han creado gráficos comparativos entre los datos del primer análisis y los de este.

Para realizar esto, también se ha utilizado la interfaz gráfica funcional, teniendo únicamente que seleccionar los datos obtenidos del primer análisis y los de este, relegando a un segundo plano la creación de estos gráficos.

Los gráficos comparativos creados son exactamente los mismos que los generados para un análisis individual, a excepción del mapa, ya que no es necesario comparar las posiciones de los Gateways y de los devices, ya que todos son elementos estáticos de la red.

Entre estos gráficos comparativos, se puede encontrar el de la Ilustración 82, donde se muestra la media de RSSI (en dBm) de todos los paquetes recibidos por cada Gateway, diferenciando el primer análisis (en azul) con el segundo (en rojo).

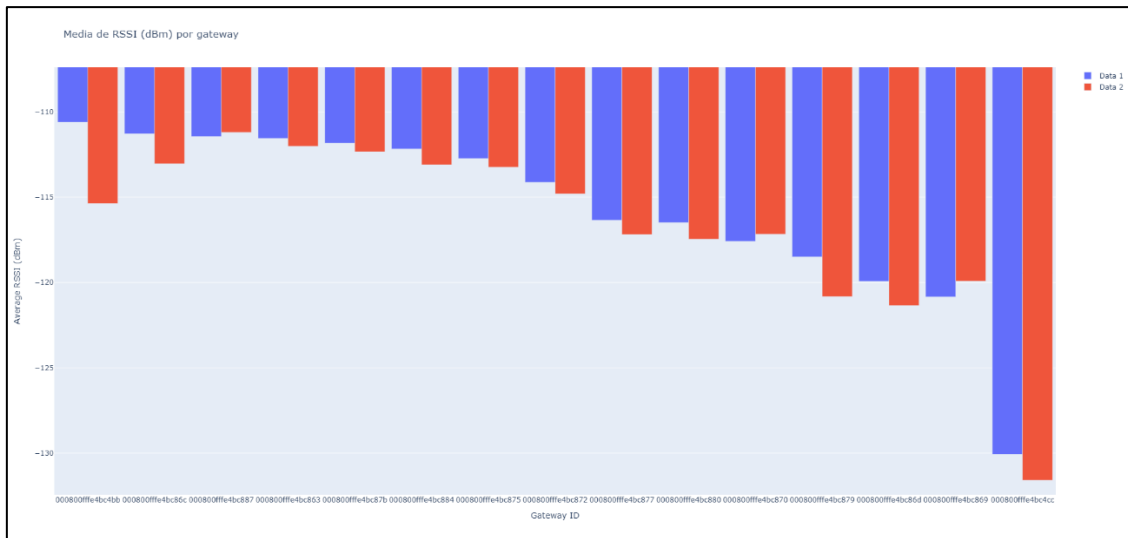


Ilustración 82: Comparativa de la media de RSSI de los mensajes recibidos en cada Gateway

Al igual que este gráfico, el resto de ellos que comparan el estudio inicial con este análisis realizado, tendrán el mismo formato, donde los datos del primer análisis se mostrarán en azul y los de la iteración de mejora en rojo.

5.4.6. Interpretación del estado de la red

Analizando más en profundidad los gráficos que se han obtenido, se pueden obtener varias conclusiones de cómo ha cambiado el estado de la red en Galdakao. Para ver mejor los resultados, a lo largo de este apartado únicamente se utilizarán los gráficos comparativos entre el estudio inicial y el análisis realizado como iteración de mejora, y se mostrarán los que contengan información más significativa para analizar el cambio realizado en la red.

Empezando por el gráfico que muestra el número de mensajes recibidos por cada Gateway (Ilustración 83), se puede observar un aumento considerable en el número de mensajes en esta iteración comparado con el estado inicial de la red.

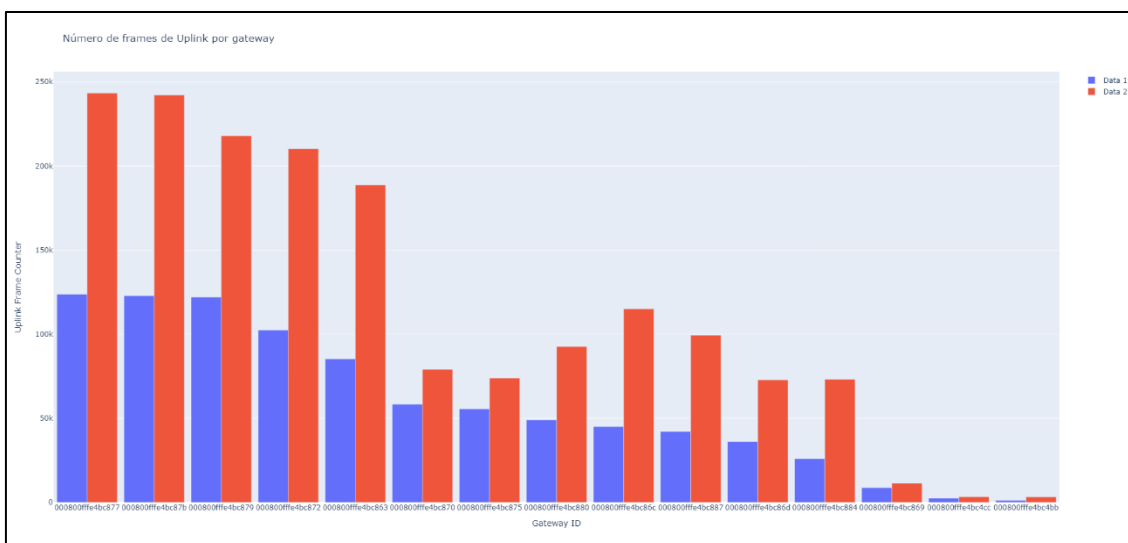


Ilustración 83: Comparación del número de mensajes recibidos por cada Gateway

Por otro lado, si se observa el gráfico que muestra a cuantos Gateways llegan de media los mensajes de los devices (Ilustración 84), se puede observar un desplazamiento hacia la derecha de las barras, esto es, que ha aumentado la media de a cuantos Gateways llegan los mensajes enviados por los devices.

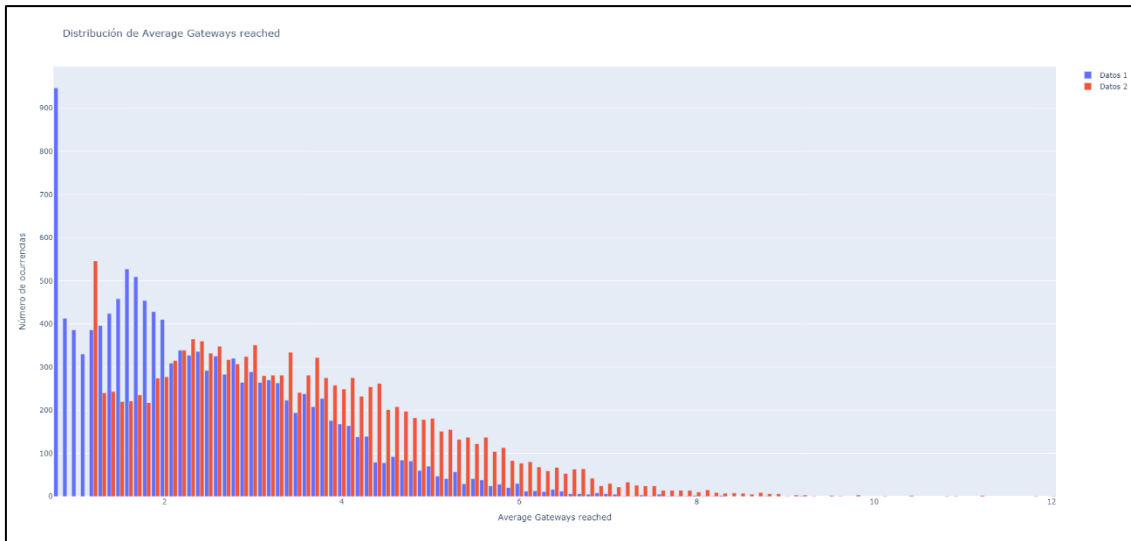


Ilustración 84: Comparación del número de Gateways al que llegan los mensajes

Si se observa el gráfico de la distribución de los devices dependiendo del SF que están utilizando (Ilustración 85), se observa claramente que una gran cantidad de devices (unos 2.500) han pasado de utilizar SF7 a trabajar con SF12.

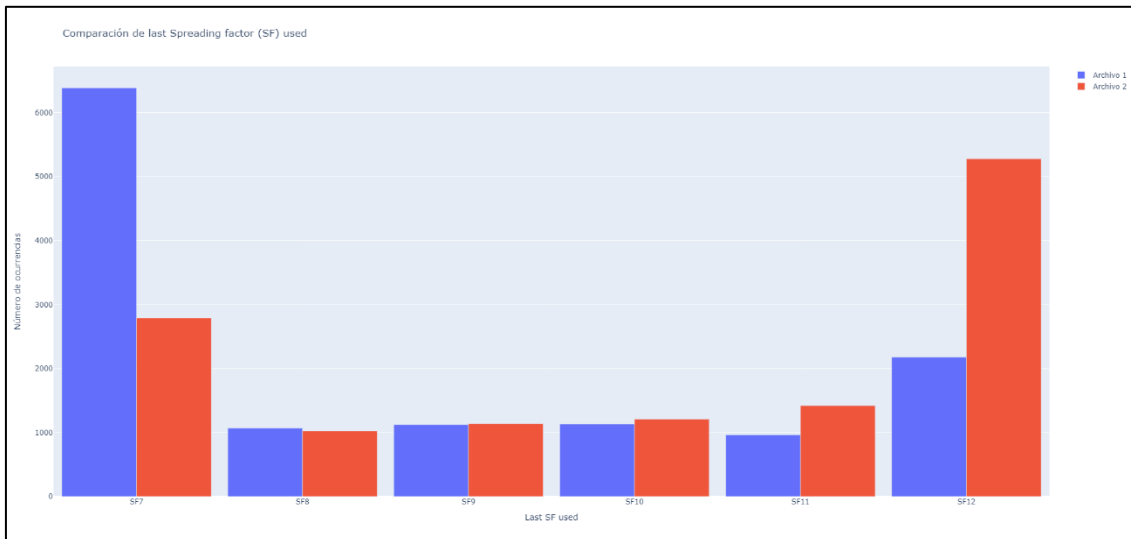


Ilustración 85: Comparación del último SF utilizado por los devices

Teniendo esto en cuenta, como el tiempo en el aire (ToA) de los mensajes aumenta a medida que aumenta el SF, en el gráfico que muestra la media de ToA dependiendo a la distancia a la que se encuentren del Main Gateway (Ilustración 86), se puede observar claramente que la distancia ya no es un factor determinante en el tiempo que tarda el mensaje en llegar al Gateway, como pasaba en el estudio inicial, sino que ahora es mucho más constante independientemente de la distancia.

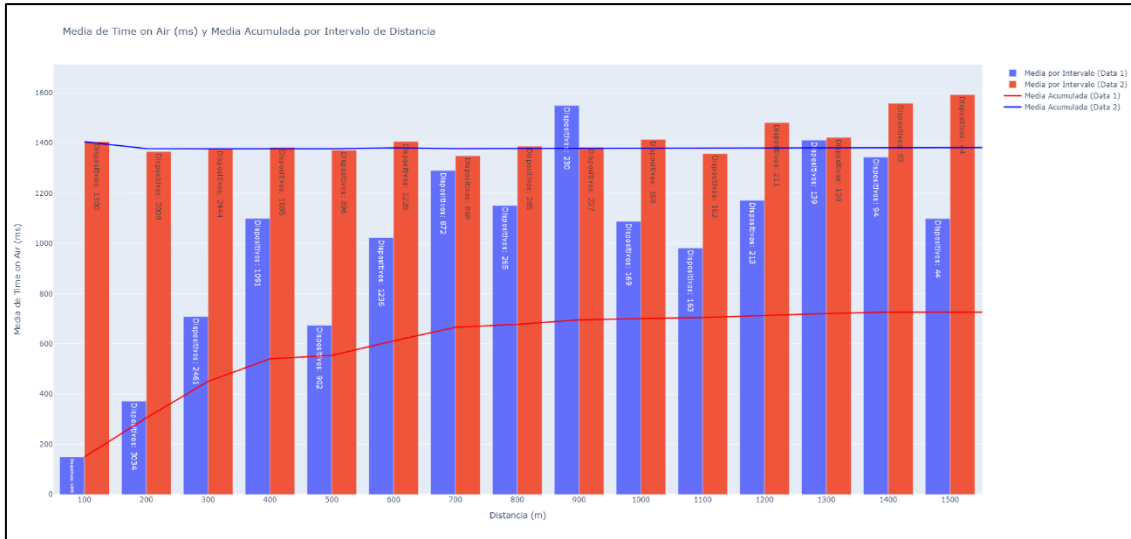


Ilustración 86: Comparación del ToA relativo a la distancia al Main Gateway

Si se analiza las calidades de las señales recibidas, RSSI y SNR, se puede ver que ha habido un aumento considerable en estos dos parámetros. Mirando el grafico de la media de RSSI por cada SF diferente (Ilustración 87), se puede observar un aumento de entre 7 y 10 dBm para cada SF. Y si se observa el grafico de SNR en función del SF (Ilustración 88), se puede ver que ahora el único SF que tiene un SNR negativo es el SF12, en el resto se recibe más señal que ruido.

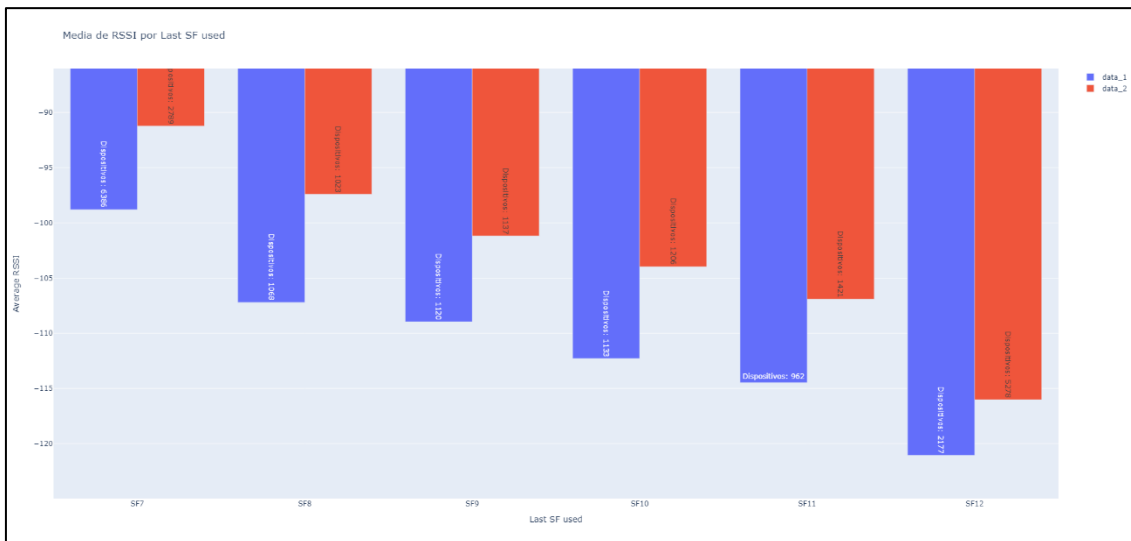


Ilustración 87: Comparación de RSSI en función de SF

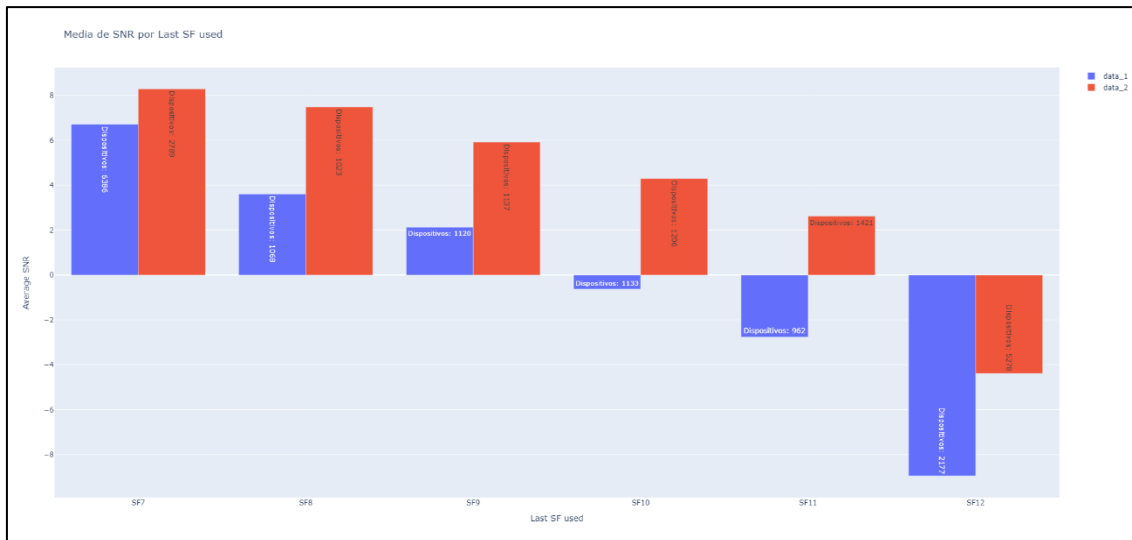


Ilustración 88: Comparación de SNR en función de SF

Observando las distribuciones de estos dos parámetros, se puede ver que, en las medidas que había un mayor número de devices, una parte de esos se han desplazado hacia la derecha, aumentando el nivel de RSSI y de SNR. Esto se puede ver en la Ilustración 89 e Ilustración 90 respectivamente.

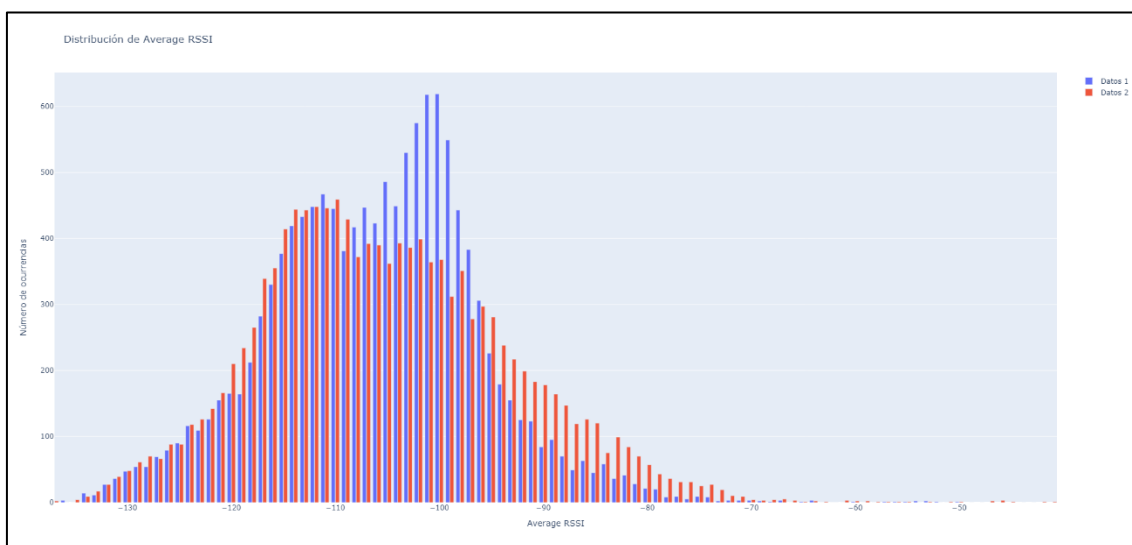


Ilustración 89: Comparación de la distribución de RSSI

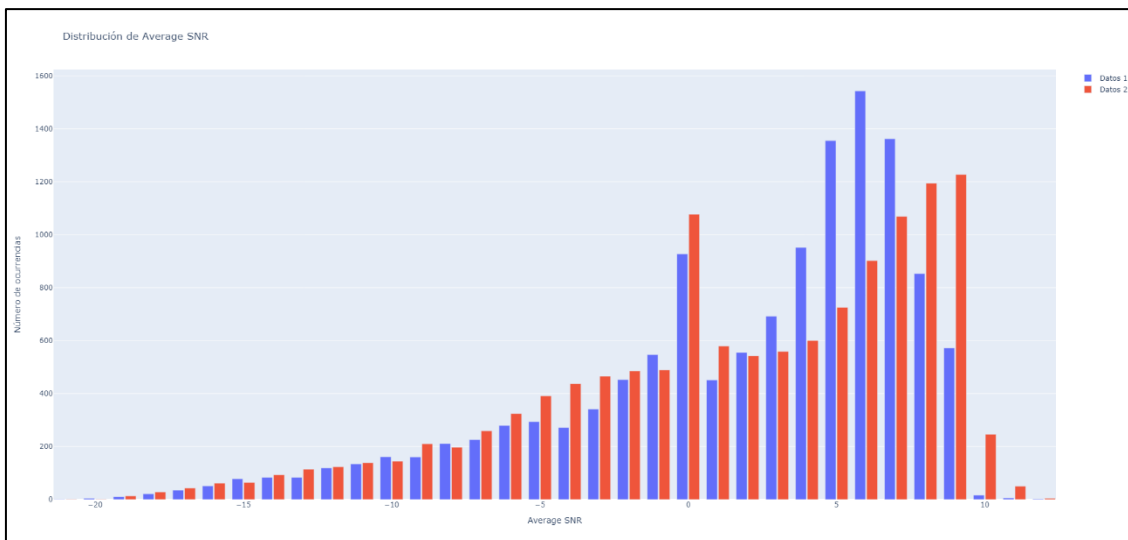


Ilustración 90: Comparación de la distribución de SNR

Finalmente, analizando los gráficos que muestran las tasas de mensajes recibidos y perdidos, se puede observar que el número de devices que han recibido una mayor cantidad de mensajes a aumentado, como se puede ver en la Ilustración 91. Además, se puede ver que, en este análisis, algunos devices (424) han llegado a transmitir 33 mensajes, en vez de los 32 como máximo que se habían transmitido en los 7 días del estudio inicial.

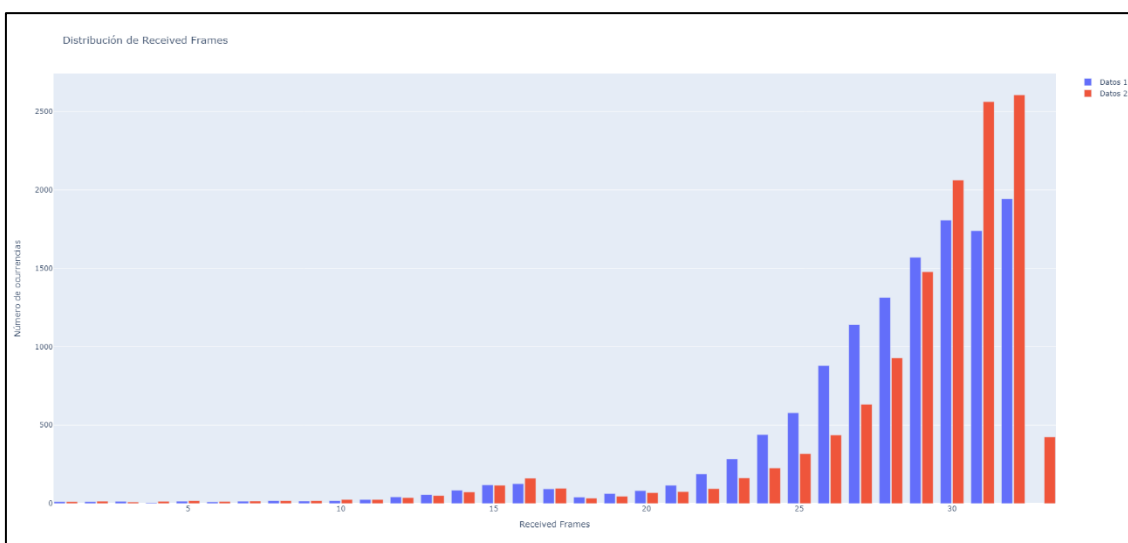


Ilustración 91: Comparación del número de mensajes recibidos

Por otro lado, si se analizan los mensajes perdidos dependiendo de la distancia (Ilustración 92), se puede ver que la distancia al Main Gateway ya no es un factor tan determinante, y que el porcentaje de pérdidas se mantiene más o menos constante independientemente de la distancia a la que se encuentre el device del Gateway. Además, la media acumulada de los mensajes perdidos no supera el 10%, al contrario de lo que pasaba en el estudio inicial.

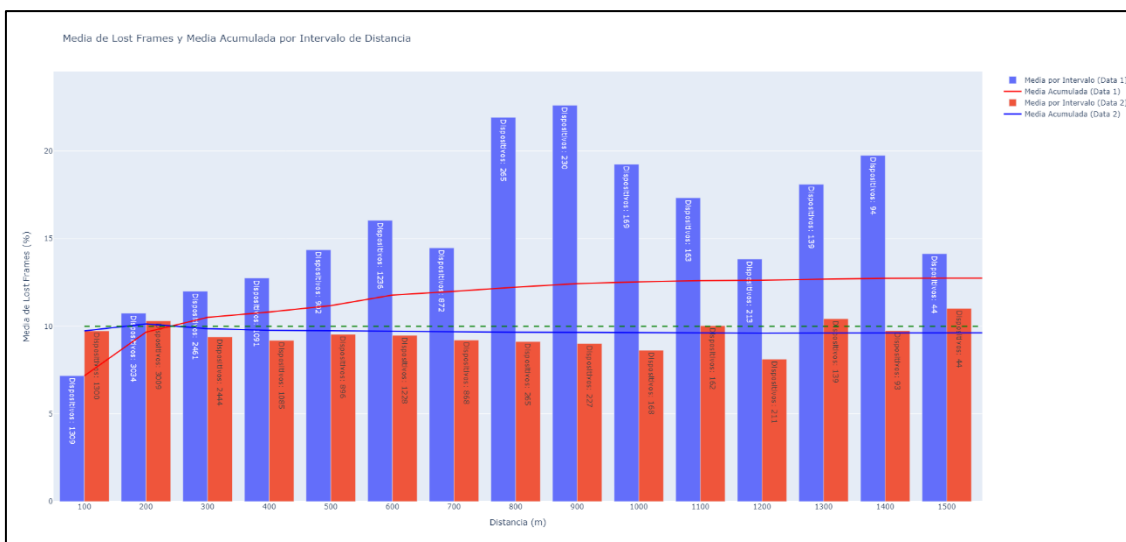


Ilustración 92: Comparación de porcentaje de mensajes perdidos en relación a la distancia al Main Gateway

Y viendo la información del porcentaje de mensajes perdidos dependiendo del SF que está utilizando los devices (Ilustración 93), se ve claramente como en cada uno de los SF se ha reducido el porcentaje de perdidas, excepto en SF7, que se ha mantenido igual.

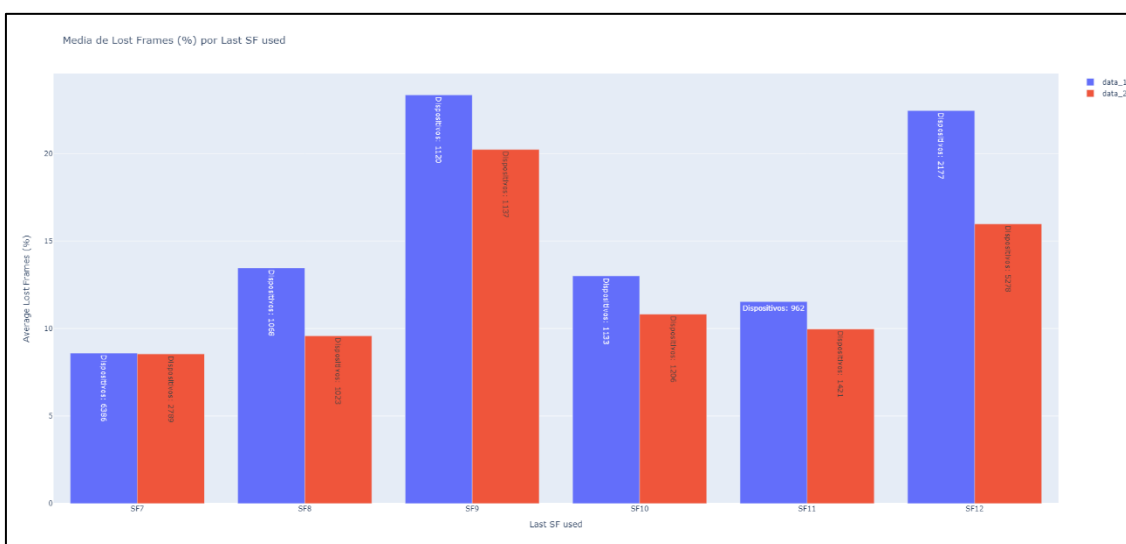


Ilustración 93: Comparación de porcentaje de perdidas en relación al SF

5.4.7. Valoración de los resultados tras los cambios

Después de haber analizado los gráficos de esta iteración y haberlos comparado con los obtenidos del estudio inicial, se pueden obtener varias conclusiones respecto al nuevo estado de la red y como han afectado estos cambios a esta.

Lo primero que se ha podido observar es que el número de tramas totales ha aumentado considerablemente, haciéndola aproximadamente el doble. Esto se debe a que los devices se habían configurado para que mínimo llegasen a dos Gateways, y en caso de que no fuese así, que envíen el mensaje de nuevo para tener una copia en el mismo Gateway.

Esto ha provocado que el porcentaje de tiempo en el que ha habido tramas en el aire haya aumentado, aumentando a su vez las probabilidades de colisión entre estas, lo que implicaría la pérdida o corrupción de la trama.

Además, como se ha obligado a que los devices lleguen a una mayor distancia con sus tramas, para llegar a más Gateways, esto ha provocado que muchos de los devices que estaban usando SF7 para transmitir datos, haya pasado a utilizar SF12, para cubrir esta nueva necesidad.

El aumento de SF implica un aumento de distancia a la que puede llegar la trama, a cambio de un mayor ToA y un mayor uso de la radio, lo que implica un mayor gasto de batería. En la Ilustración 94 se puede ver la diferencia de tiempo de ToA de cada SF.

Data-Rate	SF	BW	Maximum MACPayload size, bytes	Maximum FRMPayload size, bytes	Shortest uplink frame ToA, ms	Longest uplink frame ToA, ms
DR0	12	125	59	51	1155	2793
DR1	11	125	59	51	578	1560
DR2	10	125	59	51	289	698
DR3	9	125	123	115	165	677
DR4	8	125	230	222	82	656
DR5	7	125	230	222	46	369

Ilustración 94: Relación entre el SF y ToA

Pero, por mucho que se haya visto un aumento considerable tanto en el número de tramas en la red como de ToA de estas, con el aumento de la probabilidad de colisión entre tramas que esto supone, no se ha visto que haya habido un número mayor de paquetes perdidos. Por el contrario, el porcentaje de mensajes recibidos en comparación a la situación inicial de la red ha aumentado, quedando por debajo de un 10% de mensajes perdidos.

Esto implica que la red no ha llegado a saturarse y ha aguantado el aumento de tramas y de ToA sin mayor problema, habiendo 13.000 devices y 15 Gateways (16 totales, pero uno de ellos está apagado), con una ratio de 870 devices por Gateway aproximadamente.

Este es un dato a tener en cuenta para futuros análisis en diferentes municipios, ya que, en caso de que la ratio de devices por Gateway sea mayor, pueden empezar a generarse colisiones y pérdida de paquetes, empeorando considerablemente la calidad de la red.

También es destacar el aumento de la potencia de la señal recibida, la cual se puede apreciar tanto en el RSSI como en el SNR. Gracias al diseño del protocolo LoRaWAN, este permite recibir con un nivel de SNR negativo y aun así ser capaz de demodular la señal y obtener los datos que venían en ella (Ilustración 95).

SpreadingFactor (RegModulationCfg)	Spreading Factor (Chips / symbol)	LoRa Demodulator SNR
6	64	-5 dB
7	128	-7.5 dB
8	256	-10 dB
9	512	-12.5 dB
10	1024	-15 dB
11	2048	-17.5 dB
12	4096	-20 dB

Ilustración 95: SNR mínimo para demodular dependiendo el SF [19]

Por mucho que durante el estudio inicial se viene que los valores de SNR estaban por encima de los mínimos, con los nuevos parámetros de QoS aplicado, se ha conseguido aumentar la media de SNR recibido en hasta 5 dB en el mejor de los casos, significando una gran mejora en cuanto a la relación señal ruido recibida.

Resumiendo, una gran cantidad de devices ha pasado de utilizar SF7 a SF12 para la transmisión de datos, lo que esto supone un mayor tiempo en el aire de los mensajes que estos envían. Además, los devices que aun estando en SF12 no son capaces de llegar a al menos 2 Gateways diferentes han sido forzados a reenviar el mismo mensaje para tener cierta redundancia. Todo esto ha traído consigo un aumento considerable de numero de tramas, a lo que la red ha sabido responder sin ningún problema.

El aumento de numero de tramas y de un mayor tiempo de emisión de los mensajes puede traer que la batería de los devices se agote más deprisa de lo normal, por lo que es un factor a tener en cuenta a la larga, ya que los devices en este caso son contadores de agua, los cuales no están conectados a la red eléctrica y deben de tener al menos 12 años de vida útil, sin ningún tipo de mantenimiento.

Pero, aunque el número de tramas haya aumentado, la red no se ha visto congestionada y ha sido capaz de manejar el aumento de tráfico. Esto ha resultado en una mejora en el porcentaje de mensajes perdidos, reduciendo la media a menos de un 10%. Todo esto teniendo en cuenta que la ratio de devices por Gateway es de unos 870 aproximadamente.

5.5. Configuración final

En este último apartado se volverán a cambiar diferentes parámetros avanzados de la red en función de los resultados obtenidos y valoraciones realizadas de los anteriores análisis, intentando optimizar aún más la calidad de la red.

Para ello, se seguirá la misma estructura del anterior análisis. Primero, se redefinirán diferentes parámetros QoS que se consideren importantes para mejorar la calidad de la red, partiendo siempre de la base de los anteriores estudios.

Una vez cambiados estos parámetros, se tendrá que volver a esperar un tiempo predefinido para poder obtener datos tanto de los Gateways como de los devices sobre el nuevo estado de la red.

A estos nuevos datos se les volverá a aplicar un tratamiento para guardarlos en el formato correcto, y una vez realizado esto, se generarán nuevamente los gráficos de este análisis. Además, como ya se han realizado 3 estudios diferentes, se generarán también los gráficos comparativos entre los 3 análisis, para tener una visión más global de los cambios que ha sufrido la red.

Todo esto se realizará utilizando ya la aplicación final ya creada, para una mayor comodidad y agilidad a la hora de ejecutar todos los pasos requeridos para la realización del análisis.

Finalmente, se hará una interpretación de los gráficos obtenidos, utilizando los comparativos entre los diferentes análisis y se valorarán los resultados obtenidos para finalmente obtener unas conclusiones sobre el nuevo estado de la red y como ha ido evolucionando gracias a los cambios aplicados.

En resumen, en este bloque se explicarán en profundidad los siguientes subapartados:

- Replanteamiento de nuevos parámetros QoS
- Obtención de los datos de los Gateways
- Obtención de los datos de los devices
- Procesado de datos
- Generación de gráficos
- Interpretación del estado de la red
- Valoración de los resultados y conclusiones finales

5.5.1. Replanteamiento de nuevos parámetros QoS

Con la intención de optimizar aún más la red provocando el mínimo de pérdidas de paquetes mientras que el resto de parámetros importantes se maximizan, a través de lo aprendido durante la iteración de mejora, se plantean hacer nuevos cambios en los parámetros QoS de la red LoRa.

Durante el anterior estudio, se han cambiado 4 parámetros de calidad de servicio, los cuales han forzado en los devices dos cambios mayores. Primero, la potencia de transmisión ha aumentado considerablemente, y segundo, un gran número de devices han pasado de trabajar con SF7 a trabajar usando SF12.

El cambio de estos parámetros ha tenido como consecuencia el aumento de SNR y RSSI de la señal recibida, al igual que una disminución de porcentaje de paquetes perdidos. Pero, esto ha hecho que muchos devices pasen a usar SF12, lo cual supone un mayor tiempo en el aire y una mayor probabilidad de colisión. Además, a esto implica también un mayor uso de la radio, por lo que el consumo de batería también se verá afectado.

Para intentar paliar los efectos negativos y aumentar los positivos, se plantea revertir 3 de los 4 cambios realizados en los parámetros QoS y dejar solo uno de ellos, para poder ver el comportamiento de la red. El parámetro en cuestión que no se va a modificar respecto al estudio anterior es la potencia de transmisión mínima, que se mantendrá a 16 dBm.

Los otros 3 parámetros volverán a configurarse con los valores por defecto del Network Server, los cuales se pueden ver en la Tabla 11.

Parámetro	Valor original	Nuevo valor
RSSI margin in dB	18	12
Target redundancy	1	0
Min number of gateways to receive device	2	1

Tabla 11: Nuevos valores de los parámetros QoS modificados (final)

Con estos nuevos cambios se volverá a medir la calidad de la red para analizar el efecto que han tenido los cambios aplicados.

5.5.2. Obtención de los datos de los Gateways

Al igual que en el estudio anterior, para la obtención de los datos de los Gateways es necesario que los parámetros modificados se apliquen a los devices correctamente, siendo necesario un margen de tiempo para ello.

Con el fin de ser consistente con el procedimiento llevado a cabo en el anterior estudio, para este último también se esperarán 2 semanas hasta que las nuevas configuraciones de los devices hayan sido aplicadas.

Además, como también se van a analizar 7 días de datos, en total será necesario esperar un mínimo de 3 semanas para poder comenzar con la extracción de datos.

Para este análisis final, la aplicación para realizar los análisis ya está completa y lista para su uso, por lo que, para la extracción de datos tanto de los Gateways como de los devices, además del posterior tratamiento y generación de gráficos, se hará utilizando esta nueva herramienta desarrollada.

5.5.3. Obtención de los datos de los devices

La obtención de los datos de los devices es muy similar a la de los Gateways, como ya se ha visto anteriormente.

Para obtenerlos, es necesarios también esperar las 3 semanas necesarias para que se aplique la nueva configuración en los devices y que se manden suficientes datos (7 días) con esta nueva configuración.

Una vez esperado ese tiempo, se utiliza la aplicación desarrollada para mayor comodidad a la hora de extraer los datos, además de que los siguientes procesos pueden hacerse de una forma prácticamente automática.

5.5.4. Procesado de datos

El formato de los datos obtenidos tanto del estudio inicial y de la iteración de mejora como el de esta configuración final es idéntico, por lo que el proceso del tratamiento de datos es completamente idéntico.

Utilizando la aplicación desarrollada este proceso se convierte en algo trivial, ya que es la propia aplicación la que se encarga de tratar los datos originales y convertirlos en archivos JSON que luego sirvan para alimentar los gráficos del análisis.

Este proceso lo lleva a cabo la aplicación en segundo plano, de forma que el usuario únicamente ve el progreso del proceso, no las acciones que se están llevando a cabo durante el procesamiento.

5.5.5. Generación de gráficos

La generación de gráficos sobre este último análisis se lleva a cabo de forma automática una vez se ha finalizado el tratamiento de datos, gracias al funcionamiento de la aplicación desarrollada en este trabajo.

Los tipos de gráficos que se obtienen son los mismo que los que se han conseguido durante los anteriores dos análisis:

- Mapas
- Ordenados por Gateway
- Distribuciones
- Ordenados por distancias al Main Gateway
- Ordenados por el ultimo SF utilizado

Pero, al igual que en la iteración de mejora, también se han generado gráficos comparativos. Esta vez, en vez de ser entre el anterior análisis y este recién hecho, se compararán los 3 estudios, el inicial, el de mejora y este final.

Estos gráficos comparativos entre los 3 estudios realizados seguirán la misma estructura que el de la Ilustración 96, que muestra la media de RSSI de los mensajes recibidos en cada Gateway durante los 7 días del análisis.

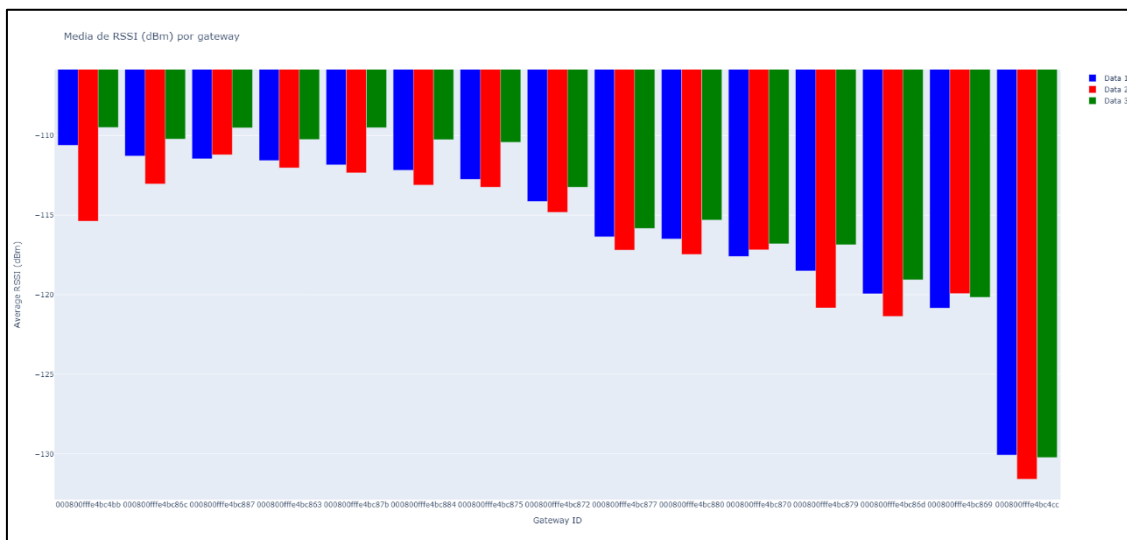


Ilustración 96: Comparativa de la media de RSSI recibida por cada Gateway

A lo largo del siguiente apartado, los gráficos seguirán la misma estructura que este, donde de azul se podrán ver los datos del estudio inicial, de rojo los de la iteración de mejora y de verde los de este último análisis final.

5.5.6. Interpretación del estado de la red

Analizando los gráficos comparativos obtenidos, se aprecian los cambios que ha soportado la red mediante los ajustes de parámetros QoS que se han ido realizando. En este apartado se analizarán estos cambios para ver que es en lo que ha cambiado la red, para después poder realizar diferentes valoraciones.

Empezando por el número de mensajes totales recibidos por cada Gateway, en la Ilustración 97 se puede apreciar que, con esta configuración de parámetros QoS final, el número total de mensajes recibidos en cada Gateway ha disminuido considerablemente respecto a la iteración de mejora. Pero, respecto a los datos obtenidos en el estudio inicial, el número de mensajes ha aumentado.

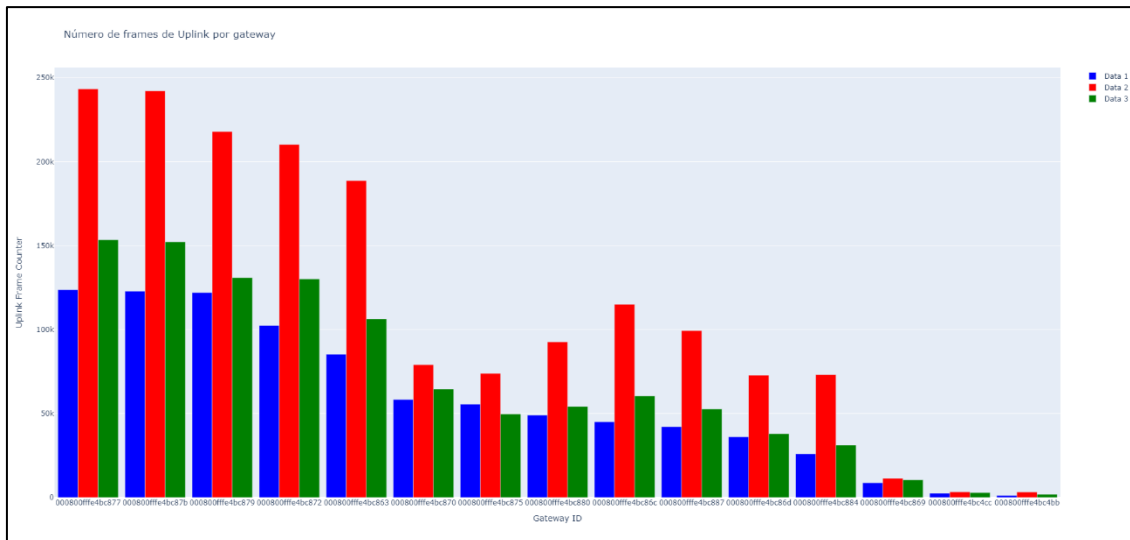


Ilustración 97: Comparación del número de mensajes recibidos por cada Gateway

Por otro lado, si se observa el número de Gateways, de media, al que llegan los mensajes de los devices en esta configuración final (Ilustración 98), se puede apreciar que, respecto a la iteración de mejora, los mensajes no llegan a tantos Gateways, pero, respecto a la configuración inicial, el número medio de Gateways al que llegan los mensajes ha aumentado considerablemente, estando la mayoría de devices en el rango de entre 1,5 y 3,5 Gateways por mensaje.

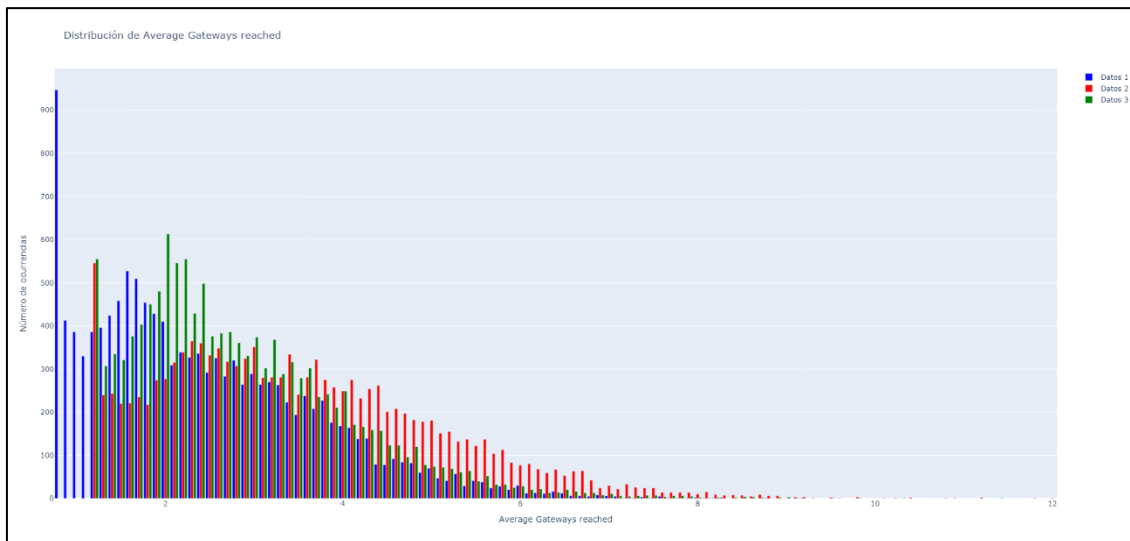


Ilustración 98: Comparación del número de Gateways al que llegan los mensajes

Observando el gráfico de la distribución del SF de los devices (Ilustración 99), se puede observar que los devices han vuelto a utilizar el SF que estaban utilizando antes de realizar ningún cambio en la red, trabajando la mitad de los devices aproximadamente con SF7, y siendo SF12 el segundo Spreading Factor más utilizado.

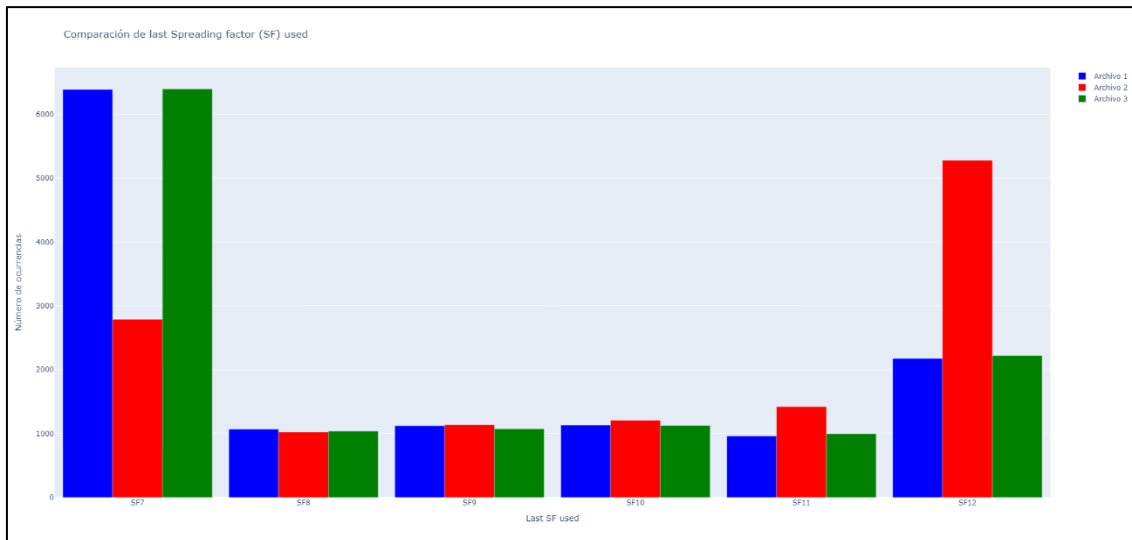


Ilustración 99: Comparación del último SF utilizado por los devices

Habiendo visto el cambio que han sufrido los devices en cuanto al SF que están utilizando, se puede observar un cambio acorde a esto en el ToA de los mensajes, ya que, cuanto menor es el SF, menor es el ToA de este.

En la Ilustración 100 se puede observar que el ToA se ha visto reducido a aproximadamente la mitad respecto al estudio anterior. Por otro lado, también se puede observar que el ToA deja de ser tan dependiente de la distancia a la que se encuentra el device del Main Gateway, como pasaba con la configuración inicial. Esto hace que los devices más cercanos tengan un ToA más alto que el que tenían al inicio del estudio, pero cuanto mayor es la distancia entre el device y el Gateway, en este estudio final el ToA se mantiene más o menos constante, al contrario que lo que pasaba al inicio, que iba aumentando.

Aun así, teniendo en cuenta esta gran diferencia, la media total del parámetro ToA acaba convergiendo en aproximadamente el mismo valor (740 ms) en el estudio inicial y en este final, dejando la media total de ToA de la iteración de mejora en un valor muy superior (1385 ms).

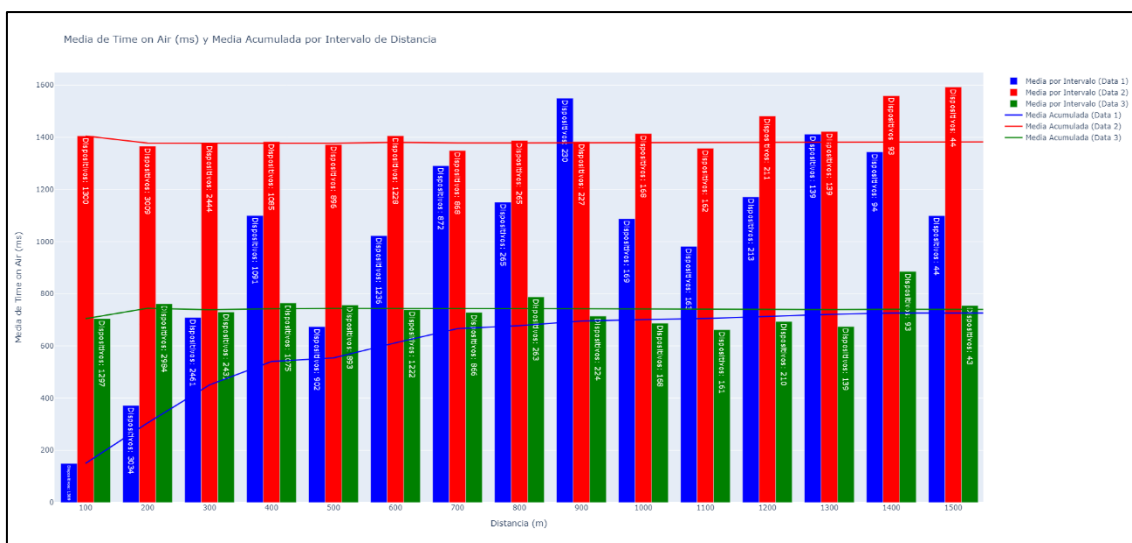


Ilustración 100: Comparación del ToA relativo a la distancia al Main Gateway

Analizando los valores de RSSI y SNR en función del SF que están utilizando los devices, se puede observar que los valores han vuelto a los que eran durante el estudio inicial, antes de realizar algún tipo de cambio en los parámetros avanzados de la red.

La mayor diferencia entre esta configuración final y la inicial se ve en SF7, donde el RSSI ha aumentado casi 3 dB (Ilustración 101) y el valor de SNR también ha aumentado, en aproximadamente 0.5 dBm (Ilustración 102).

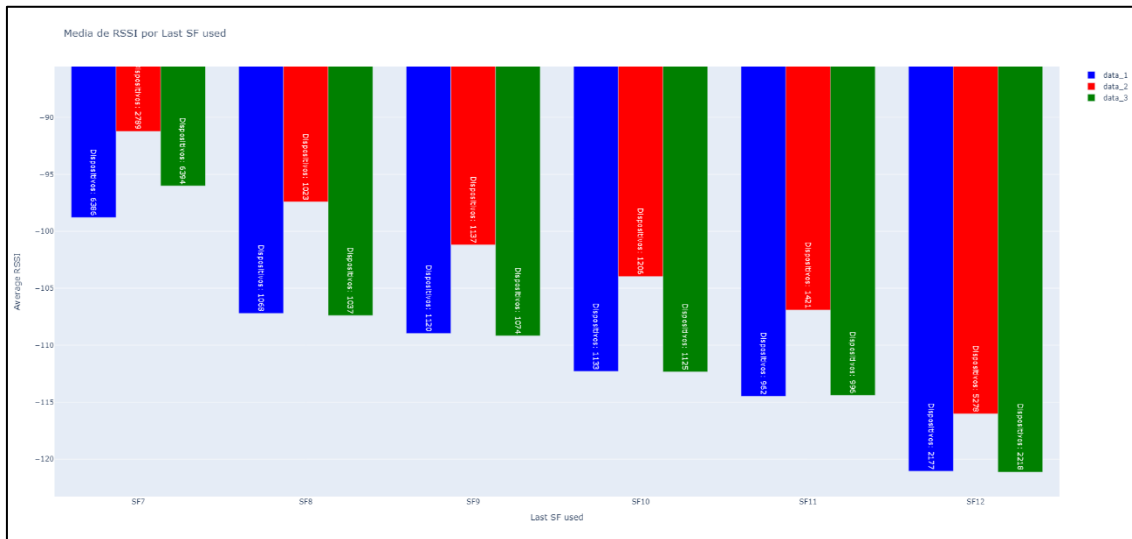


Ilustración 101: Comparación de RSSI en función de SF

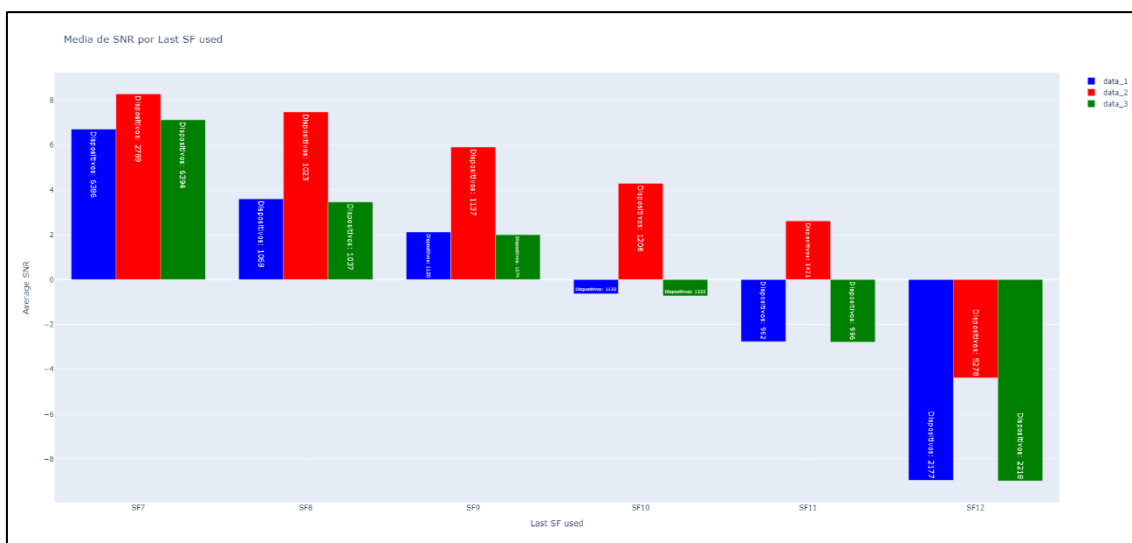


Ilustración 102: Comparación de SNR en función de SF

Observando las distribuciones de estos parámetros en vez de analizar la media por cada SF, se puede ver que tanto la distribución de RSSI (Ilustración 103) como la de SNR (Ilustración 104) se encuentran en un punto intermedio entre las distribuciones del estado inicial y las obtenidas en la iteración de mejora, acercándose más a esta última más que a la primera, pero sin llegar a ser tan parecida, sobre todo el SNR.

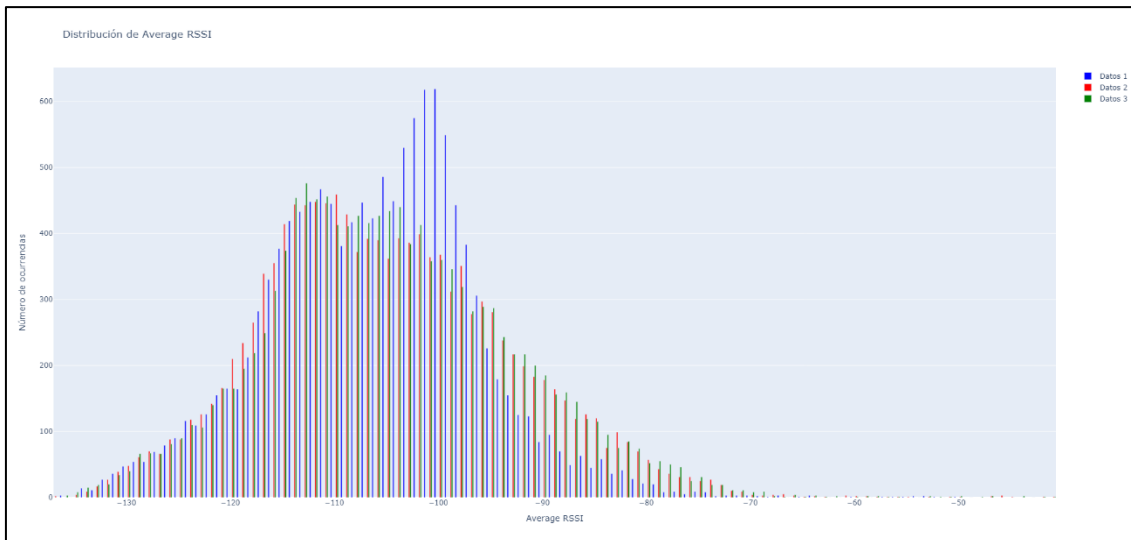


Ilustración 103: Comparación de la distribución de RSSI

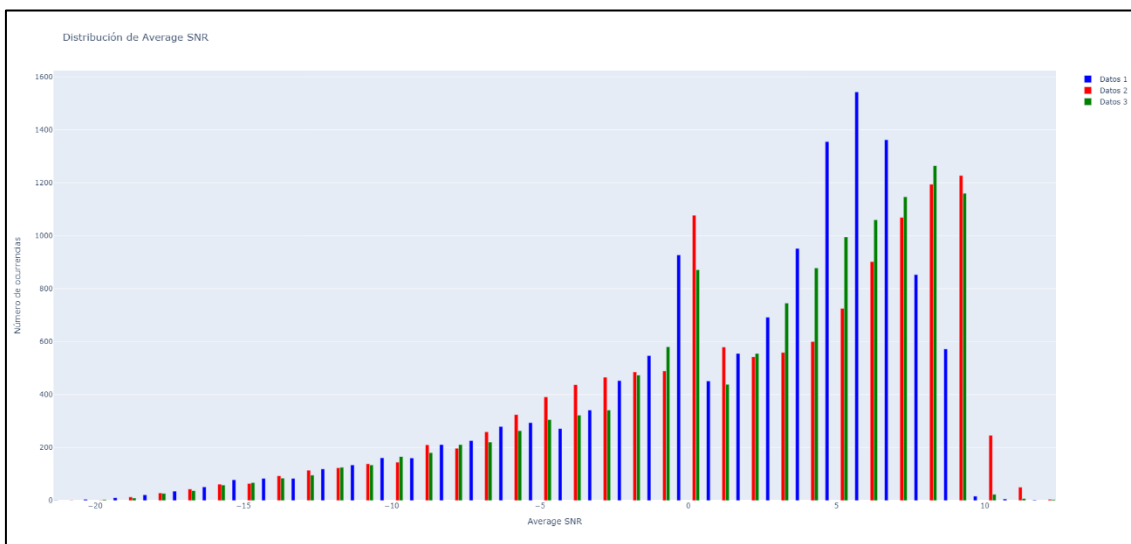


Ilustración 104: Comparación de la distribución de SNR

Finalmente, analizando los gráficos que muestran las tasas de recepción y pérdida de paquetes, se puede observar que, el número de mensajes recibidos ha aumentado respecto a la iteración de mejora, que a su vez había aumentado en comparación al estudio inicial.

Observando la distribución de mensajes recibidos (Ilustración 105), se puede ver que el mayor cambio ha sido en los decives de los cuales en la iteración de mejora se recibían 29-30 mensajes, ahora se reciben 31-32, aumentando el número total de mensajes recibidos. Además, es de mencionar que, en este análisis final, el número máximo de mensajes es de 32 y no 33 como pasaba en la iteración d mejora.

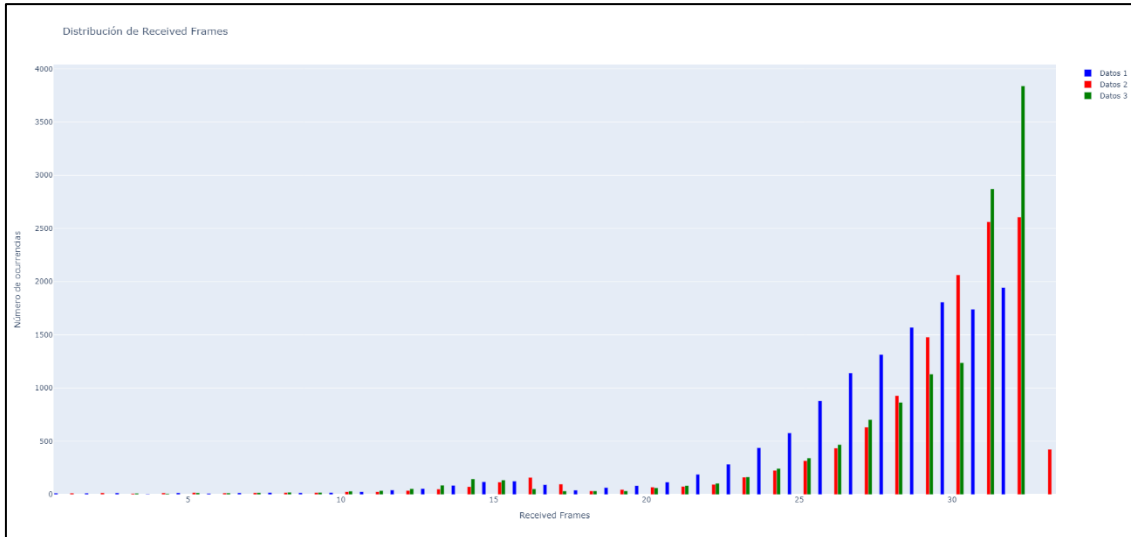


Ilustración 105: Comparación de número de mensajes recibidos

Por otro lado, analizando los mensajes perdidos en función de la distancia a la que se encuentran los dispositivos del Main Gateway, se puede observar que el cambio de la iteración de mejora al estudio final no es muy significativo, al contrario que con el estudio inicial, que la mejora es muy notable.

En la Ilustración 106 se puede observar que, comparando la iteración de mejora con este último estudio, dependiendo la distancia el porcentaje de pérdidas es similar en los casos. Pero, viendo la media total acumulada, se puede observar que en general, la pérdida de paquetes es menor con esta configuración final, donde anteriormente la media era de 9,6% de paquetes perdidos y con esta configuración es de 9,39%.

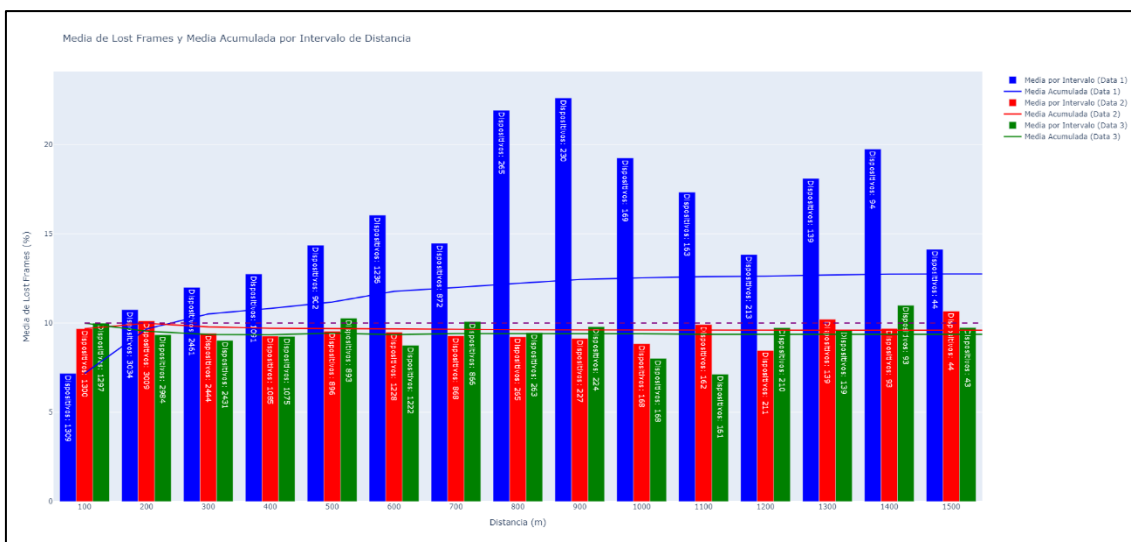


Ilustración 106: Comparación de porcentaje de mensajes perdidos en relación a la distancia al Main Gateway

En la Ilustración 107 se puede observar únicamente la media acumulada de los 3 estudios, viéndose claramente como en el estudio inicial la distancia era un factor muy determinante, al contrario que en los otros 2 estudios.

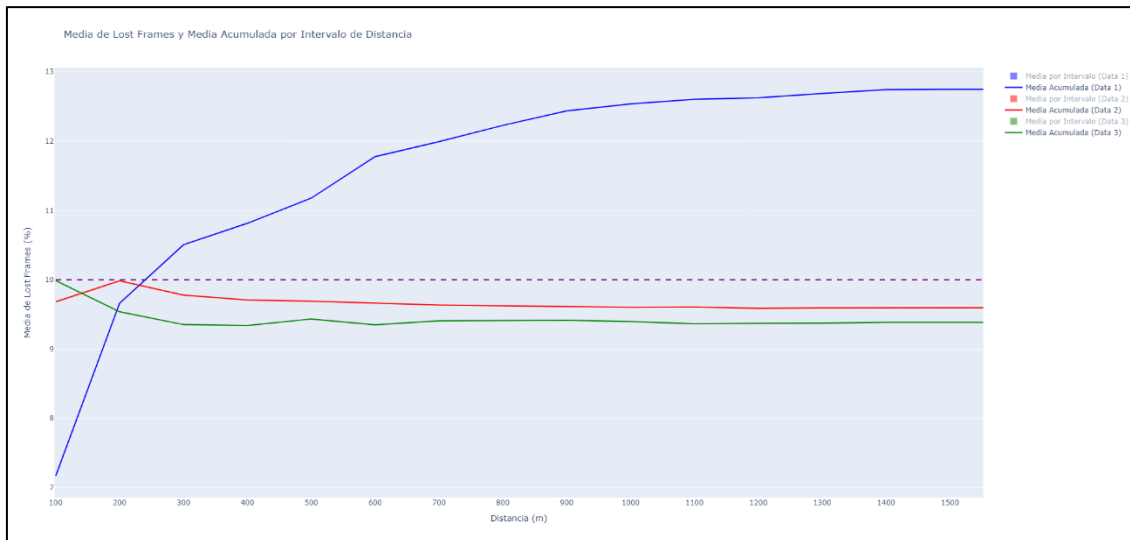


Ilustración 107: Comparación del porcentaje de la media acumulada de mensajes perdidos en relación a la distancia con el Main Gateway

Si se observa también el porcentaje de pérdidas de paquetes en función del SF que están utilizando los devices (Ilustración 108), se puede ver que, con los SF7, SF10 y SF11, el porcentaje de pérdidas de paquetes ha disminuido entre un 2% y un 3% en comparación a la iteración de mejora. Por otro lado, con los SF8 y SF9, las pérdidas se han mantenido constantes, pero, en lo que al SF12 respecta, las perdidas han aumentado un 4%.

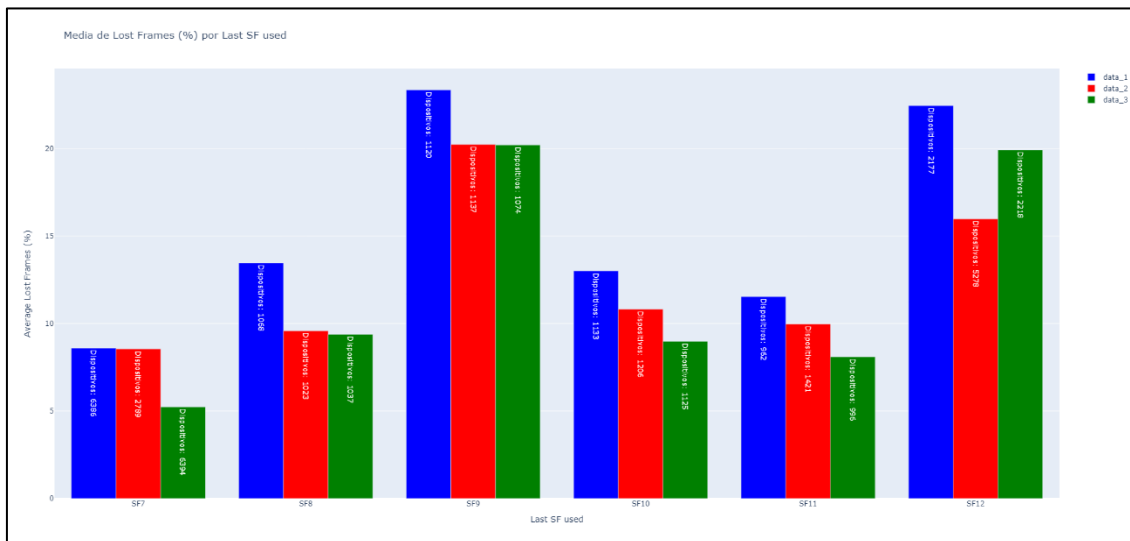


Ilustración 108: Comparación de porcentaje de perdidas en función del SF

5.5.7. Valoración de los resultados y conclusiones finales

Una vez analizados los gráficos con información relevante para el nuevo estado de la red, se pueden obtener varias conclusiones respecto al nuevo estado de esta, además de cómo ha variado con los diferentes cambios aplicados y cuáles han sido las mejoras que ha sufrido.

Empezando por el número total de mensajes recibidos por cada Gateway, se puede observar una gran disminución respecto a la iteración de mejora. Esto podría haber supuesto una mayor pérdida de paquetes, pero ha sido justo al revés, el porcentaje de paquetes perdidos ha

disminuido, por lo que, se deduce que la mayoría de los mensajes extra que se recibían en los Gateways eran redundantes y no aportaban información nueva.

Es de mencionar que, respecto al estado inicial, el número de paquetes totales ha aumentado ligeramente por Gateway. Esto es bastante probable que haya propiciado aumento de mensajes recibidos por cada devices, reduciendo el porcentaje de pérdidas.

En relación a esto último, también se ha podido apreciar una mejora en cuanto al número medio de Gateways que llegan los mensajes de los devices respecto a la configuración inicial, únicamente habiendo indicado a los devices que aumentasen su potencia mínima de transmisión. Pero, respecto a la iteración de mejora, el número de Gateways alcanzados por mensaje ha disminuido, ya que en la iteración se forzaba explícitamente a una redundancia espacial (o temporal). Aun así, vistos los resultados del porcentaje de mensajes perdidos, se deduce que no es necesaria esa redundancia forzada para garantizar la correcta recepción de los paquetes.

Dado que los parámetros QoS de redundancia forzada se han restablecido a por defecto en esta configuración final, la distribución de los Spreading Factor utilizados por los devices es casi idéntica a la del estudio inicial, dejando la gran mayoría de devices trabajando en SF7. Esto afecta en gran parte al uso de la batería, ya que cuanto menor es el SF, menor es el uso de la radio, por lo que la batería se agota más despacio. El uso de la batería es un parámetro difícil de medir en este proyecto, ya que las baterías tienen una vida útil de 15 años sin tener que cargarlas, y el proyecto dura únicamente algo más de 5 meses en su totalidad.

Pero, la disminución de SF no solo afecta a la batería, sino que también al ToA de los mensajes, ya que, al igual que el uso de la radio, cuando menor es el SF, menor es el ToA de los mensajes que se envían. Esto afecta a las colisiones entre los mensajes, siendo más probable que ocurran estas colisiones cuanto mayor sea el ToA.

Analizando los valores de RSSI y SNR, se puede ver que estos son prácticamente idénticos a los obtenidos del estudio inicial, excepto los de los devices que trabajan con SF7. Esto se debe al aumento de mínimo de potencia, ya que estos devices han sido obligados a transmitir con una mayor potencia de la necesaria, por lo que han aumentado la calidad de la señal recibida.

En comparación con la iteración de mejora, los valores sí que son relativamente peores, pero esto no afecta a la calidad de la señal, ya que tanto los valores del estudio inicial como los del final están dentro de los rangos del propio protocolo, por márgenes bastante amplios, siendo el SNR unos 12 dB mayor del mínimo necesario para la demodulación.

Finalmente, analizando los resultados obtenidos en cuanto a la pérdida de mensajes, se ha podido ver una continua mejora de este parámetro, dejándolo por debajo del 10% de mensajes perdidos totales.

En este estudio final, al haber aumentado la potencia mínima de los devices, se ha podido ver que los devices que usan SF7 han sido los más afectados, siendo notable la mejora en cuanto a RSSI y SNR. Teniendo en cuenta esto, se entiende por qué los devices utilizando un SF7 han tenido el menor porcentaje de pérdidas, un 5% aproximadamente.

En el resto de Spreading Factors, se ha visto una mejora respecto a la iteración de mejora, excepto en los que utilizan SF12, en los cuales las pérdidas han aumentado un 4%. Esto se debe a que en la iteración de mejora un gran número de devices trabajaban con SF12, y estos transmitían los mensajes con redundancia espacial o temporal, por lo que se entiende que el

número de mensajes recibidos fuese mayor. Aun así, respecto al estudio inicial, los porcentajes de pérdidas de paquetes han disminuido en cada uno de los diferentes Spreading Factors.

Analizando también el número de mensajes recibidos de cada device, se puede observar que los devices que en los otros estudios se estaban recibiendo 29 o 30 mensajes, de un total de 32, con esta configuración final se están recibiendo 31 o 32. Esto tiene sentido si se mira con la mejora del porcentaje de pérdidas de SF7, ya que los devices que utilizan SF7 suelen ser los más cercanos a los Gateways por lo que suelen tener una menor tasa de pérdidas.

Por lo que, se puede deducir que aumentando la potencia mínima de transmisión de los devices se ha conseguido que los devices que estaban teniendo pérdidas de 2 o 3 mensajes (en 7 días de duración del análisis), ahora tengan 0 o un único mensaje perdido.

En relación a la pérdida de paquetes, es de mencionar también que, durante el estudio inicial, estas pérdidas estaban bastante relacionadas con la distancia a la que se encontraban los devices de su Main Gateway, aumentando las dos proporcionalmente. Sin embargo, en la iteración de mejora y en este estudio final, este no es el caso, y la pérdida de paquetes se mantiene más o menos constante, indiferente a la distancia a la que se encuentra el device del Main Gateway.

En conclusión, en la iteración de mejora se han conseguido disminuir el porcentaje de mensajes perdidos, pero, además de ello, por consecuencia de los parámetros QoS cambiados, también se ha hecho que una cantidad importante de devices trabajen en SF12 en vez de SF7, aumentando el consumo de batería y el ToA de los mensajes. Hay que añadir que, también se ha conseguido mejorar la calidad de la señal recibida, como se puede ver tras analizar los parámetros RSSI y SNR.

Pero, con la configuración final aplicada, es decir, con únicamente el aumento de la potencia de transmisión mínima hasta igualarla con la máxima, se ha conseguido reducir el porcentaje de mensajes perdidos, incluso más que en la iteración de mejora. Además, los devices han vuelto a trabajar en SF7, aumentando su vida útil de batería y reduciendo el ToA de los mensajes considerablemente.

En contrapartida, los niveles de señal recibidos apenas han variado respecto a los del estudio inicial, únicamente mejorando los de SF7, pero esto no supone ningún problema, ya que desde un principio los niveles entraban dentro de los márgenes para poder demodular la señal por un amplio margen.

6. Planificación del proyecto

Este trabajo de fin de master se ha dividido en diferentes paquetes de trabajo, los cuales se han dividido en varias tareas. En este apartado se explicará lo realizado en cada una de estas tareas y se añadirá la duración de cada tarea al igual que la fecha en la que inició.

En los paquetes de trabajo centrales del trabajo (PT3, PT4 y PT5), las tareas realizadas son prácticamente idénticas, pero la finalidad global de cada paquete de trabajo es diferente.

6.1. Paquetes de trabajo

PT1: Metodología de captura de datos y resultados

Fecha de inicio: Semana 1.

Duración: 1 semana.

En este primer paquete de trabajo el objetivo es definir los parámetros importantes para hacer un buen análisis de la red IoT de Itelazpi además de su método de obtención de las bases de datos del *Network Server*. Por otro lado, también se definirán los gráficos que se generarán para hacer una buena interpretación de los resultados obtenidos.

El trabajo de fin de master (TFM) tendrá inicio el 1 de enero, el cual será considerado “semana 1” para el resto de este apartado.

T101: Definición de parámetros

Fecha de inicio: Semana 1.

Duración: 1 semana.

En esta primera tarea se definirán los parámetros necesarios para hacer un buen y completo análisis de la red IoT de Itelazpi. Para ello, se listarán todos los parámetros que el *Network Server* ofrece, tanto para los Gateways como para los dispositivos LoRa, y se hará un cribado para seleccionar los realmente útiles para el futuro análisis de la red.

T102: Definición de los métodos de captura de parámetros

Fecha de inicio: Semana 1.

Duración: 1 semana.

Mientras se definen los parámetros necesarios, se analizará cómo se pueden capturar estos de una forma automatizada, analizando los diferentes métodos de extracción de los datos y eligiendo el que mejor se adapte al proyecto.

T103: Definición de gráficos a obtener

Fecha de inicio: Semana 1.

Duración: 1 semana.

Teniendo en cuenta los parámetros que se pueden llegar a obtener, se definirán varios gráficos que servirán de ayuda para interpretar los datos previamente obtenidos y así poder analizar los resultados.

PT2: Creación de interfaz de usuario

Fecha de inicio: Semana 2.

Duración: 16 semanas.

Este segundo paquete de trabajo consistirá en la creación de una interfaz de usuario para poder replicar el análisis de la red en diferentes zonas de una manera sencilla y guiada. Para ello primero se generarán scripts básicos para la obtención de datos y generación de gráficos, los cuales más tarde se unirán en un único programa con una interfaz de usuario “user friendly”.

Este paquete de trabajo está dividido en el tiempo, ya que se utiliza el tiempo necesario para obtención de los datos de los diferentes paquetes de trabajo para llevarlo a cabo.

T201: Generación de scripts para la obtención de datos

Fecha de inicio: Semana 2.

Duración: 2 semanas.

En esta primera tarea se generarán los scripts para la obtención de los parámetros previamente definidos. Para ello, se utilizarán los métodos definidos en la tarea T102. Estos scripts se crearán para poder trabajar con ellos de la forma más automatizada posible, al igual que con la mayor flexibilidad a la hora de cambiar parámetros, para su futura reutilización en nuevos análisis.

T202: Generación de scripts para la visualización de gráficos

Fecha de inicio: Semana 3.

Duración: 2 semanas.

En esta tarea se generarán los scripts necesarios para visualizar los gráficos previamente definidos. Al igual que los scripts de la tarea T201, estos también se crearán para trabajar con ellos de la manera más automatizada y con la mayor flexibilidad posible.

T203: Creación de interfaz de usuario funcional

Fecha de inicio: Semana 9.

Duración: 3 semanas.

Esta tarea comienza después de la T401, mientras se lleva a cabo la obtención de los datos de la iteración de mejora (T402 y T403), ya que este proceso lleva su tiempo y es totalmente pasivo.

En esta tarea se creará una interfaz de usuario funcional unificando los scripts previamente generados, tanto de obtención de datos como de generación de gráficos.

T204: Creación de interfaz de usuario final

Fecha de inicio: Semana 15.

Duración: 3 semanas.

Al igual que pasaba con la anterior tarea, esta comienza después de la T501, mientras se lleva a cabo la obtención de los datos de la configuración final (T502 y T503).

En esta tarea se modificará la interfaz de usuario funcional para crear una más amigable para el usuario, haciendo más fácil su uso por diferentes personas, tanto participantes del proyecto,

como personas ajenas a él. Además, se creará un ejecutable para poder instalar la aplicación en cualquier ordenador de la empresa.

T205: Corrección de errores y casos límites

Fecha de inicio: Semana 9.

Duración: 9 semanas.

Durante esta tarea se analizará el código ya generado en busca de errores y se harán pruebas de casos límites, generando así una aplicación más robusta antes posibles fallos. Esta tarea iniciara a la vez que la T203 y no finalizara hasta haber completado la aplicación.

PT3: Estudio inicial

Fecha de inicio: Semana 5.

Duración: 4 semanas.

En este paquete de trabajo se englobará todo lo relacionado con el análisis de la red LoRa tal y como se encuentra por defecto, sin haber realizado ningún cambio. En esta fase se obtendrán los datos tanto de los Gateways como de los devices, se procesarán estos datos hasta conseguir los gráficos deseados y se analizará la red para dar una valoración final del estado de esta.

Este proceso se hará de una forma automatizada y que sea fácilmente escalable, tanto para llevarlo a cabo en los siguientes paquetes de trabajo como en diferentes proyectos similares en otras zonas de cobertura.

T301: Obtención de los datos de los Gateways

Fecha de inicio: Semana 5.

Duración: 1 semana.

En esta primera tarea del estudio inicial, se procederá a la obtención de los parámetros de los Gateways definidos anteriormente. Para ello, se hará uso del *Network Server* donde están dados de alta los Gateways y se obtendrán los datos de las bases de datos.

T302: Obtención de los datos de los devices

Fecha de inicio: Semana 5.

Duración: 1 semana.

En esta otra tarea del estudio inicial, se obtendrán los datos de los dispositivos que se habían definidos previamente. Para ello, se hará uso del *Network Server* donde están dados de alta los dispositivos y se obtendrán los datos de las bases de datos.

T303: Procesado de datos

Fecha de inicio: Semana 6.

Duración: 1 semana.

En esta tarea se procesarán los datos obtenidos previamente, tanto de los Gateways como de los devices, obteniendo diferentes resultados según el método de procesado. Para ello se hará uso de los scripts creados en la tarea T201.

T304: Generación de gráficos

Fecha de inicio: Semana 6.

Duración: 1 semana.

En esta tarea se obtendrán los gráficos definidos haciendo uso de los datos procesados de la tarea anterior, T303. Para llevar esto a cabo, se utilizarán los scripts previamente creados en la tarea T202.

T305: Interpretación del estado de la red

Fecha de inicio: Semana 7.

Duración: 1 semana.

Durante esta tarea se interpretarán los gráficos obtenidos de los datos del estudio inicial, haciendo más fácil la futura valoración de los resultados y la elección de los parámetros avanzados de la red que se deberían de modificar.

T306: Valoración de los resultados iniciales

Fecha de inicio: Semana 8.

Duración: 1 semana.

Esta tarea será la última del estudio inicial, en la cual se valorarán los resultados obtenidos de los gráficos y así obtener conclusiones del estado de la red, antes de realizar ningún cambio.

PT4: Iteraciones de mejora

Fecha de inicio: Semana 8.

Duración: 7 semanas.

En este paquete de trabajo se englobará todo lo relacionado con el análisis de la red LoRa después de haber analizado el estado inicial y haber cambiado ciertos parámetros avanzados de la red, para analizar el cambio que estos provocan.

Este paquete de trabajo comenzará con el replanteamiento de los parámetros QoS de la red que se quieren modificar para analizar su impacto. Después de esto, el análisis será prácticamente idéntico al anterior paquete de trabajo (PT3), contando con las mismas tareas.

Durante este trabajo solo se plantea hacer una iteración intermedia entre la inicial y la final, pero el número de iteraciones podría aumentar sin dificultades ya que se ha preparado para ello, para futuros análisis.

T401: Replanteo de nuevos parámetros QoS

Fecha de inicio: Semana 8.

Duración: 1 semana.

Esta es la primera tarea de las iteraciones de mejora, y en esta se plantearán nuevos parámetros de QoS de la red LoRa, con el fin de optimizarla, tras haber hecho un análisis de los datos obtenidos en el estudio inicial.

T402: Obtención de los datos de los Gateways

Fecha de inicio: Semana 9.

Duración: 3 semanas.

En esta tarea se procederá a la obtención de los parámetros de los Gateways definidos anteriormente. Para ello, se hará uso del *Network Server* donde están dados de alta los Gateways y se obtendrán los datos de las bases de datos.

Es necesario tener en cuenta que desde que se aplican los cambios de los parámetros QoS (T401) tiene que pasar un tiempo para que la red asimile los cambios y que los devices empiecen a trabajar con estos nuevos parámetros. Además, como la muestra de datos que se va a tomar para el análisis es de 1 semana, los devices deben transmitir con los nuevos parámetros al menos 7 días.

T403: Obtención de los datos de los devices

Fecha de inicio: Semana 9.

Duración: 3 semanas.

En esta otra tarea de las iteraciones de mejora, se obtendrán los datos de los devices que se habían definidos previamente. Para ello, se hará uso del *Network Server* donde están dados de alta los dispositivos y se obtendrán los datos de las bases de datos.

Al igual que con los Gateways en la T402, es necesario un tiempo de espera para que se apliquen los cambios de los parámetros QoS y que los devices transmitan 7 días de datos con los nuevos parámetros.

T404: Procesado de datos

Fecha de inicio: Semana 12.

Duración: 1 semana.

En esta tarea se procesarán los datos obtenidos en las tareas T402 y T403 de las iteraciones de mejora. Al igual que en la tarea T303, se obtendrán diferentes datos depende de su procesamiento, y para ello se hará uso de la interfaz de usuario funcional de la tarea T203.

T405: Generación de gráficos

Fecha de inicio: Semana 12.

Duración: 1 semana.

En esta tarea se obtendrán los gráficos definidos haciendo uso de los datos procesados de la tarea anterior, T404. Para llevar esto a cabo, se utilizará la interfaz de usuario funcional de la tarea T203.

T406: Interpretación del estado de la red

Fecha de inicio: Semana 13.

Duración: 1 semana.

Durante esta tarea se interpretarán los gráficos obtenidos de los datos de la iteración de mejora, haciendo más fácil la futura valoración de los resultados y la elección de los nuevos parámetros avanzados de la red que se deberían de modificar.

T407: Valoración de los resultados tras los cambios

Fecha de inicio: Semana 14.

Duración: 1 semana.

Esta tarea será la última de la iteración de mejora, en la cual se valorarán los resultados obtenidos de los gráficos y así obtener conclusiones del estado de la red, para así poder realizar nuevos cambios viendo el progreso de la red respecto a la situación inicial.

PT5: Configuración final

Fecha de inicio: Semana 14.

Duración: 7 semanas.

En este paquete de trabajo se englobará todo lo relacionado con el análisis de la red LoRa después de haber realizado el estudio inicial y la iteración de mejora y haber cambiado ciertos parámetros avanzados de la red, para analizar el cambio que estos provocan.

Este paquete de trabajo será prácticamente idéntico al anterior paquete de trabajo (PT4), contando con las mismas tareas, pero siendo este el encargado de encontrar una configuración final óptima.

T501: Replanteo de nuevos parámetros QoS

Fecha de inicio: Semana 14.

Duración: 1 semana.

En esta tarea se plantearán nuevos parámetros de QoS de la red LoRa, con el fin de optimizarla, tras haber hecho un análisis de los datos obtenidos en el estudio inicial y tras la iteración de mejora.

T502: Obtención de los datos de los Gateways

Fecha de inicio: Semana 15.

Duración: 3 semanas.

En esta tarea se procederá a la obtención de los parámetros de los Gateways definidos anteriormente. Para ello, se hará uso del *Network Server* donde están dados de alta los Gateways y se obtendrán los datos de las bases de datos.

Esta tarea es similar a la T402, por lo que también será necesario un tiempo de espera pasivo para que la red se ajuste y poder obtener datos nuevos.

T503: Obtención de los datos de los devices

Fecha de inicio: Semana 15.

Duración: 3 semanas.

En esta otra tarea, se obtendrán los datos de los devices que se habían definidos previamente. Para ello, se hará uso del *Network Server* donde están dados de alta los dispositivos y se obtendrán los datos de las bases de datos.

Al igual que en el anterior paquete de trabajo (PT4), es necesario un tiempo de espera para que se apliquen los cambios de los parámetros QoS y que los devices transmitan 7 días de datos con los nuevos parámetros.

T504: Procesado de datos

Fecha de inicio: Semana 18.

Duración: 1 semana.

En esta tarea se procesarán los datos obtenidos en las tareas T502 y T503 de la configuración final. Al igual que en las tareas T303 y T404, se obtendrán diferentes datos depende de su procesamiento, y para ello se hará uso de la interfaz de usuario final de la tarea T204.

T505: Generación de gráficos

Fecha de inicio: Semana 18.

Duración: 1 semana.

En esta tarea se obtendrán los gráficos definidos haciendo uso de los datos procesados de la tarea anterior, T504. Para llevar esto a cabo, se utilizará la interfaz de usuario final de la tarea T204.

T506: Interpretación del estado de la red

Fecha de inicio: Semana 19.

Duración: 1 semana.

Durante esta tarea se interpretarán los gráficos obtenidos de los datos de la configuración final, haciendo más fácil la futura valoración de los resultados y la obtención de conclusiones finales de todo el trabajo.

T507: Valoración de los resultados y conclusiones finales

Fecha de inicio: Semana 20.

Duración: 1 semana.

Esta tarea será la última de la configuración final, en la cual se valorarán los resultados obtenidos de los gráficos y así obtener conclusiones del estado actual de la red. Además, se obtendrán las conclusiones finales del trabajo teniendo en cuenta el estado inicial de la red, las iteraciones de mejora y la configuración final.

PT6: Gestión del proyecto

Fecha de inicio: Semana 1.

Duración: 23 semanas.

Este paquete de trabajo es el último. En él, las diferentes tareas estarán relacionadas en su totalidad con la documentación del trabajo, además de en redactar todo lo realizado durante este trabajo de fin de master. Como los demás paquetes de trabajo, este también estará dividido en diferentes tareas realizadas.

T601: Reuniones de seguimiento y control

Fecha de inicio: Semana 1.

Duración: 23 semanas.

Esta tarea se realizará desde el comienzo del proyecto hasta la conclusión de la finalización de este. Estas reuniones se realizarán con el director del TFM periódicamente.

T602: Documentación del proyecto

Fecha de inicio: Semana 1.

Duración: 23 semanas.

Esta tarea tiene como objetivo resumir y redactar todo lo realizado desde el comienzo del TFM. Para ello, se realizará una memoria cumpliendo con todos los campos exigidos y siguiendo los pasos descritos.

6.2. Diagrama de GANTT

En la Ilustración 109 se puede observar el diagrama de GANTT del TFM. En este, se pueden observar todas las tareas realizadas durante el trabajo, las cuales se pueden diferenciar por los paquetes de trabajo. En el diagrama, se pueden ver claramente las similitudes de los paquetes de trabajo 3, 5 y 5, los cuales contienen tareas prácticamente idénticas.

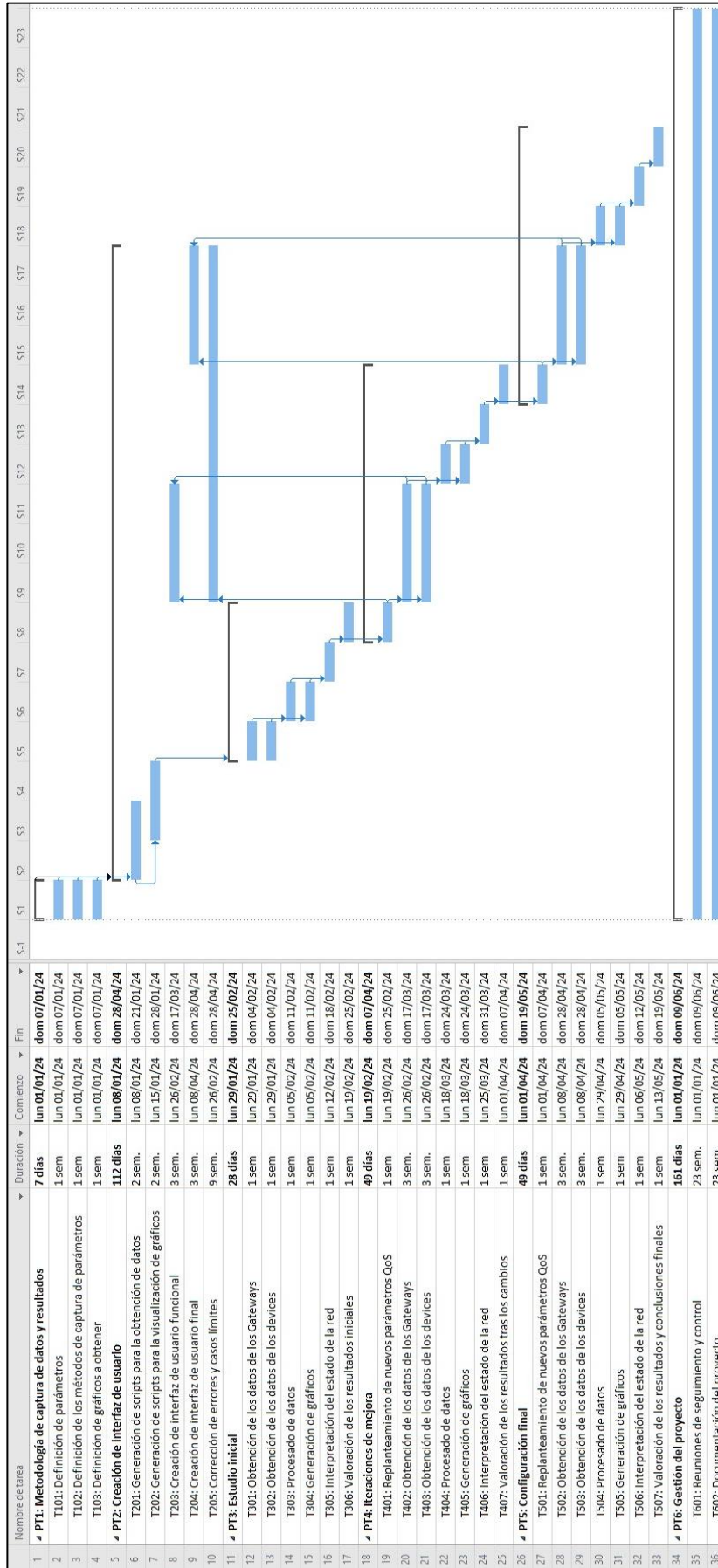


Ilustración 109: Diagrama de Gantt

7. Presupuesto

En este apartado se muestra el presupuesto del trabajo de fin de master. En este se calcula la cantidad de dinero necesaria para llevar a cabo el TFM, para poder gestionar los costes y para poder cumplir los objetivos de este. Para ello se definen tres apartados: gasto de personal, amortizaciones y gastos.

A continuación, se hace un análisis en profundidad de cada uno de estos apartados, y se concluye con un breve resumen.

7.1. Gasto de personal

El equipo está formado por dos ingenieros Senior y un ingeniero Junior, por lo cual, es necesario diferenciarlos a la hora de realizar los cálculos de los sueldos y nóminas (Tabla 12).

Código	Nombre	Coste (€/h)	Horas	Total (€)
I1	Ingeniero Senior	60	100	6.000
I2	Ingeniero Senior	60	100	6.000
I3	Ingeniero Junior	35	600	21.000
			Total (€)	33.000

Tabla 12: Sueldos y nóminas

7.2. Amortizaciones

Con las amortizaciones se pretende distribuir el gasto de algunos elementos a lo largo de su vida útil. Ya que estos elementos tienen una vida útil más allá del trabajo, no se incluye su valor en totalidad, sino que se incluye un porcentaje (Tabla 13).

Hay que tener en cuenta que, aunque la licencia de Orbiwise tengan solo un año de duración, al tener un coste tan elevado se ha decidido amortizarla. El coste de la licencia es proporcional al número de Gateways dados de alta en ese *Network Server*, 400€/Gateway al año. Es por eso que, en la columna "cantidad" el valor es mayor que uno, ya que se han tenido en cuenta el número de Gateways dados de alta.

Código	Concepto	Unidades	Coste unitario	Vida útil (años)	Duración (meses)	Total amortizado (€)
PC	Ordenador portátil	1	1.500	4	5	156,25
CN	Contadores	13.000	100	15	5	36.111,11
GW	Gateways	16	2.200	6	5	2.444,44
OW	Licencia anual Orbiwise	16	400	1	5	2.666,66
					Total (€)	41.378,47

Tabla 13: Amortizaciones

7.3. Resumen

Para finalizar, se combinan todas las anteriores partes para obtener la cantidad total (Tabla 14). Aunque para ello, hay que tener en cuenta que hay que sumar los costes indirectos, los cuales pueden ser la electricidad consumida por los servidores de Orbiwise y Gateways entre otros. Todos esos costes se incluirán como un 10% de los costes directos.

Finalmente, hay que añadir otro 10% de imprevistos, ya que cabe la posibilidad de que algún Gateway falle o por cualquier otra razón haya que hacer algún tipo de gasto no planeado.

Concepto	Total (€)
Gasto de personal	33.000
Amortizaciones	41.378,47
Subtotal I	74.378,47
Costes indirectos (10%)	7.437,85
Subtotal II	81.816,32
Imprevistos (10%)	8.181,63
Total (€)	89.997,95

Tabla 14: Resumen de presupuesto

8. Conclusiones

Concluyendo el trabajo de fin de master, se puede decir que los objetivos han sido cumplidos, tanto el principal como los secundarios. Se ha logrado optimizar la red IoT LoRa desplegada por Itelazpi mediante ajustes realizados en los parámetros avanzados de la red. Además, se ha desarrollado una aplicación de escritorio para poder realizar nuevos análisis en diferentes zonas de estudio de una forma sencilla.

Para ello, se han realizado varios estudios sobre la red, observando así, como afectaban a la calidad de servicio los cambios de ciertos parámetros QoS. Estos análisis se han podido automatizar en la medida de lo posible gracias a la creación de la aplicación de escritorio, la cual no es únicamente útil en el marco de este TFM, sino que puede utilizarse para cualquier otro estudio similar en la red LoRa de Itelazpi.

Analizando los resultados obtenidos tras los diferentes análisis realizados, se ha concluido que únicamente forzando a los devices a transmitir a máxima potencia, sea cual sea su SF, se ha conseguido minimizar la pérdida de mensajes de estos devices por debajo del 10%, sin alterar en gran medida el resto de variables, como podían ser el SF, ToA, RSSI, SNR, etc.

Por otro lado, gracias al trabajo realizado a lo largo de 3 años en las prácticas de cooperación educativa, se ha podido trabajar a fondo con una red LoRa real, pudiendo ver el proceso que supone desplegar una red. Desde el despliegue de los Gateways hasta la transmisión de datos a los diferentes clientes finales, pasando por un tratamiento y almacenaje de datos exhaustivo.

En lo personal, el trabajo ha sido todo un reto, ya que ni Itelazpi ni Orbiwise contaban con una manera de poder visualizar el estado de la red de grupos (municipios) específicos, por lo que, haciendo uso de las herramientas disponibles, se ha conseguido desarrollar una manera de realizar esto, lo que lo convierte en algo muy satisfactorio.

Además, con las conclusiones obtenidas, a partir de ahora el estado de la red mejorará poco a poco, ya que se ha descubierto una manera de poder reducir la pérdida de paquetes sin grandes efectos secundarios.

Resumiendo, gracias al desarrollo de la aplicación de escritorio y las conclusiones obtenidas a través de los estudios realizados, el análisis y la optimización de la red LoRa es mucho más sencilla, por lo que, en caso de ser necesario, se tendrán todas las facilidades para poder hacerlo.

9. Bibliografía

- [1] “LoRaWAN architecture”, *The Things Network*. [En línea]. Disponible en: <https://www.thethingsnetwork.org/docs/lorawan/architecture/>. [Consultado: 27-may-2024].
- [2] “Smart metering with LoRaWAN”, *TEKTELIC*, 30-oct-2023. [En línea]. Disponible en: <https://tektelic.com/expertise/lorawan-smart-metering/>. [Consultado: 27-may-2024].
- [3] “Home - OrbiWise - LoRaWAN Network Server, Internet of things technologies”, *OrbiWise - LoRaWAN Network Server, Internet of things technologies*, 12-abr-2021. [En línea]. Disponible en: <https://orbiwise.com/>. [Consultado: 27-may-2024].
- [4] “LoRa y el duty cycle”, *Lpwan.es*, 09-mar-2022. [En línea]. Disponible en: <https://lpwan.es/lora/lora-y-el-duty-cycle/>. [Consultado: 27-may-2024].
- [5] “¿Qué Es el Web Scraping? Cómo Extraer Legalmente el Contenido de la Web”, *Kinsta*®, 28-jul-2022. [En línea]. Disponible en: <https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/>. [Consultado: 27-may-2024].
- [6] “Distance on a sphere: The haversine formula”, *Esri Community*, 05-oct-2017. [En línea]. Disponible en: <https://community.esri.com/t5/coordinate-reference-systems-blog/distance-on-a-sphere-the-haversine-formula/ba-p/902128>. [Consultado: 27-may-2024].
- [7] “Welcome to GeoPy’s documentation! — GeoPy 2.4.1 documentation”, *Readthedocs.io*. [En línea]. Disponible en: <https://geopy.readthedocs.io/en/stable/>. [Consultado: 27-may-2024].
- [8] “Folium — folium 0.16.1.dev67+gb80e7e92 documentation”, *Github.io*. [En línea]. Disponible en: <https://python-visualization.github.io/folium/latest/index.html>. [Consultado: 27-may-2024].
- [9] “Plotly”, *Plotly.com*. [En línea]. Disponible en: <https://plotly.com/python/>. [Consultado: 27-may-2024].
- [10] “tkinter — Python interface to Tcl/Tk”, *Python documentation*. [En línea]. Disponible en: <https://docs.python.org/es/3/library/tkinter.html>. [Consultado: 27-may-2024].
- [11] “Tkcalendar — tkcalendar 1.5.0 documentation”, *Readthedocs.io*. [En línea]. Disponible en: <https://tkcalendar.readthedocs.io/en/stable/>. [Consultado: 27-may-2024].
- [12] “Figma: The Collaborative Interface Design Tool”, *Figma*. [En línea]. Disponible en: <https://www.figma.com/>. [Consultado: 27-may-2024].

- [13] “Tkinter-Designer: An easy and fast way to create a Python GUI ”, *GitHub*. [En línea]. Disponible en: <https://github.com/ParthJadhav/Tkinter-Designer>. [Consultado: 27-may-2024].
- [14] “Git - git con bash”, *Git-scm.com*. [En línea]. Disponible en: <https://git-scm.com/book/es/v2/Ap%C3%A9ndice-A%3A-Git-en-otros-entornos-Git-con-Bash>. [Consultado: 27-may-2024].
- [15] “Fast and reliable end-to-end testing for modern web apps”, *Playwright.dev*. [En línea]. Disponible en: <https://playwright.dev/python/>. [Consultado: 27-may-2024].
- [16] “PyInstaller manual — PyInstaller 6.7.0 documentation”, *Pyinstaller.org*. [En línea]. Disponible en: <https://pyinstaller.org/en/stable/>. [Consultado: 27-may-2024].
- [17] “Inno setup”, *Jrsoftware.org*. [En línea]. Disponible en: <https://jrsoftware.org/isinfo.php>. [Consultado: 27-may-2024].
- [18] “Device classes”, *The Things Network*. [En línea]. Disponible en: <https://www.thethingsnetwork.org/docs/lorawan/classes/>. [Consultado: 27-may-2024].
- [19] “RSSI and SNR”, *The Things Network*. [En línea]. Disponible en: <https://www.thethingsnetwork.org/docs/lorawan/rssi-and-snr/>. [Consultado: 27-may-2024].
- [20] “LoRa alliance - homepage - LoRa alliance®”, *LoRa Alliance®*, 03-nov-2023. [En línea]. Disponible en: <https://lora-alliance.org/>. [Consultado: 27-may-2024].
- [21] “Itelazpi – Digitalizaziorako funtsezkoak”, *Itelazpi.eus*. [En línea]. Disponible en: <https://www.itelazpi.eus/es/>. [Consultado: 27-may-2024].
- [22] IoT Consulting, “LoRa vs LoRaWAN - ¿Cuál es la diferencia?”, *Tu fuente experta en IoT*, 15-mar-2021. [En línea]. Disponible en: <https://iotconsulting.tech/lora-vs-lorawan/>. [Consultado: 27-may-2024].